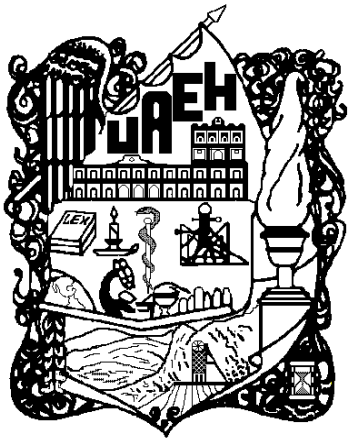


UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

ÁREA ACADÉMICA DE INGENIERÍA Y ARQUITECTURA

LICENCIATURA EN INGENIERÍA INDUSTRIAL

TESIS

Optimización de Localizaciones: Aplicación de Algoritmos Evolutivos y Técnicas Avanzadas de Ciencia de Datos

Para obtener el Grado de
Ingeniero Industrial

P R E S E N T A

Luis Alfredo González Zamora

Director
Dr. Jaime Garnica González

Codirector(a)
Dr. Héctor Rivera Gómez

Comité tutorial
Dr. Jaime Garnica González
Dr. Héctor Rivera Gómez
Dr. Juan Carlos Seck Tuoh Mora
Dr. Joselito Medina Marín

Cd. del Conocimiento, Mineral de la Reforma, Hidalgo, México. Diciembre de 2023

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences
Área Académica de Ingeniería y Arquitectura
Department of Engineering and Architecture

Mineral de la Reforma, Hgo., a 05 de diciembre del 2023

Número de control: ICBI-AAIyA/3626/2023
Asunto: Autorización de impresión de tesis.

MTRA. OJUKY DEL ROCIO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

De conformidad con el Artículo 13, fracción III del Reglamento de Titulación vigente, el Comité Tutorial de la Tesis titulada **"Optimización de Localizaciones: Aplicación de Algoritmos Evolutivos y Técnicas Avanzadas de Ciencia de Datos"**, realizada por el sustentante LUIS ALFREDO GONZÁLEZ ZAMORA, con número de cuenta 278213 perteneciente al Programa Educativo de la Licenciatura de Ingeniería Industrial, una vez que se ha revisado, analizado y evaluado el documento recepcional, tiene a bien extender la presente:

AUTORIZACIÓN DE IMPRESIÓN

Por lo que el sustentante deberá cumplir los requisitos del Reglamento de Titulación vigente para obtener el título profesional de licenciatura por elaboración de tesis.

Atentamente
"Amor, Orden y Progreso"
Comité Tutorial

Dr. Jaime Garnica González
Director de Tesis

Dr. Héctor Rivera Gómez
Codirector de Tesis

Dr. Juan Carlos Seck Tuoh Mora
Integrante del Comité

Dr. Joselito Medina Marín
Integrante del Comité

Ciudad del Conocimiento
Carretera Pachuca-Tulancingo km 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: 771 71 720 00 ext. 2231 Fax 2109
direccion_icbi@uaeh.edu.mx
www.uaeh.edu.mx

DEDICATORIA

A mi querida madre July y mi estimada tía Marta.

A ustedes, pilares fundamentales en mi vida, dedico el esfuerzo y los desvelos que significó esta investigación. Sus sabios consejos y amor incondicional fueron la fuerza que impulsó la culminación de este trabajo.

Que este logro académico sea un pequeño homenaje a la incansable labor de apoyo y estímulo que han emprendido a mi favor desde siempre. Son mi mayor inspiración para superarme cada día como persona y profesional.

Con profundo cariño y admiración, les dedico esta tesis. Ojalá se sientan orgullosas de este nuevo peldaño que gracias a ustedes he podido escalar. Las quiero mucho.

AGRADECIMIENTOS

En primer lugar quiero expresar mi más profundo agradecimiento a mi madre July y mi tía Marta, por su apoyo incondicional y sus sabios consejos durante todo mi camino académico. Sin ellas, este logro no hubiera sido posible.

Asimismo, agradezco a mi amigo Julio por despejar varias de mis dudas y brindarme orientación en momentos clave del desarrollo de este trabajo de investigación. Su guía y disposición fueron vitales.

También deseo reconocer a la Universidad Autónoma del Estado de Hidalgo y en especial a los profesores del programa de Ingeniería Industrial, por los conocimientos y herramientas proporcionadas durante mi formación. Llevaré siempre sus enseñanzas conmigo.

Además, quisiera mencionar al Comité Tutorial de mi tesis de licenciatura, compuesto por los doctores Jaime Garnica González, Héctor Rivera Gómez, Juan Carlos Seck Tuoh Mora y Joselito Medina Marín, agradezco su presencia en este proceso académico.

Finalmente, un especial reconocimiento a las instituciones que facilitaron los datos utilizados en este estudio, en particular el SIAP, SAGARPA e INEGI. La información estadística de acceso público fue esencial como insumo de la investigación.

La culminación de esta tesis de licenciatura ha requerido esfuerzo, dedicación y apoyo. Estoy profundamente agradecido con todos quienes, de una u otra forma, han contribuido a este logro que marca el inicio de una nueva etapa en mi vida profesional.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICAS

ÍNDICE DE TABLAS

GLOSARIO DE TÉRMINOS

A

Análisis de agrupamiento
Técnicas para dividir un conjunto de objetos en grupos con alto grado de asociación entre sus miembros. 98, 140

Autocorrelación
Correlación de una serie de tiempo consigo misma a través del tiempo. Indica dependencia entre observaciones sucesivas. 58

D

Dendrograma
Representación visual en forma de árbol de la agrupación jerárquica en un análisis clúster. 100

E

Estacionariedad
Propiedad de una serie de tiempo en la que sus estadísticas como media y varianza son constantes a lo largo del tiempo..... 91

Estandarización
Escalado de variables numéricas para que tengan media 0 y desviación estándar 1..... 67

F

Folium
Librería de Python para visualización de datos geográficos. 100, 104

Frente de Pareto
En optimización multiobjetivo, conjunto de soluciones eficientes no dominadas entre sí. 7

H

Haversine
Fórmula para calcular distancias entre puntos dados sus coordenadas. 118

Heterocedasticidad
Situación en la que la varianza de una variable no es constante. Violación de supuestos en modelos de regresión. 76, 81, 82

I

INEGI
Instituto Nacional de Estadística y Geografía de México..... 4, 28

M

MAE
Error absoluto medio, métrica para evaluar precisión de predicciones. ... 66, 69, 70, 74, 75, 87, 89, 90, 93, 4

MAPE
Error porcentual absoluto medio, métrica de precisión expresada en porcentaje. 93

Matplotlib
Librería de Python para visualización y graficado. 21

O

OneHotEncoding
Técnica que convierte variables categóricas en formato numérico interpretable. 66, 67

Optimización multiobjetivo
Técnicas para encontrar soluciones que optimicen múltiples criterios o funciones objetivo simultáneamente. 17

OR-Tools
Librería de Google para modelado y optimización combinatoria. 122

P

Pandas
Librería de Python para análisis de datos y manipulación de datos estructurados. 21, 28, 104

R

R2
Coeficiente de determinación, indica proporción de varianza explicada por un modelo..... 75, 76

S

SAGARPA
Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación de México. 4, 21
Seaborn
Librería de Python que provee una interfaz para Matplotlib, enfocada en estadística. 21
SIACON
Sistema de Información Agroalimentaria de Consulta. 4, 27
SIAP
Sistema de Información Agroalimentaria de Consulta, fuente de datos productivos. 4, 21
Spark
Framework open-source para procesamiento distribuido en clúster que puede manejar grandes conjuntos de datos (..... 7

T

TSP
Problema del agente viajero para encontrar la ruta más corta visitando todos los puntos. 122, 128, 130, 145, 146

V

VRP
Problema de enrutamiento de vehículos para optimizar rutas de entrega. 122, 128, 130, 134, 147

X

XGBoost
Implementación optimizada de gradient boosting. Rápida y efectiva para problemas de regresión. ... 66, 68, 71, 73

RESUMEN

El presente estudio tuvo como objetivo determinar la ubicación óptima para la instalación de un rastro Tipo Inspección Federal (TIF) en el estado de Hidalgo, México, considerando factores como la producción y exportación potencial de ganado bovino en los diferentes municipios, así como la importancia de identificar una localización estratégica debido a que este tipo de rastro permite la exportación de carne bovina a otros países.

Se realizó un análisis cuantitativo utilizando datos históricos de 2006 a 2021 sobre la producción de ganado bovino en los 84 municipios de Hidalgo. Las variables analizadas fueron: producción en canal, producción en pie, precio promedio, valor de producción, peso promedio y cabezas de ganado bovino. Se generaron predicciones con técnicas de series de tiempo para prever situaciones hasta el año 2030 y se identificaron municipios con capacidad exportadora actual y futura.

Además, se modelaron las relaciones entre variables críticas como producción, precio y población, y se aplicaron algoritmos de optimización multiobjetivo para minimizar costos de transporte y asignación de demanda considerando restricciones logísticas y de capacidad, así como algoritmos de optimización para minimizar costos de transporte y asignación de demanda considerando restricciones.

Los resultados indican que el municipio de Molango de Escamilla presenta las mejores condiciones para la ubicación del rastro TIF, según métricas de distancia recorrida y costos calculados. Asimismo, se identificaron municipios estratégicos para el establecimiento de centros de acopio y distribución auxiliares, así como posibles centros de acopio en otros municipios.

Este estudio provee una metodología efectiva para determinar localizaciones óptimas aplicando herramientas de inteligencia artificial y optimización, permitiendo maximizar el potencial exportador de carne bovina en la región y minimizar costos logísticos asociados. Los hallazgos pueden guiar la toma de decisiones informada sobre inversiones en infraestructura ganadera con impacto económico regional.

ABSTRACT

The objective of this study was to determine the optimal location for the installation of a Federal Inspection Type (TIF) slaughterhouse in the state of Hidalgo, Mexico, considering factors such as the production and export potential of beef cattle in the different municipalities, as well as the importance of identifying a strategic location because this type of slaughterhouse allows the export of beef to other countries.

A quantitative analysis was carried out using historical data from 2006 to 2021 on cattle production in the 84 municipalities of Hidalgo. The variables analyzed were: carcass production, live production, average price, production value, average weight and head of cattle. Predictions were generated using time series techniques to forecast situations up to the year 2030, and municipalities with current and future export capacity were identified.

In addition, relationships between critical variables such as production, price and population were modelled, and multi-objective optimization algorithms were applied to minimize transport costs and demand allocation considering logistic and capacity constraints, as well as optimization algorithms to minimize transport costs and demand allocation considering constraints.

The results indicate that the municipality of Molango de Escamilla presents the best conditions for the location of the TIF slaughterhouse, according to distance travelled metrics and calculated costs. In addition, strategic municipalities were identified for the establishment of auxiliary collection and distribution centers, as well as possible collection centers in other municipalities.

This study provides an effective methodology to determine optimal locations by applying artificial intelligence and optimization tools to maximize the region's beef export potential and minimize associated logistics costs. The findings can guide informed decision making on livestock infrastructure investments with regional economic impact.

INTRODUCCIÓN

La ganadería bovina representa una actividad económica fundamental en México. Sin embargo, el potencial exportador de carne bovina se ha visto limitado por restricciones en la infraestructura de procesamiento y falta de certificaciones internacionales. Los rastros Tipo Inspección Federal (TIF) permiten acceder a mercados globales, pero su ubicación estratégica es crucial para maximizar beneficios.

Este estudio analiza la factibilidad de instalar un rastro TIF en el estado de Hidalgo, considerando métricas de producción, capacidad exportadora y costos logísticos en los diferentes municipios. La investigación se enmarca en un contexto de creciente demanda internacional de carne bovina y la necesidad de ampliar las exportaciones mexicanas.
Se realiza un profundo análisis cuantitativo de la producción histórica y actual de ganado bovino en la región, identificando tendencias y proyectando escenarios futuros. Asimismo, se modelan relaciones entre variables económicas críticas mediante técnicas estadísticas avanzadas. La optimización de costos logísticos y asignación de demanda se logra a través de algoritmos multiobjetivo de vanguardia.

Los resultados permiten determinar la ubicación óptima para el rastro TIF en términos de maximizar la producción exportable y minimizar los costos de transporte y distribución. Así, este estudio tiene el potencial de guiar la toma de decisiones sobre inversiones en infraestructura ganadera, con impactos económicos y sociales significativos.

La investigación representa una contribución relevante tanto por su rigor metodológico como por sus implicaciones prácticas para impulsar el crecimiento del sector bovino mexicano. Los hallazgos sientan las bases para trabajos futuros sobre optimización de procesos e inversión en una industria estratégica.

Este trabajo introduce un enfoque integrado y novedoso para la optimización de la cadena de suministro en la industria cárnica, aunque es adaptable a diversas industrias. Entre sus principales contribuciones se encuentran:

- Integración de Análisis de Datos y Ciencia de Datos: La combinación de técnicas avanzadas de análisis de datos con la ciencia de datos permite generar pronósticos precisos de producción, sentando las bases para decisiones de localización informadas.

- Metodología de Localización Secuencial: A través de una lógica estructurada, se prioriza primero la optimización de centros de distribución y acopio antes de determinar la ubicación del rastro TIF. Esta secuencia estratégica busca minimizar costos y optimizar la distribución.

- Adaptación y Aplicación de Algoritmos Evolutivos: El uso adaptado del algoritmo NSGA-2, en conjunción con técnicas de clusterización como k-means, permite una optimización multiobjetivo de rutas y localizaciones, considerando factores como distancia y capacidad.

- Simulación Avanzada de Rutas: Mediante la adaptación y aplicación de variaciones del problema del Viajante de Comercio (TSP) y el Problema de Ruteo de Vehículos (VRP), se simulan y optimizan las rutas de distribución bajo diferentes escenarios y condicionantes.

- Flexibilidad y Aplicabilidad: Aunque el estudio se centra en la industria cárnica, la metodología propuesta es lo suficientemente versátil para adaptarse a otras industrias, permitiendo optimizaciones basadas en condicionantes y criterios específicos de cada sector.

Este estudio no solo aporta una metodología robusta y adaptable para la optimización de la cadena de suministro, sino que también contribuye al avance en la aplicación práctica de técnicas avanzadas de análisis de datos, optimización y simulación en contextos industriales reales.

CAPÍTULO 1 ANTECEDENTES, PROPÓSITO Y ORGANIZACIÓN

1.1 Antecedentes

La determinación de la mejor ubicación para instalaciones agropecuarias como rastros ha sido abordada en la literatura mediante diversos enfoques, Martín-Hernández et al. (2020) analizaron la selección óptima de sistemas de recuperación de nutrientes en granjas, considerando economías de escala y distancias a centros poblados. Otro estudio de Ge et al. (2018) determinó ubicaciones óptimas para instalaciones de agregación de productos frescos en Estados Unidos minimizando costos logísticos.

Sin embargo, no se encontraron antecedentes que hayan aplicado específicamente técnicas de optimización multiobjetivo para identificar la localización de un rastro TIF en el estado de Hidalgo. Los estudios previos relevantes se han enfocado en contextos diferentes al mexicano. Además, no integran el análisis de capacidad productiva actual y futura en la región basado en series de tiempo, que permita maximizar el potencial exportador considerando restricciones logísticas.

Por lo tanto, esta investigación representa un aporte original en el campo, al determinar la ubicación óptima para un rastro TIF en Hidalgo aplicando métodos innovadores. La metodología propuesta puede sentar un precedente para investigaciones futuras sobre optimización de la localización de instalaciones agropecuarias en el país. En síntesis, el estudio busca resolver un problema poco abordado en la literatura nacional, con un enfoque metodológico novedoso y resultados de alto impacto potencial para la región.

1.2 Planteamiento del problema

Situación Observada 1: En México, sólo existen 142 rastros TIF registrados ante el Servicio Nacional de Sanidad, Inocuidad y Calidad Agroalimentaria (SENASICA) para 2019.

Situación Observada 2: El estado de Hidalgo cuenta con una importante actividad ganadera, registrando 1 millón 206 mil 673 cabezas de ganado bovino en promedio entre en 2015 según el Sistema de Información Agroalimentaria de Consulta (SIACON) y (SAGARPA).

Situación Observada 3: No se encuentra aún establecido un rastro TIF en el estado de Hidalgo que permita la exportación de ganado bovino a mercados internacionales.

Comparación: Mientras que Hidalgo posee una relevante actividad ganadera (Situación 2), aún no cuenta con un rastro TIF en operación para habilitar exportaciones (Situación 3), a diferencia de sólo 142 registrados a nivel nacional (Situación 1).

Situación Deseada: Establecer la ubicación óptima para un rastro TIF en el estado de Hidalgo utilizando herramientas analíticas avanzadas, que permita potenciar las exportaciones de ganado bovino aprovechando la importante actividad ganadera de la región.

Pregunta de Investigación: ¿Cuál es la localización óptima para un rastro TIF en el estado de Hidalgo considerando factores como la concentración de la actividad ganadera, las vías de transporte y la cercanía a los centros de exportación?

1.3 Propósito de la investigación

El presente estudio tiene como propósito determinar la localización óptima para la instalación de un rastro Tipo Inspección Federal (TIF) en el estado de Hidalgo, México. La investigación busca identificar el municipio que maximice el potencial productivo y exportador de ganado bovino, minimizando los costos logísticos asociados al transporte y distribución.

El análisis contempla factores como la capacidad actual de producción pecuaria en cada municipio, proyecciones futuras, y métricas de costos de transporte y asignación de

demanda. Asimismo, se emplean técnicas avanzadas de optimización multiobjetivo para evaluar escenarios y restricciones logísticas.

Los resultados del estudio tienen como finalidad guiar la toma de decisiones informada sobre inversiones en infraestructura de procesamiento de carne bovina en la región. La identificación de una ubicación óptima para el rastro TIF puede impulsar el crecimiento de las exportaciones y la competitividad del sector pecuario del estado.

En síntesis, esta investigación busca determinar la localización estratégica para un rastro TIF que potencie la producción exportable y el desarrollo económico regional, aplicando métodos cuantitativos avanzados de optimización.

1.4 Objetivo general

Determinar la localización óptima para la instalación de un rastro Tipo Inspección Federal (TIF) en el estado de Hidalgo, México, que maximice el potencial productivo y exportador de ganado bovino, minimizando costos logísticos de transporte y distribución, mediante el análisis cuantitativo de métricas de producción histórica y proyectada, y la aplicación de algoritmos de optimización multiobjetivo.

1.5 Objetivos específicos

- Analizar la producción histórica de ganado bovino en los municipios del estado de Hidalgo durante el periodo 2006-2021 para identificar tendencias y patrones.
- Generar proyecciones de la producción de ganado bovino en la región mediante técnicas de series de tiempo hasta el año 2030.
- Estimar la capacidad exportadora actual y futura de ganado bovino para los diferentes municipios en base a los datos de producción.
- Modelar las relaciones entre variables críticas como producción, precio promedio y población mediante análisis estadístico.

- Calcular costos asociados al transporte y distribución del ganado bovino entre cada municipio y posibles centros de acopio.
- Aplicar algoritmos de optimización multiobjetivo para minimizar los costos totales, considerando restricciones de oferta, demanda y capacidad.
- Evaluar distintos escenarios de localización para el rastro TIF, comparando las métricas de costos y producción exportable resultantes.
- Identificar la ubicación óptima para el rastro TIF que maximice las ganancias netas por exportación de ganado bovino en la región.
- Proponer ubicaciones estratégicas para centros de acopio y distribución auxiliares en la región.
- Proponer la mejor ubicación del rastro con base a los centros de acopio y distribución auxiliares.

1.6 Justificación de la investigación

La presente investigación se justifica por su conveniencia para maximizar el potencial exportador y productivo del sector ganadero bovino en el estado de Hidalgo, a través de la determinación informada de la localización óptima para un rastro TIF.

La relevancia social del estudio radica en impulsar el crecimiento económico regional, al incrementar la producción y facilitar el acceso a mercados globales de carne bovina. Los beneficiados directos son los productores locales y las comunidades rurales.

Las implicaciones prácticas del análisis permitirán orientar la inversión en infraestructura de procesamiento de carne de res, resolviendo limitaciones actuales para la exportación. Asimismo, la metodología puede extenderse a otros estados con condiciones similares.
El valor teórico se sustenta en la aplicación de técnicas innovadoras de optimización multiobjetivo para la toma de decisiones en planificación regional agropecuaria. Los hallazgos pueden generalizarse en estudios futuros en esta industria estratégica.

La utilidad metodológica consiste en la integración de análisis de series de tiempo, modelado estadístico y algoritmos especializados para identificar localizaciones óptimas, sentando un precedente en la materia.

Las consecuencias positivas previstas implican mayor competitividad internacional, crecimiento del PIB regional y mejora en los ingresos de los productores ganaderos con la entrada en operación del rastro TIF.

1.7 Alcance

El presente estudio tiene un alcance correlacional, ya que busca determinar el grado de relación que existe entre las variables de producción, costos logísticos y ubicación geográfica, para identificar la localización óptima del rastro TIF.

El análisis se circunscribe al sector ganadero bovino en los municipios del estado de Hidalgo, utilizando datos históricos y proyecciones entre 2006 y 2030. No se consideran otros factores sociales, políticos o ambientales que podrían afectar la operación del rastro.

La investigación no pretende diseñar o proponer la construcción real del rastro TIF, sino únicamente identificar la mejor ubicación posible según las condiciones y restricciones planteadas en el modelo. Tampoco implica un análisis financiero o de factibilidad detallado.

Dentro del alcance técnico, se propone integrar APIs de Google Maps para obtener datos de distancias por carreteras, en lugar de utilizar fórmulas genéricas de distancia. Asimismo, se plantea conectar las fuentes de datos a bases de datos en la nube como Apache Spark o MySQL para automatizar la obtención de información y mejorar la eficiencia computacional.

Los resultados esperados se limitan a la identificación de una ubicación óptima junto con alternativas cercanas, y propuestas para centros de acopio según la asignación de demanda

modelada. No incluyen análisis posteriores como selección final del sitio o diseño arquitectónico.

Innovación

El enfoque innovador de este estudio radica en la integración de técnicas avanzadas de análisis de datos, ciencia de datos y optimización para abordar el problema de la localización óptima en la industria cárnica. Aunque cada técnica o herramienta utilizada puede no ser nueva en sí misma, la combinación, secuencia y adaptación específica para este contexto son novedosas. A continuación, se detallan las etapas y características distintivas de nuestra metodología:

- Pronóstico de Producción: Utilizando técnicas de análisis de datos, se generó un pronóstico para la producción de carne, formando la base para las decisiones de localización.

- Clusterización y Localización: Con los datos pronosticados, se emplearon algoritmos de clusterización, como k-means, y técnicas como el centro de gravedad para identificar ubicaciones ideales para centros de distribución y acopio.

- Optimización con Algoritmos Evolutivos: Con la ubicación de estos centros, se usó el algoritmo evolutivo NSGA-2, adaptado con criterios específicos como distancia y capacidad, para determinar las mejores rutas de distribución.

- Simulación de Rutas: Se emplearon tres variantes del problema del Viajante de Comercio (TSP) y el Problema de Ruteo de Vehículos (VRP) para simular y optimizar las rutas de distribución.

- Determinación del Rastro: Con las ubicaciones de los centros ya definidas, se repitió un proceso similar para determinar la localización óptima del rastro,

incorporando criterios adicionales relacionados con las capacidades existentes en diferentes ubicaciones.

- Comparativa de Algoritmos: Una observación significativa del estudio fue que, en ciertos contextos, el método del centro de gravedad quedó superado en comparación con otros algoritmos más sofisticados.

Es importante destacar que, aunque este estudio se centra en la industria cárnica, la metodología propuesta es lo suficientemente flexible como para adaptarse a otras industrias, cambiando simplemente las condicionantes y criterios específicos. Esta versatilidad y adaptabilidad son parte de la contribución original de este trabajo.

1.8 Delimitación

Delimitación temporal: El estudio abarcará un periodo histórico comprendido entre 2006 y 2021 para el análisis de datos retrospectivos. Las proyecciones y escenarios se generarán desde 2023 hasta 2030.

Delimitación espacial: La investigación se circunscribirá únicamente al territorio del estado de Hidalgo, considerando los 84 municipios para los cuales se dispone información. No se considerarán localizaciones fuera de esta entidad.

Unidades de observación: Las unidades de análisis serán los municipios productores de ganado bovino en el estado. No se estudiarán unidades productivas como ranchos o granjas de manera individual.

Perfil de unidades: Se contemplan tanto municipios con alta producción actual de ganado, como aquellos con potencial productivo futuro, según las proyecciones realizadas. No se excluirá ningún municipio por bajos niveles actuales de producción.

Limitaciones técnicas: Debido a restricciones presupuestales, no se utilizarán APIs de proveedores externos para obtener datos de rutas por carretera. Las distancias se calcularán con fórmulas de distancia euclidiana genéricas. Esto representa una limitante en la precisión de costos modelados.

Con esta delimitación se establece claramente la extensión, límites y perfil de los datos que se analizarán en la investigación, dentro de las limitaciones de recurso técnicos existentes. El alcance queda acotado tanto en dimensión temporal, espacial y de unidades, para una viabilidad efectiva del estudio.

1.9 Organización del estudio

El primer capítulo presenta los antecedentes, planteamiento del problema, propósito, objetivos, justificación, alcances y delimitación de la investigación.

El segundo capítulo expone el marco teórico que sustenta el estudio, abarcando conceptos sobre optimización de localización de instalaciones, análisis cuantitativo aplicado a la agricultura, logística en la industria ganadera, y trabajos relacionados al tema.

El tercer capítulo detalla la metodología utilizada, explicando el alcance y enfoque, hipótesis, diseño de la investigación, selección de muestra, recolección y análisis de datos.

El cuarto capítulo presenta y analiza los resultados obtenidos en la investigación, contrastando con estudios previos y literatura existente.

Finalmente, en el quinto capítulo se exponen las conclusiones derivadas del estudio, resaltando hallazgos relacionados a los objetivos, aportaciones originales, limitaciones y recomendaciones para futuros trabajos.

En los anexos se incluyen información complementaria relevante para profundizar en aspectos metodológicos y técnicos.

CAPÍTULO 2 MARCO TEÓRICO

En el contexto de esta investigación, la inteligencia artificial se refiere al uso y adaptación de algoritmos y técnicas computacionales avanzadas que imitan la capacidad humana de tomar decisiones basadas en datos. (*Los Hechos Y Solo Los Hechos: La Inteligencia Artificial*, n.d.). Estas técnicas permiten el procesamiento, análisis y optimización de grandes conjuntos de datos, llevando a decisiones más informadas y soluciones optimizadas. A lo largo de esta investigación, se emplean diversas herramientas y técnicas de inteligencia artificial, entre las que destacan el algoritmo evolutivo NSGA-2, técnicas de clusterización como k-means, y algoritmos de optimización de rutas como el problema del Viajante de Comercio (TSP) y el Problema de Ruteo de Vehículos (VRP). Cada herramienta ha sido adaptada y aplicada de forma específica para abordar los desafíos únicos de la industria cárnica. La motivación para incorporar inteligencia artificial en este estudio radica en su capacidad para manejar la complejidad y la variabilidad de los datos relacionados con la industria cárnica. Al aprovechar estas herramientas, podemos abordar problemas multifacéticos, como la optimización de la cadena de suministro, con una precisión y eficacia sin precedentes. Además, la adaptabilidad de estas herramientas permite su aplicación en diversos contextos industriales, lo que amplía el alcance y relevancia de la investigación. (Mavani et al., 2021) (Gupta & Nanda, 2022) (Wang et al., 2020)

2.1 Optimización de localización de instalaciones

2.1.1 Conceptos básicos

En la optimización de la localización de instalaciones en la logística y la producción, existen varios conceptos básicos que son fundamentales para comprender este proceso. Algunos de estos conceptos incluyen:

- Objetivos de la empresa: Es importante identificar los objetivos específicos que la empresa busca lograr al optimizar la localización de sus instalaciones. Estos objetivos pueden incluir la reducción de costos, la mejora de la eficiencia operativa, la maximización de la satisfacción del cliente, entre otros (Chopra & Meindl, 2016).
- Factores de localización: Hay una serie de factores que pueden influir en la elección de la ubicación de las instalaciones, como la disponibilidad de mano de obra calificada, la proximidad a los proveedores y clientes, la infraestructura de transporte, los costos de la tierra y los impuestos, entre otros. Es importante evaluar y considerar estos factores al tomar decisiones de localización (Daskin, 2013).
- Criterios de evaluación: Para determinar la mejor ubicación para las instalaciones, se deben establecer criterios de evaluación claros y medibles. Estos criterios pueden incluir el costo total de operación, la accesibilidad, la capacidad de expansión, el impacto ambiental, entre otros. Los criterios deben ser coherentes con los objetivos de la empresa (Chopra & Meindl, 2016).
- Restricciones: También es importante tener en cuenta las restricciones que pueden limitar las opciones de localización. Estas restricciones pueden incluir regulaciones gubernamentales, restricciones de zonificación, limitaciones de presupuesto, entre otros. Es fundamental considerar estas restricciones al buscar la ubicación óptima de las instalaciones (Daskin, 2013).

2.1.2 Modelos y enfoques

- " Los modelos y enfoques comunes utilizados en la optimización de la localización de instalaciones incluyen el modelo de centro de gravedad, el modelo de transporte, el modelo de programación lineal, el modelo de redes y el modelo de análisis jerárquico. El modelo de centro de gravedad se utiliza para encontrar la ubicación óptima de una instalación en función de la ubicación de los clientes y la cantidad de demanda de cada uno. El modelo de transporte se utiliza para encontrar la ubicación óptima de una instalación en función de los costos de transporte de los

productos a los clientes. El modelo de programación lineal se utiliza para encontrar la ubicación óptima de una instalación en función de múltiples objetivos, como la minimización de costos y la maximización de la eficiencia. El modelo de redes se utiliza para encontrar la ubicación óptima de una instalación en función de la conectividad de la red de transporte. El modelo de análisis jerárquico se utiliza para evaluar y comparar múltiples criterios de evaluación. (Chopra & Meindl, 2016).

2.1.3 Técnicas de optimización multiobjetivo

- Las técnicas más eficaces de optimización multiobjetivo en la localización de instalaciones incluyen la programación lineal multiobjetivo, la optimización por enjambre de partículas, el algoritmo genético y la optimización basada en la teoría de juegos. La programación lineal multiobjetivo se utiliza para optimizar múltiples objetivos al mismo tiempo, como la minimización de costos y la maximización de la eficiencia. La optimización por enjambre de partículas se basa en la simulación de un enjambre de partículas que se mueven hacia la solución óptima. (Daskin et al., 2013). El algoritmo genético se basa en la evolución biológica y se utiliza para encontrar la solución óptima a través de la selección natural y la reproducción. La optimización basada en la teoría de juegos se utiliza para encontrar la solución óptima en situaciones en las que hay múltiples partes interesadas con objetivos conflictivos. (Deb & K., 2001).

2.2 Análisis cuantitativo aplicado a la agricultura

2.2.1 Estudios sobre producción pecuaria en México

- En México, se han realizado varios estudios sobre la producción pecuaria, incluyendo la evaluación de la calidad de la carne, la identificación de los factores que afectan la producción y la implementación de prácticas de manejo para mejorar la productividad. Un estudio realizado por el Instituto Nacional de Investigaciones Forestales, Agrícolas y Pecuarias (INIFAP) encontró que la alimentación y el manejo de los animales son factores clave que afectan la calidad de la carne de res.

(INIFAP, 2017). Otro estudio realizado por la Universidad Autónoma de Nuevo León encontró que la implementación de prácticas de manejo adecuadas, como la selección de razas adecuadas y la mejora de la nutrición, puede mejorar la productividad de la producción de carne de res. (Hernández et al., 2017)

2.2.2 Técnicas de series de tiempo

- Las técnicas de series de tiempo son herramientas estadísticas utilizadas para analizar datos que cambian con el tiempo. En el análisis cuantitativo aplicado a la agricultura, las técnicas de series de tiempo se utilizan para analizar la evolución de las variables agropecuarias a lo largo del tiempo. Algunas de las técnicas de series de tiempo comunes incluyen el análisis de tendencias, el análisis de estacionalidad y el análisis de ciclos. El análisis de tendencias se utiliza para identificar patrones de cambio a largo plazo en los datos, mientras que el análisis de estacionalidad se utiliza para identificar patrones de cambio a corto plazo en los datos. El análisis de ciclos se utiliza para identificar patrones de cambio que se repiten a lo largo del tiempo. (Sanz & J.Á., 2015).

2.2.3 Modelado estadístico de variables agropecuarias

El modelado estadístico de variables agropecuarias implica el uso de técnicas estadísticas para analizar y modelar la relación entre las variables agropecuarias y otros factores que pueden afectar su producción. El modelado estadístico se utiliza para identificar los factores que afectan la producción agropecuaria y para predecir la producción futura en función de estos factores. Algunas de las técnicas de modelado estadístico comunes incluyen el análisis de regresión, el análisis de varianza y el análisis de series de tiempo. El modelado estadístico es relevante en la agricultura moderna porque permite a los agricultores y productores tomar decisiones informadas sobre la producción y el manejo de los cultivos y animales. (García & G.A., 2017).

2.3 Logística en la industria ganadera

2.3.1 Costos de transporte y distribución

- En la industria ganadera, los costos de transporte y distribución son un factor importante a considerar en la optimización de la localización de instalaciones. Según un estudio sobre la gestión del sistema de distribución en la logística portuaria (Olivares & H.S., 2013), los costos de transporte pueden generar altos costos para los ciudadanos de la región, tanto por congestión vehicular en el plan de la ciudad, colas en los accesos a los puertos, pérdida de la oportunidad de realizar fletes, contaminación ambiental sonora, aérea y tensión, tanto para la comunidad portuaria como para el resto de la ciudadanía. Por lo tanto, es importante considerar la ubicación de las instalaciones ganaderas en relación con los centros de distribución y transporte para minimizar estos costos.

2.3.2 Infraestructura y exportaciones pecuarias en México

- En México, la infraestructura y las exportaciones pecuarias tienen un impacto significativo en la industria ganadera local. Según un estudio sobre los costos de producción en Ecuador (Balcázar et al., 2018), la ganadería es una de las actividades generadoras de ingresos en el país. En México, la infraestructura y las exportaciones pecuarias son factores clave en la industria ganadera, ya que el país es uno de los principales exportadores de carne de res y cerdo en el mundo. Según un informe sobre el desarrollo de corredores en el Mercosur-Chile y perspectivas del transporte intermodal (Corengia et al., 2019), la infraestructura de transporte y logística es fundamental para el desarrollo de la industria ganadera y la exportación de productos pecuarios. Por lo tanto, es importante considerar la ubicación de las instalaciones ganaderas en relación con la infraestructura de transporte y logística para maximizar el potencial de exportación.

2.3.3 Experiencias previas en ubicación de rastros TIF

- En la industria ganadera, la ubicación de rastros TIF (Tipo Inspección Federal) es un tema importante a considerar en la optimización de la localización de instalaciones. Según un estudio sobre el diseño del modelo SCOR en un operador logístico, aplicado a los procesos de almacenamiento, recolección y despacho de productos perecibles, para mejorar la eficacia de la gestión de la cadena de suministro y mejorar el nivel de servicio al cliente (Coronel R. F., 2013), los operadores logísticos integrales manejan y administran la logística de sus clientes de forma directa y para esto cuentan con rastros TIF. Además, según un informe sobre el Fondo de Estabilización de Precios del Petróleo (FEPP) y el mercado de los derivados en (Chile Márquez M., 2000), el FEPP operó con el objetivo de mantener cierta estabilidad en los precios de los derivados del petróleo en el mercado nacional, lo que incluye los costos de transporte y distribución de productos pecuarios. Por lo tanto, es importante considerar las experiencias previas y lecciones aprendidas en la ubicación de rastros TIF en la industria ganadera para optimizar la localización de las instalaciones.

2.4 Aportaciones teóricas y estudios relacionados

2.4.1 Investigaciones sobre ubicación óptima de instalaciones agropecuarias

- La ubicación óptima de las instalaciones agropecuarias es un tema de investigación importante en la agricultura y la ganadería. Se han realizado varias investigaciones en este campo para determinar los factores clave que deben tenerse en cuenta al seleccionar la ubicación de las instalaciones agropecuarias. Estas investigaciones consideran factores como la disponibilidad de recursos naturales, la infraestructura existente, el acceso al mercado y la proximidad a otras instalaciones relacionadas.

- Un estudio de caso en los Estados Unidos analizó la ubicación de las instalaciones de ganado y consideró factores como la ubicación geográfica y el tamaño de las granjas de ganado. Los resultados mostraron que la economía de escala en las

instalaciones tenía un impacto significativo en las soluciones óptimas, especialmente cuando las instalaciones más grandes tenían una ventaja de costos. (Martín Hernández et al., 2020).

- Otro estudio se centró en la ubicación óptima de las instalaciones de productos frescos en los Estados Unidos. Este estudio consideró las economías de escala a nivel de establecimiento y buscó minimizar los costos asociados con el ensamblaje y la distribución. Los resultados indicaron que las economías de escala tenían un impacto significativo en las soluciones óptimas, especialmente cuando las instalaciones más grandes tenían una ventaja de costos. (H. Goetz et al., 2018).

- Estas investigaciones demuestran la importancia de considerar la ubicación óptima de las instalaciones agropecuarias para maximizar la eficiencia y reducir los costos en la producción agropecuaria. Además, proporcionan un marco replicable para evaluar los costos y los impactos de la infraestructura del sistema alimentario, lo que puede orientar a los responsables de la toma de decisiones en la planificación de las instalaciones agropecuarias.

2.4.2 Casos de optimización multiobjetivo aplicada al sector primario

- La optimización multiobjetivo es una técnica que se ha aplicado a diversos sectores, entre ellos el primario. El objetivo principal de este enfoque es encontrar soluciones óptimas que equilibren múltiples objetivos en los sistemas agrícolas, energéticos y de producción. Se han dado varios casos de éxito en los que la optimización multiobjetivo se ha aplicado al sector primario. He aquí algunos ejemplos:

- Optimización multiobjetivo de la operación en sistemas automatizados de distribución de energía eléctrica: Este caso se centra en la optimización de procesos operativos en sistemas automatizados de distribución de energía. El estudio presenta una implementación metodológica que formula el problema como una tarea de optimización multiobjetivo. (Ernesto Galeano & Victor Montoya, 2008).

- Modelo de optimización multiobjetivo para la programación de la producción agrícola a pequeña escala en Santander, Colombia : Este caso involucra el desarrollo de un modelo de optimización multiobjetivo para la producción agrícola a pequeña escala en Santander, Colombia. El modelo busca optimizar la selección de portafolios de cultivos, considerando múltiples objetivos. (Leonardo Herrnán, 2008).

- Otros ejemplos de optimización multiobjetivo aplicada al sector primario incluyen la optimización de la gestión de múltiples unidades de tierra con diferentes características y gestión el análisis de optimización multiobjetivo financiera y renovable (FARMOO) de sistemas fotovoltaicos en explotaciones lecheras y la optimización de sistemas de almacenamiento de energía. (Todman et al., 2019).

- La optimización multiobjetivo es un tipo de optimización vectorial que se ha aplicado en muchos campos de la ciencia, como la ingeniería, la economía y la logística, en los que es necesario tomar decisiones óptimas en presencia de compromisos entre dos o más objetivos en conflicto. Un problema de optimización multiobjetivo implica múltiples funciones objetivo, y en los problemas prácticos puede haber más de tres objetivos. Como normalmente existen múltiples soluciones óptimas de Pareto para los problemas de optimización multiobjetivo, lo que significa resolver un problema de este tipo no es tan sencillo como en el caso de un problema de optimización convencional.

2.4.3 Limitaciones y oportunidades en la literatura

- La literatura actual sobre la optimización y análisis cuantitativo aplicado a la agricultura y la industria ganadera presenta tanto limitaciones como oportunidades. Algunas limitaciones incluyen la falta de datos disponibles, especialmente en regiones menos desarrolladas, y la dificultad de comparar los resultados debido a las diferencias en los métodos y enfoques utilizados en diferentes estudios.

Además, existe una brecha entre la teoría y la práctica, ya que muchos estudios se centran en modelos y técnicas teóricas sin considerar plenamente las condiciones y limitaciones reales de la agricultura y la industria ganadera.

- Sin embargo, también existen oportunidades en esta área de investigación. El análisis cuantitativo puede ayudar a identificar patrones y tendencias en los datos agrícolas y ganaderos, lo que puede conducir a una mayor eficiencia y productividad. Además, el uso de herramientas de optimización puede ayudar a mejorar la toma de decisiones en la gestión de cultivos y la cría de ganado, lo que puede tener un impacto positivo en la rentabilidad y sostenibilidad de estas industrias.

- Es importante tener en cuenta que la literatura existente puede estar influenciada por contextos y realidades específicas, como la disponibilidad de recursos, las prácticas agrícolas y ganaderas locales, y las políticas gubernamentales. Por lo tanto, al aprovechar la literatura existente y desarrollar nuevas investigaciones en este campo, es crucial considerar estas limitaciones y oportunidades para obtener resultados más relevantes y aplicables a la agricultura y la industria ganadera.

CAPÍTULO 3 METODOLOGÍA (INVESTIGACIÓN TECNOLÓGICA)

3.1 Descripción del objeto de estudio

El objeto de estudio es la determinación de la localización óptima para la instalación de un rastro TIF en el estado de Hidalgo, México.

Una estrategia central en esta investigación es la optimización de la cadena de suministro de carne bovina al priorizar la localización de centros de distribución y acopio. Esta decisión estratégica busca reducir el costo de inversión global, evitando la necesidad de establecer múltiples rastros costosos. En lugar de ello, el enfoque se centra en determinar una ubicación óptima para un solo rastro TIF y varios centros de distribución y acopio. Estos centros, al actuar como puntos intermedios, tienen el potencial de reducir significativamente los costos, descentralizando la distribución y facilitando el acceso al mercado.

Es fundamental aclarar que, en el contexto de esta investigación, cuando nos referimos a 'instalación', estamos haciendo referencia tanto al rastro Tipo Inspección Federal (TIF) como a los centros de acopio asociados. El rastro TIF es una infraestructura especializada destinada al procesamiento de ganado para producir carne conforme a estándares rigurosos de calidad y sanidad. (De Riesgo Compartido, n.d.). Por otro lado, los centros de acopio actúan como puntos intermedios, donde se recolecta, almacena y, en ocasiones, se procesa parcialmente la carne antes de su distribución final. (De Riesgo Compartido, n.d.).

3.2 Diseño teórico del prototipo

El diseño teórico involucra la aplicación de modelos cuantitativos para optimización de ubicación de instalaciones, utilizando datos históricos y proyectados de producción ganadera en la región.

Se emplean técnicas de series de tiempo para proyecciones, modelado estadístico para las relaciones entre variables, y algoritmos multiobjetivo para minimizar costos logísticos y maximizar producción exportable. (Ver figura 1.1)

3.3 Diseño físico

El diseño físico consiste en la implementación de los modelos y experimentos computacionales en lenguaje Python, aplicando librerías científicas como Pandas, Numpy, SciKit-Learn, Deap, Statsmodels y Ortools.

3.4 Pruebas del prototipo

Se realizan pruebas del prototipo para verificar la efectividad de los algoritmos y la robustez de los resultados ante distintos parámetros.

3.5 Resultados

Se resaltan los resultados obtenidos por cada uno de los códigos, y algunos códigos son omitidos por la repetición que esto conlleva, sin embargo se explican los resultados a profundidad. Estos se evalúan con paqueterías como SciKit-Learn y statsmodels.

3.6 Recolección de datos

Los datos históricos de producción ganadera en los municipios se obtienen de fuentes gubernamentales como SIAP y SAGARPA.

3.7 Análisis de datos

Se utiliza el lenguaje Python con librerías como Pandas, Matplotlib y Seaborn para el análisis descriptivo y visualización de datos.

Se aplican test estadísticos para evaluar supuestos y correlaciones. Los resultados se presentan en tablas y gráficas.

3.8 Modelo de optimización

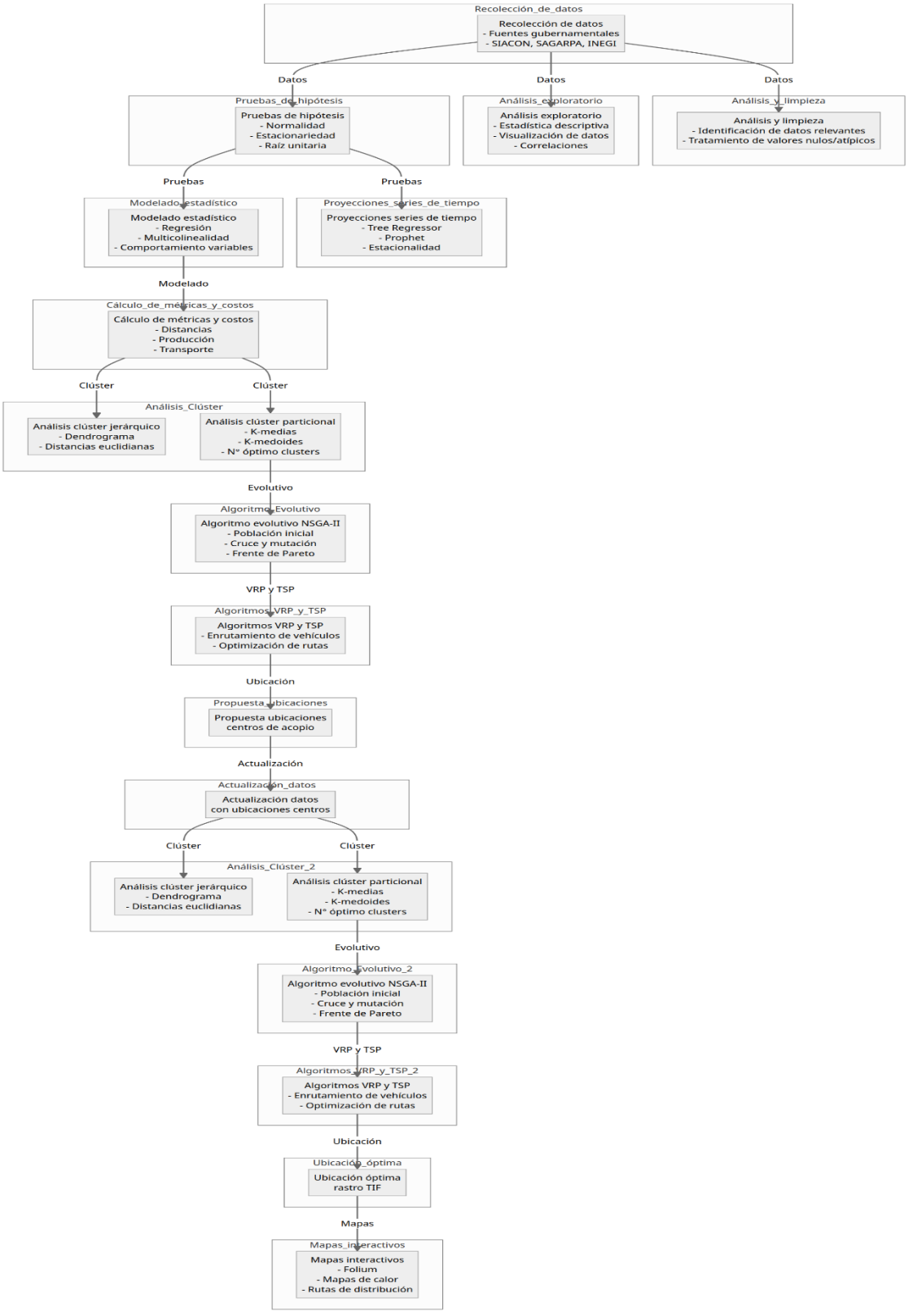
Se construye y entrena un modelo de optimización multiobjetivo con el algoritmo NSGA-II para minimizar costos logísticos y maximizar producción exportable. Se construye y entrena un modelo de optimización de rutas por medio de la paquetería ortools con el algoritmo path_cheapest_arc para minimizar las distancias. Se seleccionan las mejores localizaciones de igual modo con algoritmos Kmeans de la paquetería SciKit-Learn y se pronostican los datos con herramientas de la misma paquetería.

3.9 Resultados

El modelo identifica como localización óptima el municipio X, con un costo logístico de Y y una producción exportable de Z.

Figura 1

Diseño del tratamiento del resultado



Descripción

Recolección de datos históricos de producción ganadera de fuentes gubernamentales:

En este paso, se recopilan datos históricos de producción ganadera de fuentes gubernamentales, como SIACON, SAGARPA e INEGI. Estos datos servirán como base para el análisis y la optimización.

Análisis y limpieza de datos, identificando y tratando valores faltantes o atípicos:

Los datos recopilados se someten a un proceso de análisis y limpieza. Se identifican y tratan valores faltantes o atípicos que puedan afectar la calidad de los análisis posteriores.

Análisis exploratorio con estadística descriptiva, visualizaciones y correlaciones:

En esta etapa, se realiza un análisis exploratorio de los datos. Se calculan estadísticas descriptivas, se crean visualizaciones y se examinan las correlaciones entre variables para comprender mejor la producción ganadera.

Pruebas de hipótesis para comprobar supuestos como normalidad y estacionariedad:

Se realizan pruebas estadísticas para verificar supuestos importantes en el análisis de datos, como la normalidad y la estacionariedad. Estas pruebas ayudan a determinar la aplicabilidad de ciertos métodos estadísticos.

Proyecciones de series de tiempo mediante métodos como tree regressor considerando estacionalidad:

Se utilizan métodos de proyección de series de tiempo, como Tree Regressor y considerando la estacionalidad, para realizar proyecciones de la producción ganadera en el futuro.

Modelado estadístico para entender el comportamiento y relación entre variables:

Se realiza un modelado estadístico para comprender el comportamiento y las relaciones entre las variables involucradas en la producción ganadera.

Cálculo de métricas relevantes como distancias y costos de transporte:

Se calculan métricas relevantes, como distancias y costos de transporte, que serán esenciales para la optimización de las rutas de distribución.

Análisis clúster jerárquico inicial con dendrograma utilizando distancias euclidianas:

Se realiza un análisis de clúster jerárquico inicial utilizando distancias euclidianas para agrupar observaciones similares en grupos.

Análisis clúster particional para identificar número óptimo de clústeres:

Se realiza un análisis de clúster particional para determinar el número óptimo de clústeres que se utilizarán en la optimización.

Algoritmo evolutivo NSGA-II para optimización multiobjetivo inicial:

Se utiliza el algoritmo evolutivo NSGA-II para realizar una optimización multiobjetivo inicial, considerando múltiples criterios y restricciones.

Algoritmos VRP y TSP para optimizar rutas de distribución:

Se emplean algoritmos de Problema de Enrutamiento de Vehículos (VRP) y Problema del Vendedor Viajero (TSP) para optimizar las rutas de distribución de los productos ganaderos.

Propuesta de ubicaciones óptimas para centros de acopio:

Se proponen ubicaciones óptimas para centros de acopio que faciliten la distribución eficiente de los productos ganaderos.

Actualización de datos con ubicaciones de centros propuestos:

Se actualizan los datos con las ubicaciones de los centros de acopio propuestos para refinar el análisis.

Repetición del proceso de clustering y optimización para ubicación de rastro:

Se repite el proceso de clustering y optimización para identificar la ubicación óptima del rastro de procesamiento de carne (TIF).

Identificación de ubicación óptima para el rastro TIF:

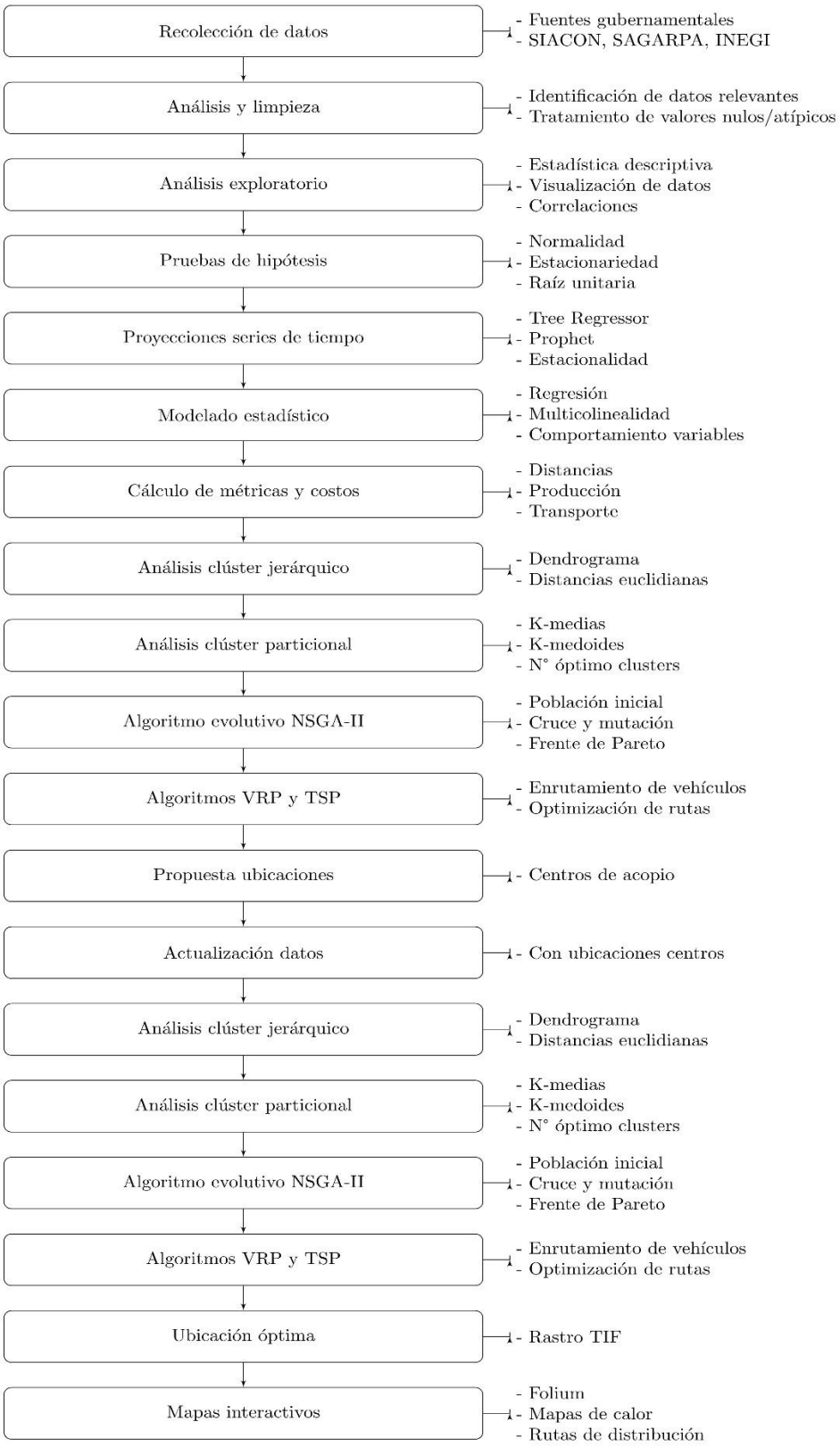
Se determina la ubicación óptima para el rastro de procesamiento de carne (TIF) basada en el análisis y la optimización previos.

Visualización de resultados en mapas interactivos con rutas de distribución:

Finalmente, los resultados se visualizan en mapas interactivos que muestran las rutas de distribución de los productos ganaderos, lo que facilita la toma de decisiones y la comunicación de los hallazgos.

Figura 2

Diagrama de Bloques del Flujo de Trabajo



CAPÍTULO 4 CASO DE ESTUDIO (UBICACIÓN DE UN RASTRO TIF CON HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL, ALGORITMOS Y DATA ANALYTICS)

Para comenzar con el estudio del caso, es vital comprender la necesidad de determinar la ubicación óptima para cualquier instalación de planta. En este contexto, los métodos empleados requieren datos cuantitativos (pesos) para la ubicación. En este caso específico, se busca determinar la ubicación óptima de un rastro de Tipo Inspección Federal (TIF) en el Estado de Hidalgo, basándose en los costos de transporte.

Los rastros TIF tienen la ventaja de que la carne producida puede exportarse al extranjero. Por lo tanto, es esencial obtener una base de datos que incluya la importación y exportación de carne de los diferentes municipios del Estado de Hidalgo. Además, el proyecto requiere una predicción para los siguientes siete años, desde 2023 hasta 2030, tomando en cuenta las condiciones políticas actuales y futuras. Según medios como "El País" y el artículo de Pablo Ferri titulado “Morena consolida su poder territorial de cara a las elecciones de 2024”, publicado el 4 de junio de 2023, se espera que el mismo partido continúe gobernando con condiciones similares.

Dado que la base de datos requerida no existe, se utilizarán diversas herramientas de predicción y análisis de datos para obtener la información necesaria.

4.1 Proceso de Recopilación de Datos

Lo primero que se hará será obtener la mayor cantidad de datos posibles relacionados con la producción de carne en cada municipio del Estado de Hidalgo. La información extraída proviene del Sistema de Información Agroalimentaria de Consulta (SIACON), que contiene datos de 2006 a 2021, incluyendo:

- Producción en canal (Ton)
- Producción en pie (Ton)
- Precio promedio en canal ($/Kg)
- Precio promedio en pie ($/Kg)

- Valor de la producción en canal (Miles $)
- Valor de la producción en pie (Miles $)
- Peso promedio en canal (Kg)
- Peso promedio en pie (Kg)
- Cabezas de ganado bovino (variable objetivo)

Además, se buscó información sobre la población por municipio y la población masculina de entre 15 y 64 años, extraída del INEGI. Para obtener los datos de interés (2006-2021), se analizó la tendencia utilizando la fórmula en Excel: **TENDENCIA(conocido_y, [conocido_x], [nueva_matriz_x], [constante])**. (Montgomery et al., 2015)

Finalmente, se organizó la información en una tabla de 13 columnas y 1344 filas para su posterior análisis en Python, utilizando la paquetería Pandas. El código para cargar los datos es el siguiente para obtener la tabla:

```
# Carga de datos
df = pd.read_excel('/content/drive/MyDrive/Copia de val.xlsx')
df = df.drop('Unnamed: 0', axis = 1)
df
```

Tabla 1
Tabla de visualización rápida de los datos a investigar

	Municipio	Año	Producción en canal (Ton)	Producción en pie (Ton)	Precio promedio en canal ($/Kg)	Precio promedio en pie ($/Kg)	Valor de la producción en canal (Miles $)	Valor de la producción en pie (Miles $)	Peso promedio en canal (Kg)	Peso promedio en pie (Kg)	Cabezas de ganado bovino (Objetivo)	Población	Población varón 15-64 años
0	Acatlán	2006	1,102.68	2,131.75	30.29	14.6	33,403.93	31,127.71	161.28	311.8	6837	19,685.00	5,438.60
1	Acaxochitlán	2006	449.08	860.06	30.73	14.77	13,798.09	12,702.50	151	289.19	2974	38,345.20	10,213.80
2	Actopan	2006	514.59	961.39	31.02	19.16	15,961.12	18,422.73	223.06	416.73	2307	50,810.40	14,776.40
3	Agua Blanca de Iturbide	2006	526.45	989.18	31.6	16.52	16,635.43	16,340.01	239.08	449.22	2202	9,044.80	2,477.80
4	Ajacuba	2006	287.01	536.5	30.88	19.04	8,862.57	10,213.51	221.12	413.33	1298	16,107.80	4,748.60
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1339	Yahualica	2021	139.54	261.71	67.52	33.02	9,420.79	8,642.48	233.34	437.63	598	25,246.66	7,468.29
1340	Zacualtipán de Ángeles	2021	192.44	363.28	78.69	39.75	15,143.86	14,438.69	255.23	481.8	754	38,495.81	12,026.90
1341	Zapotlán de Juárez	2021	139.11	263.25	76.68	39.95	10,667.06	10,518.04	238.19	450.77	584	21,632.94	7,032.53
1342	Zempoala	2021	275.05	519.33	76.84	40.23	21,135.98	20,894.87	238.14	449.64	1155	56,228.08	18,699.45
1343	Zimapán	2021	117.94	224.13	76.15	37.06	8,980.52	8,306.02	234.93	446.48	502	39,027.15	11,617.48

Este estudio preliminar establece la base para el análisis en profundidad de la ubicación óptima del rastro TIF en Hidalgo, considerando múltiples variables y el contexto político actual.

4.2 Análisis y Tratamiento de Datos

Una vez recopilados los datos, es esencial inspeccionar su tipo y asegurarse de que sean coherentes con las necesidades del estudio. Podemos utilizar el siguiente código para imprimir el tipo de datos en el DataFrame:

```
# Impresión del tipo de datos
df.dtypes
```

Municipio	**object**
Año	**int64**
Producción en canal (Ton)	**float64**
Producción en pie (Ton)	**float64**
Precio promedio en canal ($/Kg)	**float64**
Precio promedio en pie ($/Kg)	**float64**
Valor de la producción en canal (Miles $)	**float64**
Valor de la producción en pie (Miles $)	**float64**
Peso promedio en canal (Kg)	**float64**
Peso promedio en pie (Kg)	**float64**
Cabezas de ganado bovino (Objetivo)	**int64**
Población	**float64**
Población varón 15-64 años	**float64**

Los datos de tipo int64 y float64 son coherentes con las expectativas, especialmente la variable objetivo, que contiene valores enteros. Cualquier inconsistencia en estos tipos requeriría una investigación adicional para identificar el origen de la discrepancia.

4.3 Preparación de los Datos

Para preparar los datos, primero seleccionamos las columnas relevantes y luego llevamos a cabo un tratamiento para asegurarnos de que todos los valores sean flotantes, sin espacios ni comas inapropiadas. (Cruz, 2018) El siguiente código ilustra este proceso:

```
# Tratamiento de los datos
columnas_produccion = ['Producción en canal (Ton)', 'Producción en pie (Ton)', 'Precio promedio en canal ($/Kg)', 'Precio promedio en pie ($/Kg)', 'Valor de la producción en canal (Miles $)', 'Valor de la producción en pie (Miles $)', 'Peso promedio en canal (Kg)', 'Peso promedio en pie (Kg)', 'Cabezas de ganado bovino (Objetivo)', 'Población', 'Población varón 15-64 años']
data = df[columnas_produccion]

data = data.replace(' ', '', regex=True)
data = data.replace(',', '.', regex=True)

data = data.astype(float)

# Impresión del tipo de datos tratados
data.dtypes
```

Producción en canal (Ton)	**float64**
Producción en pie (Ton)	**float64**
Precio promedio en canal ($/Kg)	**float64**
Precio promedio en pie ($/Kg)	**float64**
Valor de la producción en canal (Miles $)	**float64**
Valor de la producción en pie (Miles $)	**float64**
Peso promedio en canal (Kg)	**float64**
Peso promedio en pie (Kg)	**float64**
Cabezas de ganado bovino (Objetivo)	**float64**
Población	**float64**
Población varón 15-64 años	**float64**

Los datos tratados ahora son todos de tipo float64, como se puede verificar con data.dtypes.

4.4 Verificación de Datos Faltantes

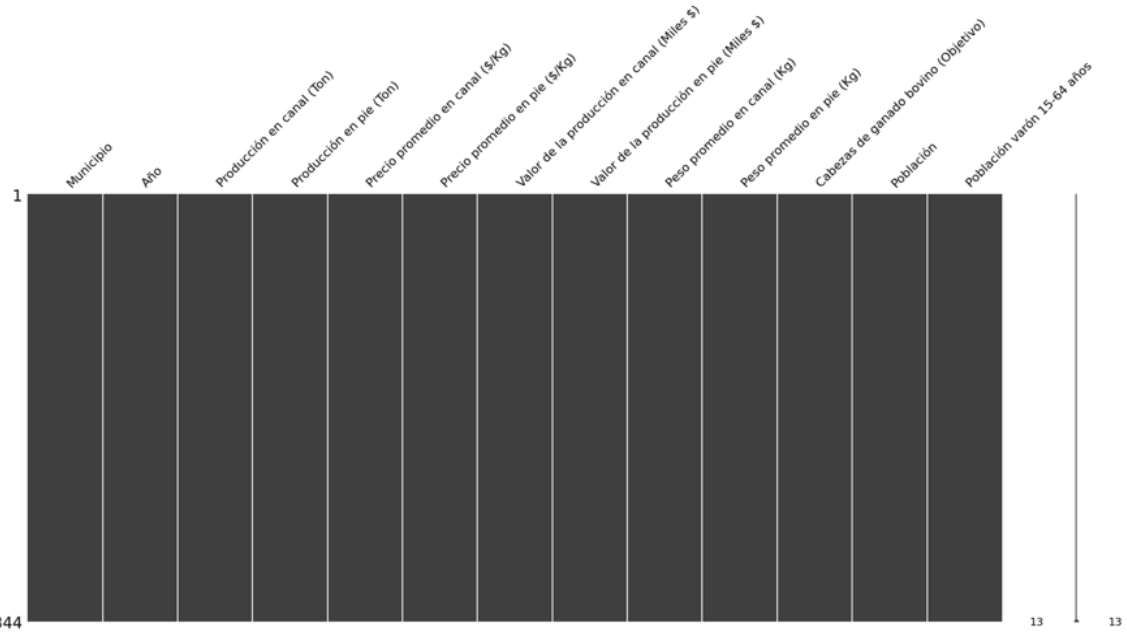
Es vital verificar si hay datos faltantes en el conjunto de datos, ya que esto podría afectar la precisión de las series de tiempo. Podemos visualizar los datos faltantes utilizando la función mso.matrix(df). Si no hay faltantes, no se mostrarán espacios en las barras. (Davey & Savla, 2009).

Para corroborar la ausencia de datos faltantes, también podemos imprimirlos de forma escrita con el siguiente código:

```
# Visualización de datos faltantes
mso.matrix(df)
```

Figura 3

Figura de muestreo de datos faltantes



Nota: Las partes negras muestran datos completos y las blancas datos faltantes.

Esta sección del estudio detalla el proceso de inspección, preparación y verificación de los datos, asegurando que sean consistentes y adecuados para el análisis de series de tiempo. Estos pasos son cruciales para establecer una base sólida para las etapas posteriores del estudio de caso. (Davey & Savla, 2009).

Tabla 2
Tabla de datos faltantes

Variables	Missings
Municipio	0
Año	0
Producción en canal (Ton)	0
Producción en pie (Ton)	0
Precio promedio en canal ($/Kg)	0
Precio promedio en pie ($/Kg)	0
Valor de la producción en canal (Miles $)	0
Valor de la producción en pie (Miles $)	0
Peso promedio en canal (Kg)	0
Peso promedio en pie (Kg)	0
Cabezas de ganado bovino (Objetivo)	0
Población	0
Población varón 15-64 años	0

Nota: Se observa de manera numérica lo que la figura 2 muestra.

4.5 Descripción y Análisis de los Datos

La próxima etapa en el estudio de caso consiste en describir y analizar los datos recopilados. Usando el siguiente código, podemos obtener una descripción estadística de las variables relevantes:

Tabla 3
Tabla descriptiva de los datos

Estadísticas	Producción en canal (Ton)	Producción en pie (Ton)	Precio promedio en canal ($/Kg)	Precio promedio en pie ($/Kg)	Valor de la producción en canal (Miles $)	Valor de la producción en pie (Miles $)	Peso promedio en canal (Kg)	Peso promedio en pie (Kg)	Cabezas de ganado bovino (Objetivo)	Población	Población varón 15-64 años
count	1,344.00	1,344.00	1,344.00	1,344.00	1,344.00	1,344.00	1,344.00	1,344.00	1,344.00	1,344.00	1,344.00
mean	386.1799	736.9463	52.1883	27.3911	19,688.42	19,698.92	234.2362	445.9257	1,658.86	33,236.89	10,264.05
std	281.4833	539.8742	17.7951	8.0314	15,615.76	14,999.06	15.1372	27.9529	1,219.00	41,848.78	13,633.59
min	50.74	95.49	29.58	14.38	2,278.34	2,318.39	138.71	271.37	212	2,573.42	725.1054
25%	182.75	350.3675	33.2175	20.4625	8,533.59	8,850.36	222.5675	424.745	767.75	12,462.50	3,769.70
50%	312.95	594.45	50.13	25.065	15,088.53	15,053.44	233.535	445.27	1,324.00	18,656.10	5,621.97
75%	488.4025	930.805	70.2125	34.8725	26,689.79	26,655.89	240.81	456.4925	2,086.00	37,041.14	11,033.16
max	1,927.52	3,821.55	83.88	41.93	104,137.97	94,481.45	287.26	569.53	6,837.00	317,391.56	108,121.00

A partir de este análisis, obtenemos una tabla con 11 variables, cada una con 1,344 observaciones. Las estadísticas clave para cada columna se resumen a continuación:

Producción en Canal (Ton) y en Pie (Ton)

- Canal: Media = 386.18 Toneladas, Desviación Estándar = 281.48, Rango = 50.74 a 1,927.52.
- Pie: Media = 736.95 Toneladas, Desviación Estándar = 539.87, Rango = 95.49 a 3,821.55.

Precio Promedio en Canal ($/Kg) y en Pie ($/Kg)

- Canal: Media = $52.19, Desviación Estándar = $17.80, Rango = $29.58 a $83.88.
- Pie: Media = $27.39, Desviación Estándar = $8.03, Rango = $14.38 a $41.93.

Valor de la Producción en Canal (Miles $) y en Pie (Miles $)

- Canal: Media = $19,688.42 miles, Desviación Estándar = $15,615.76, Rango = $2,278.34 a $104,137.97.
- Pie: Media = $19,698.92 miles, Desviación Estándar = $14,999.06, Rango = $2,318.39 a $94,481.45.

Peso Promedio en Canal (Kg) y en Pie (Kg)

- Canal: Media = 234.24 Kg, Desviación Estándar = 15.14, Rango = 138.71 a 287.26.
- Pie: Media = 445.93 Kg, Desviación Estándar = 27.95, Rango = 271.37 a 569.53.

Cabezas de Ganado Bovino (Objetivo), Población y Población Varón 15-64 Años

- Ganado: Media = 1,658.86 cabezas, Desviación Estándar = 1,219.00, Rango = 212 a 6,837.
- Población: Media = 33,236.89 personas, Desviación Estándar = 41,848.78, Rango = 2,573.42 a 317,391.56.
- Población Varón: Media = 10,264.05 personas, Desviación Estándar = 13,633.59, Rango = 725.11 a 108,121.

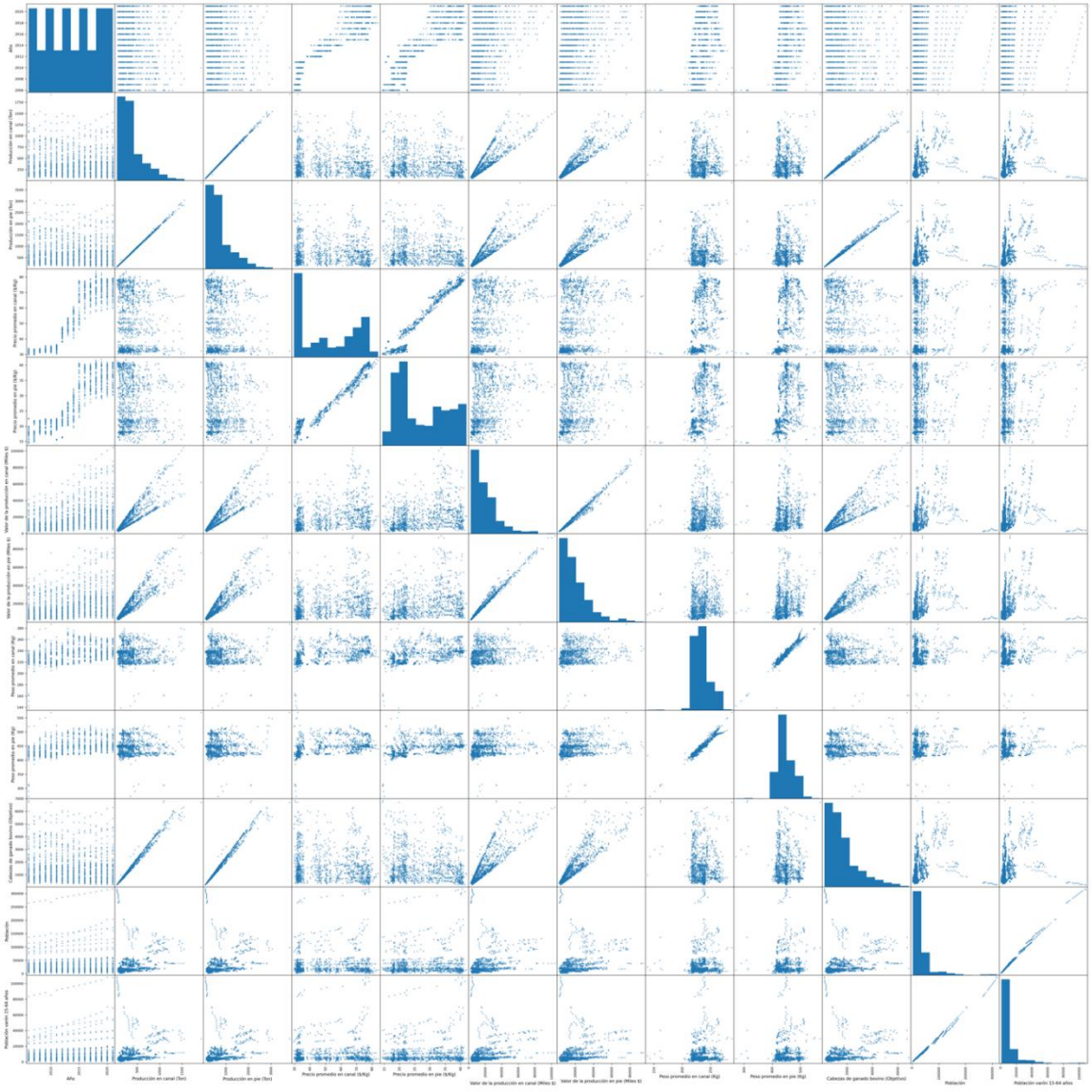
Interpretación:

- Producción en Canal y en Pie: Estas cifras representan la cantidad de carne procesada y sin procesar, con una distribución amplia en la producción.
- Precio Promedio: Refleja la valoración de la carne en diferentes etapas de procesamiento.
- Valor de la Producción: Representa el valor monetario total de la carne, con una variabilidad amplia.
- Peso Promedio: Relacionado con el tamaño promedio de los animales en diferentes etapas de procesamiento.
- Cabezas de Ganado Bovino: Métrica de cuantificación de la cantidad de ganado.
- Población y Población Varón 15-64 Años: Datos demográficos de la región, relevantes para la mano de obra o el consumo.

Estos datos ofrecen una visión completa de la producción de ganado bovino en el Estado de Hidalgo, abarcando desde la cantidad de carne producida hasta los precios y valores de la producción, así como métricas demográficas. La gran variabilidad en estos datos podría reflejar diferencias significativas en la producción entre diferentes regiones o periodos de tiempo, lo cual será crucial para el análisis posterior en este estudio de caso.
Para observar más detalladamente de forma visual podemos ocupar una visualización rápida de componentes como esta, para darnos una idea de cómo están aproximadamente distribuidos los datos:

Figura 4

Figura de visualización rápida de los datos



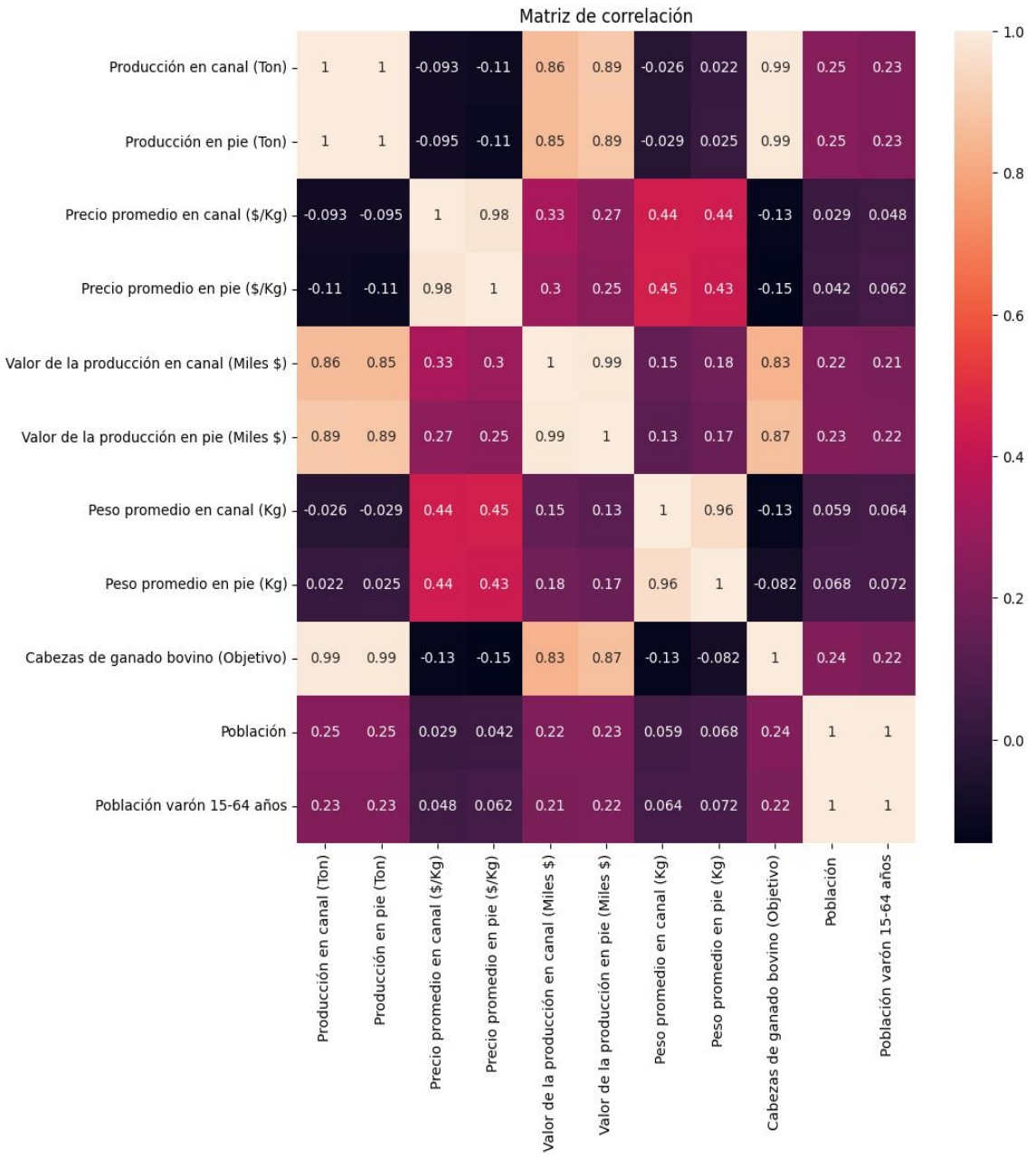
Nota: Esta figura simplemente sirve para generar una idea de cómo están distribuidos los datos de manera visual, como una forma inicial del EDA.

4.6 Análisis de Correlación

Para entender cómo las diferentes variables están relacionadas entre sí, y especialmente cómo se relacionan con la variable objetivo "Cabezas de ganado bovino (Objetivo)", se realiza un análisis correlacional.

Figura 5

Figura de la matriz de correlación entre las variables



Nota: la matriz ocupa un gráfico de calor que representa con los colores más claros una relación más fuerte entre las variables y más obscura para una relación menos fuerte o nula.

Los coeficientes de correlación en la matriz indican la fuerza y la dirección de la relación lineal entre dos variables. (Hair, 2010). A continuación, se describen los resultados del análisis correlacional:

Producción y Valor de la Producción

- Producción en Canal (Ton): Correlación = 0.9911. Fuerte correlación positiva, casi perfecta.
- Producción en Pie (Ton): Correlación = 0.9908. Fuerte correlación positiva, similar a la anterior.
- Valor de la Producción en Canal (Miles $): Correlación = 0.8305. Fuerte correlación positiva.
- Valor de la Producción en Pie (Miles $): Correlación = 0.8688. Fuerte correlación positiva, ligeramente mayor que la anterior.

Precio Promedio

- Precio Promedio en Canal ($/Kg): Correlación = -0.1288. Correlación negativa débil.
- Precio Promedio en Pie ($/Kg): Correlación = -0.1460. Correlación negativa un poco más fuerte.

Peso Promedio

- Peso Promedio en Canal (Kg): Correlación = -0.1293. Correlación negativa débil.
- Peso Promedio en Pie (Kg): Correlación = -0.0817. Correlación negativa aún más débil.

Demográficas

- Población: Correlación = 0.2377. Correlación positiva débil.
- Población Varón 15-64 Años: Correlación = 0.2166. Correlación positiva débil, similar a la población total.

Interpretación:

- Producción y Valor de la Producción: Estas variables muestran una fuerte correlación positiva con la cantidad de cabezas de ganado bovino, indicando que son buenos predictores de la variable objetivo.
- Precio Promedio y Peso Promedio: La correlación negativa débil sugiere que estas variables no son tan relevantes para predecir la cantidad de ganado bovino.

- Demográficas: La correlación positiva pero débil con la población sugiere una influencia menor en la variable objetivo, aunque se consideran para tener más datos debido a la limitada información disponible por la cantidad de municipios.

El objetivo es modelar o predecir la cantidad de cabezas de ganado bovino, las variables relacionadas con la producción y el valor de la producción serían las más relevantes. Las variables de precio, peso y población podrían ser menos significativas, pero aún pueden ofrecer insights adicionales en el análisis. Este análisis de correlación proporciona una base sólida para la selección de características y el modelado en las siguientes etapas del estudio de caso. (Hair, 2010).

Interpretación en el Contexto de los Datos

El análisis visual de los datos es esencial para comprender las tendencias, patrones y relaciones dentro del conjunto de datos. A continuación, se presentan tres tipos de gráficos que ofrecen diferentes perspectivas sobre los datos: gráficas de series de tiempo, gráficas de densidad y gráficas de correlación.

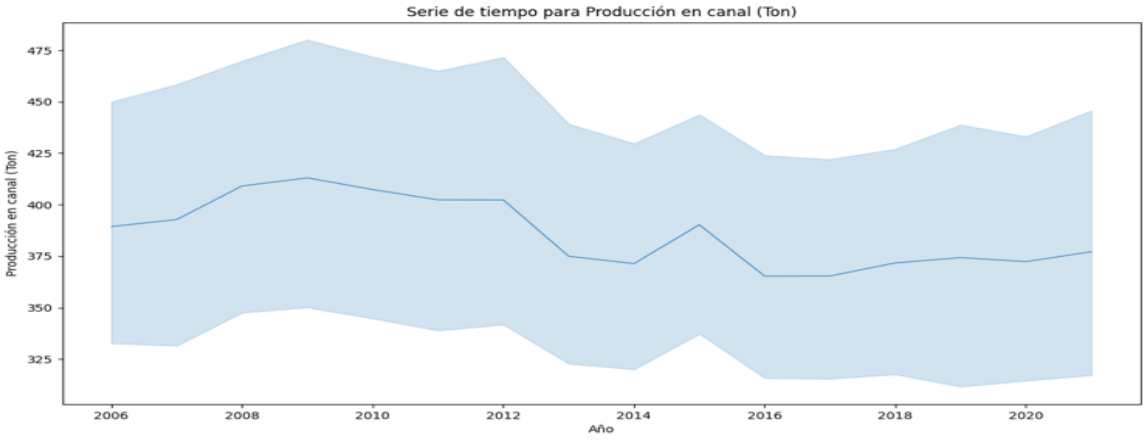
4.7 Gráficas de Series de Tiempo

Las gráficas de series de tiempo muestran cómo han cambiado las diferentes variables de producción a lo largo del tiempo. (Montgomery et al., 2015).

Estas gráficas ayudan a identificar tendencias, patrones estacionales o anomalías en los datos, como la fuerte correlación entre la "Producción en canal (Ton)" y la variable objetivo, o las fluctuaciones en los precios promedio.

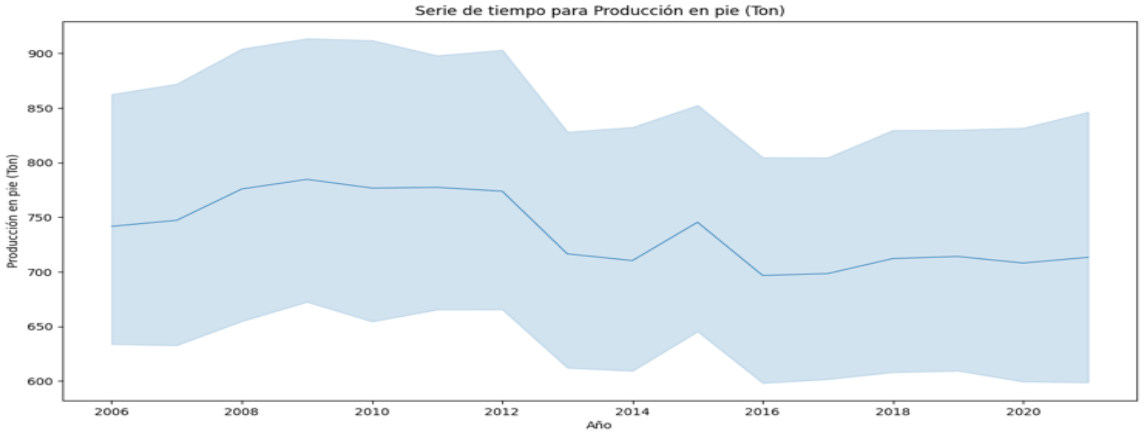
Gráfica 1

Series de tiempo para Producción en canal (Ton)



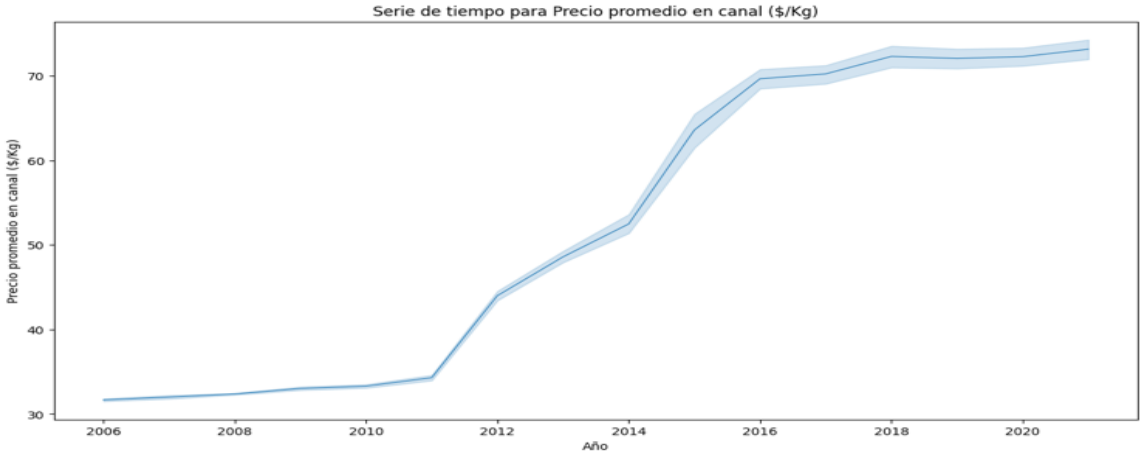
Gráfica 2

Series de tiempo para Producción en pie (Ton)



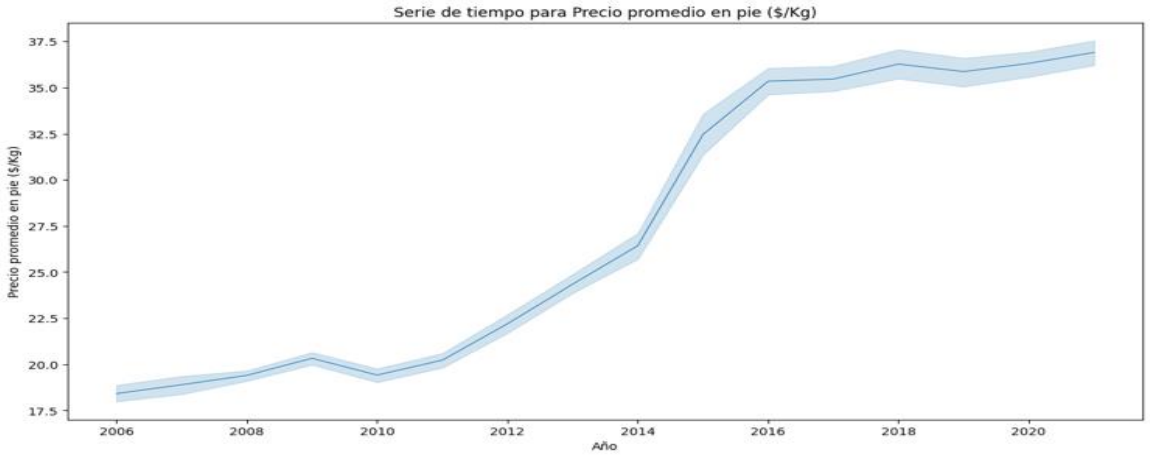
Gráfica 3

Series de tiempo para Precio promedio en canal ($/Kg)



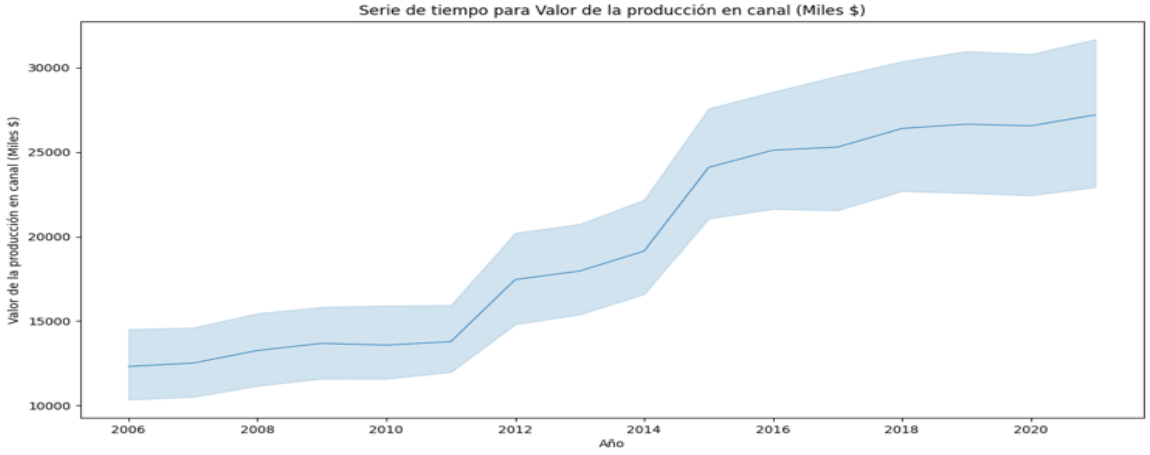
Gráfica 4

Series de tiempo para Precio promedio en pie ($/Kg)



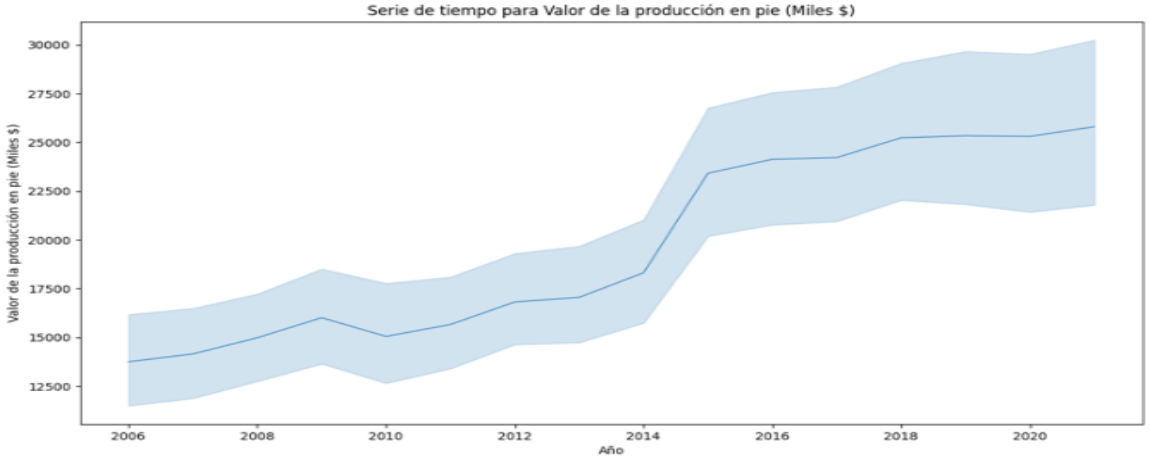
Gráfica 5

Series de tiempo para Valor de la producción en canal (Miles $)



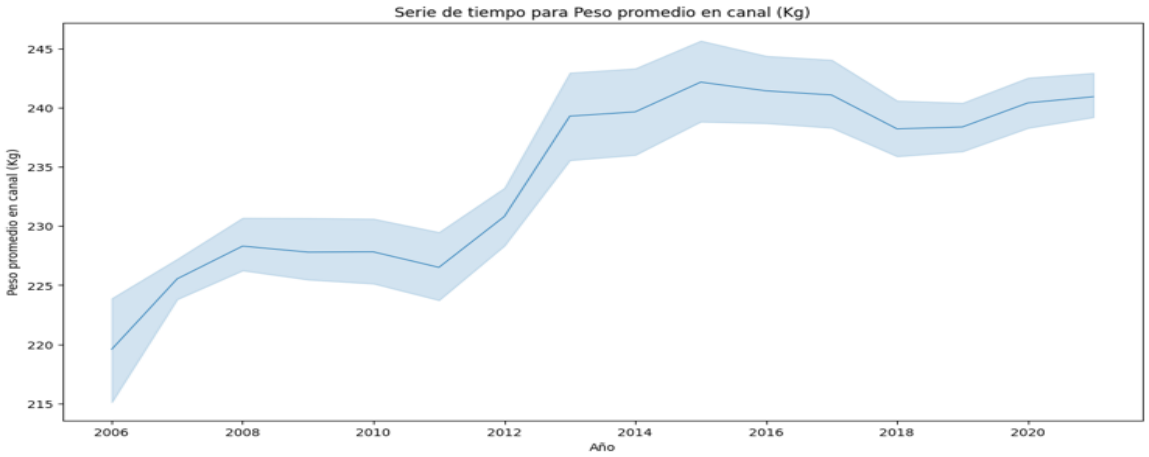
Gráfica 6

Series de tiempo para Valor de la producción en pie (Miles $)



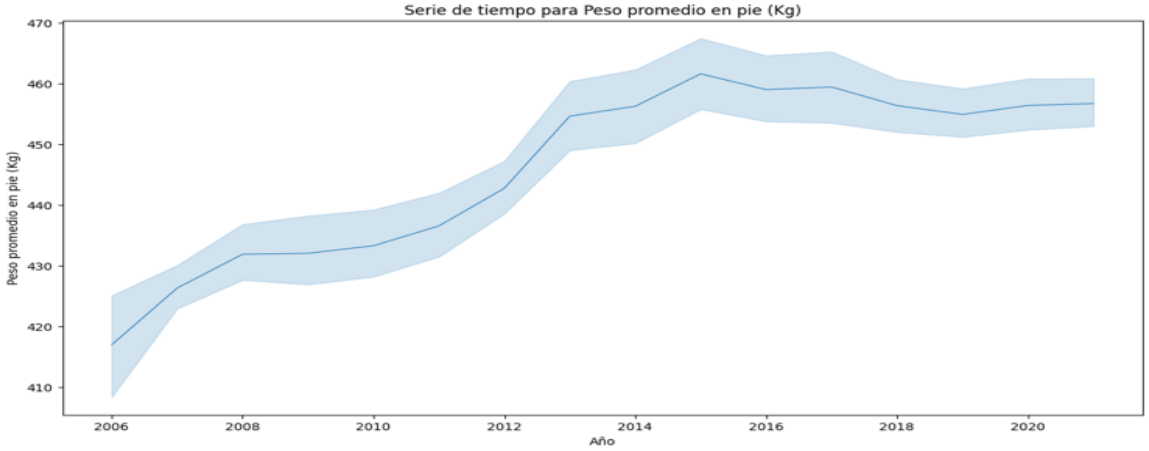
Gráfica 7

Series de tiempo para Peso promedio en canal (Kg)



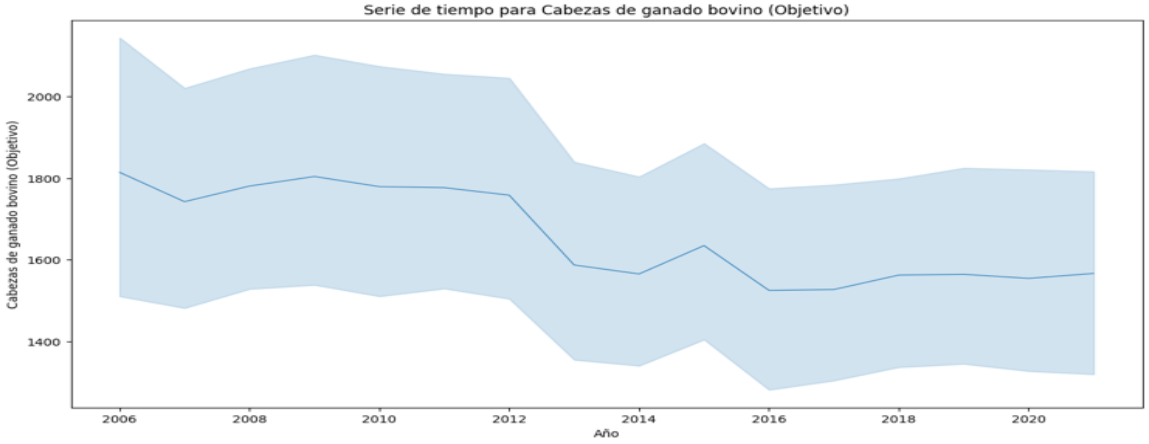
Gráfica 8

Series de tiempo para Peso promedio en pie (Kg)

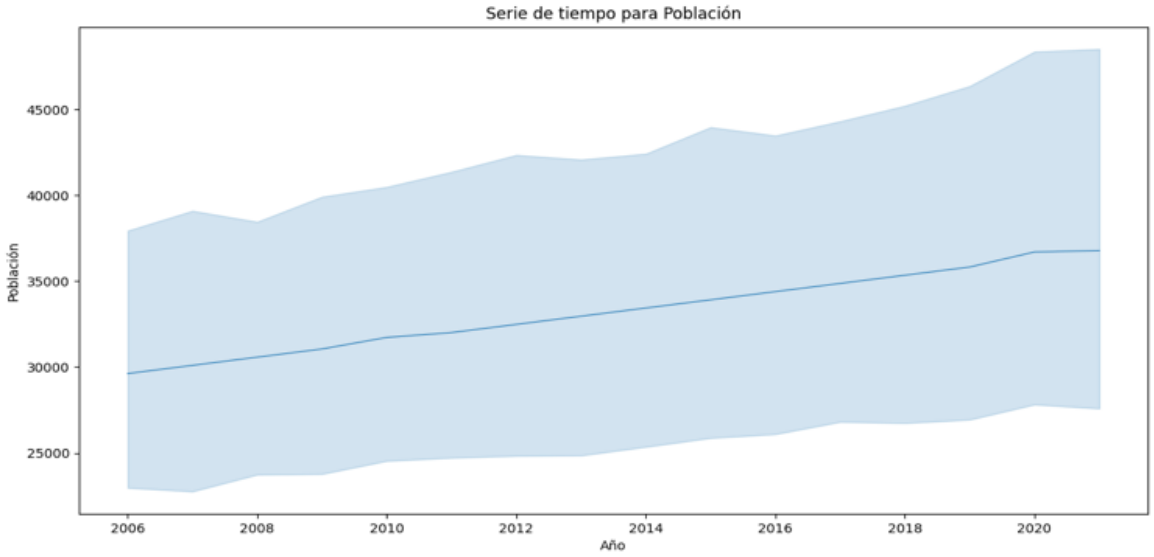


Gráfica 9

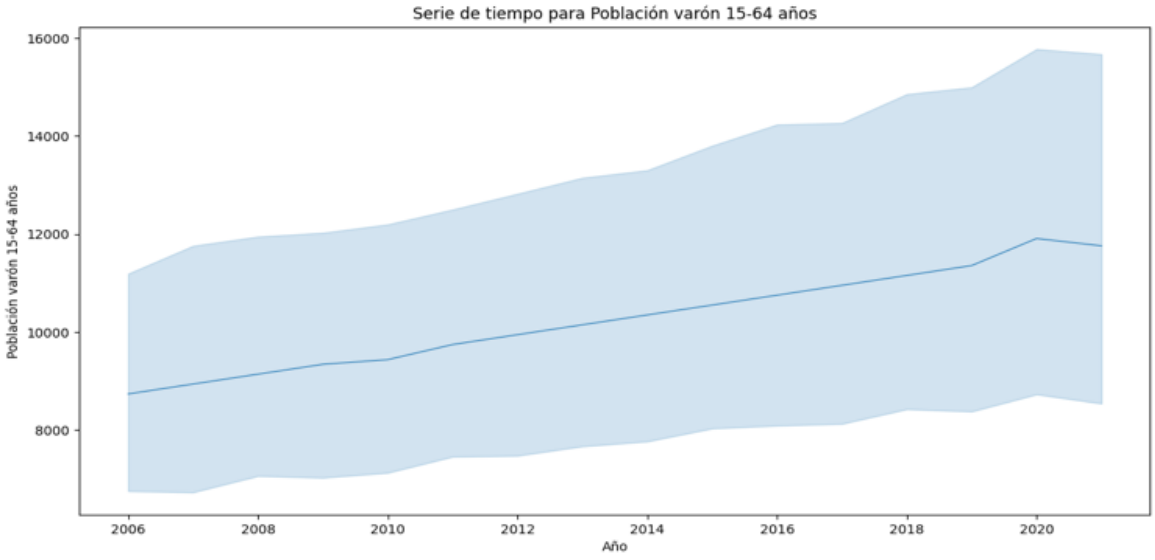
Series de tiempo para Cabezas de ganado bovino (Objetivo)



Gráfica 10
Series de tiempo para Población



Gráfica 11
Series de tiempo para Población varón 15-64 años



Estas gráficas de series de tiempo nos permiten observar cómo han evolucionado diferentes variables relacionadas con la producción y comercialización del ganado bovino a lo largo de los años. Cada línea en las gráficas representa la evolución de una variable específica en el tiempo, permitiéndonos identificar tendencias, fluctuaciones y posibles patrones que

podrían estar influenciados por factores externos como cambios en la economía, políticas agrícolas, condiciones climáticas, entre otros.

Por ejemplo, para este caso podemos ver que la serie de tiempo de:

- Cabezas de ganado bovino (Objetivo): Esta variable podría estar relacionada con las metas de producción o las proyecciones de crecimiento en la industria ganadera. Cambios en esta línea pueden reflejar estrategias de desarrollo del sector.

Donde la gráfica puede explicarse teniendo en cuenta estas consideraciones:

- Línea del Gráfico: La línea en sí representa la tendencia central de la variable a lo largo del tiempo. En el contexto de tus gráficas, esta línea indica cómo ha cambiado una variable específica (como la producción en toneladas, precio promedio por kilogramo, etc.) en cada año. Si la línea sube, baja o permanece relativamente plana, nos dice si la variable en cuestión ha aumentado, disminuido o se ha mantenido estable a lo largo del tiempo.

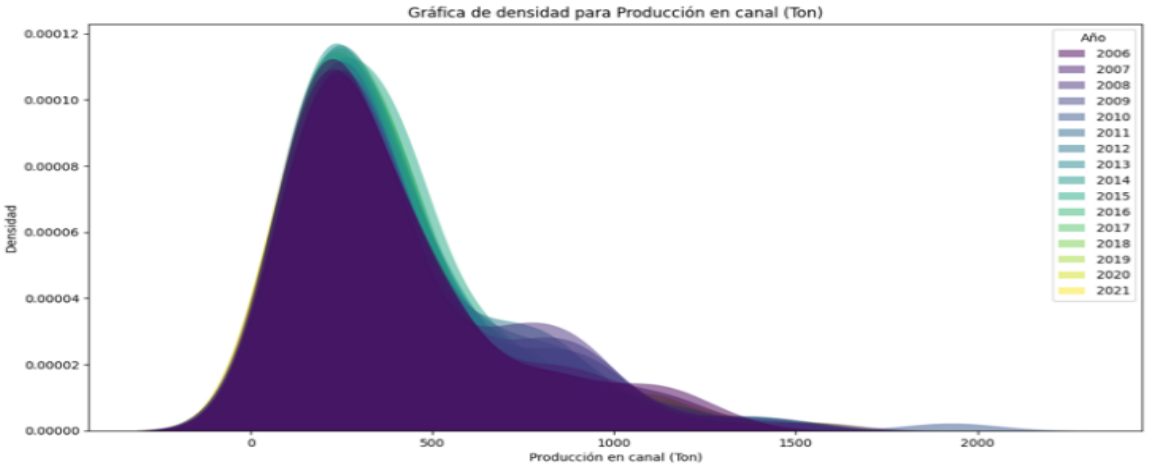
- Parte Sombreada (Rango o Intervalo de Confianza): La parte sombreada alrededor de la línea representa un intervalo de confianza o un rango de variabilidad de los datos. Este sombreado da una idea de la incertidumbre o la variabilidad en las estimaciones de la tendencia central. En términos más sencillos, muestra cuánto podrían variar los datos reales en torno a la línea de tendencia. Un área sombreada más amplia indica una mayor incertidumbre o variabilidad, mientras que un área más estrecha indica que los datos son más consistentes y hay menos variabilidad en torno a la tendencia central.

4.8 Gráficas de Densidad

Las gráficas de densidad proporcionan una manera visual de entender la distribución de cada variable en el conjunto de datos, revelando información sobre la forma, centralidad y dispersión de los datos.

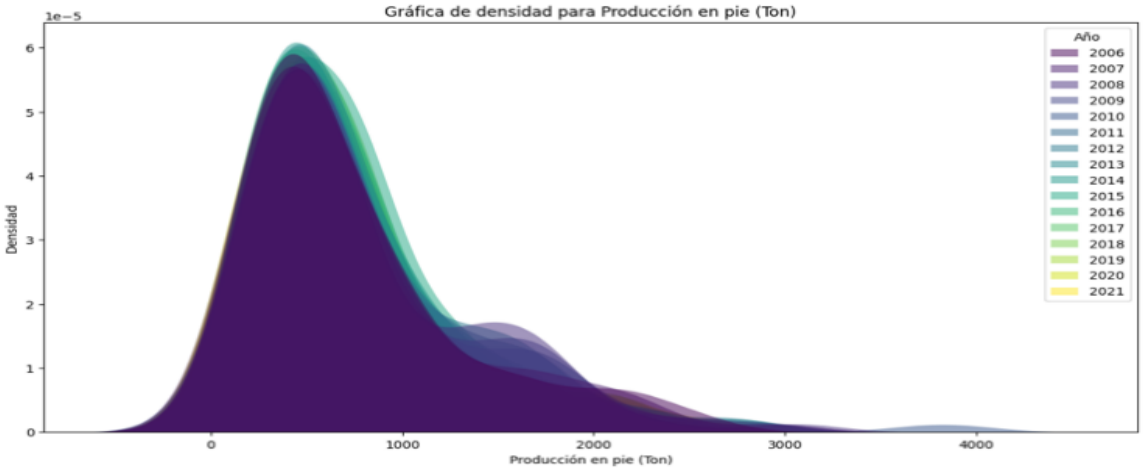
Gráfica 12

Gráfica de densidad para Producción en canal (Ton)



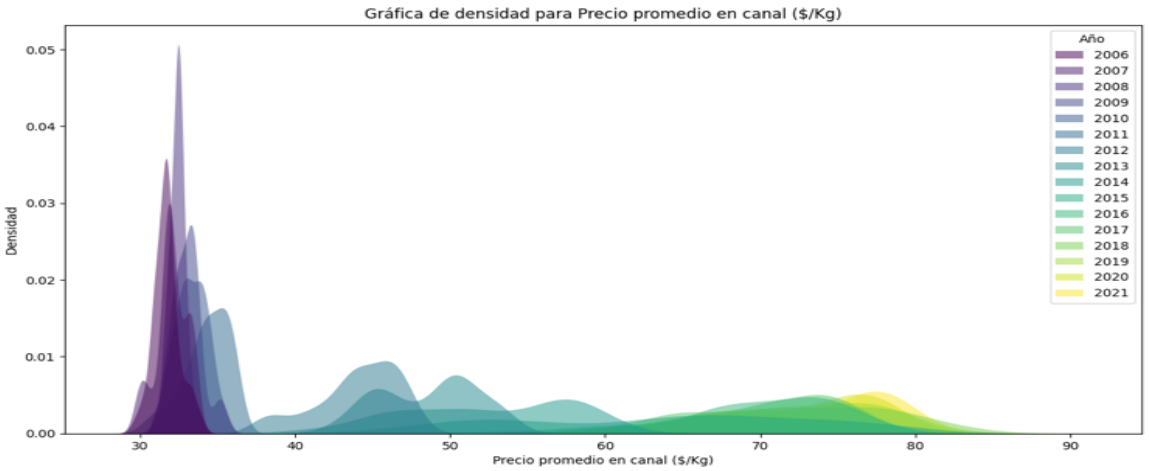
Gráfica 13

Gráfica de densidad para Producción en pie (Ton)



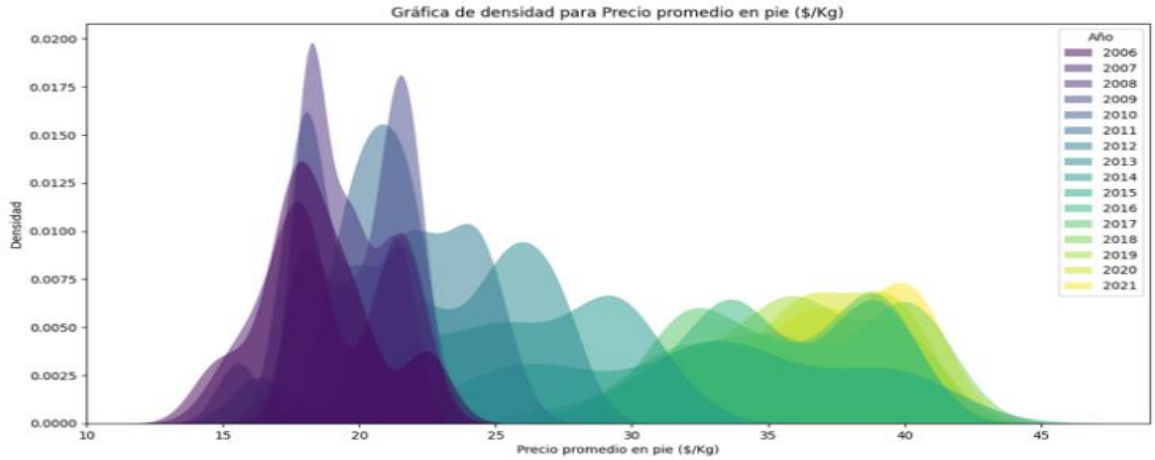
Gráfica 14

Gráfica de densidad para Precio promedio en canal ($/Kg)



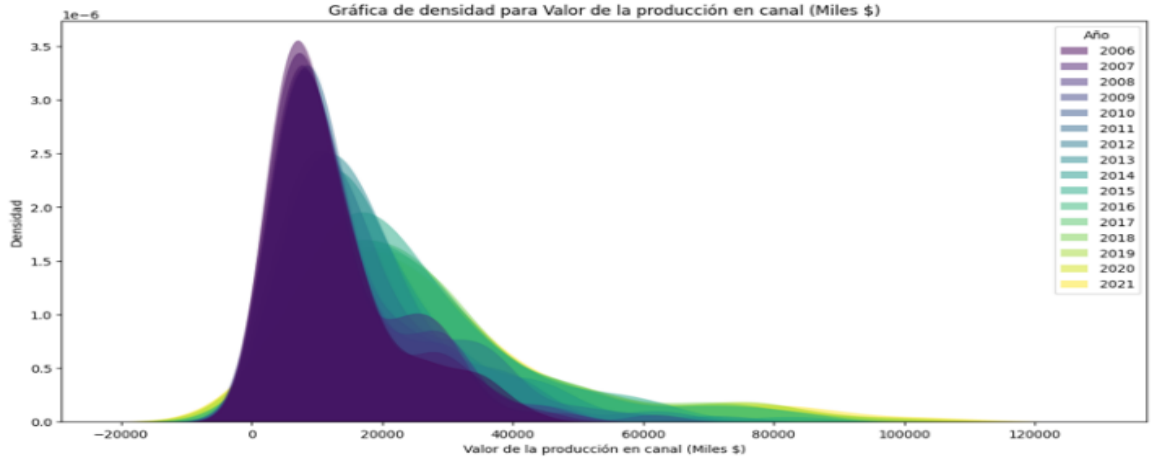
Gráfica 15

Gráfica de densidad para Precio promedio en pie ($/Kg)



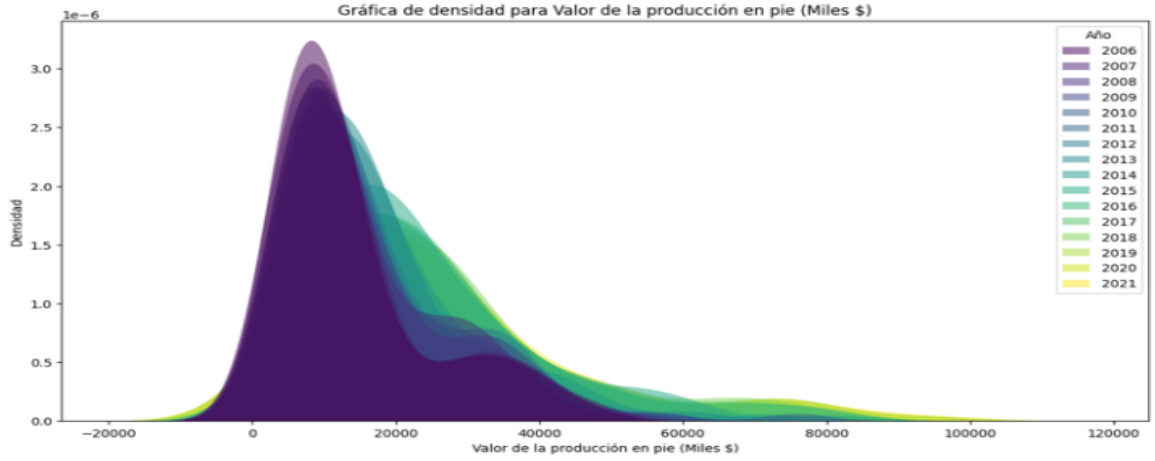
Gráfica 16

Gráfica de densidad para Valor de la producción en canal (Miles $)



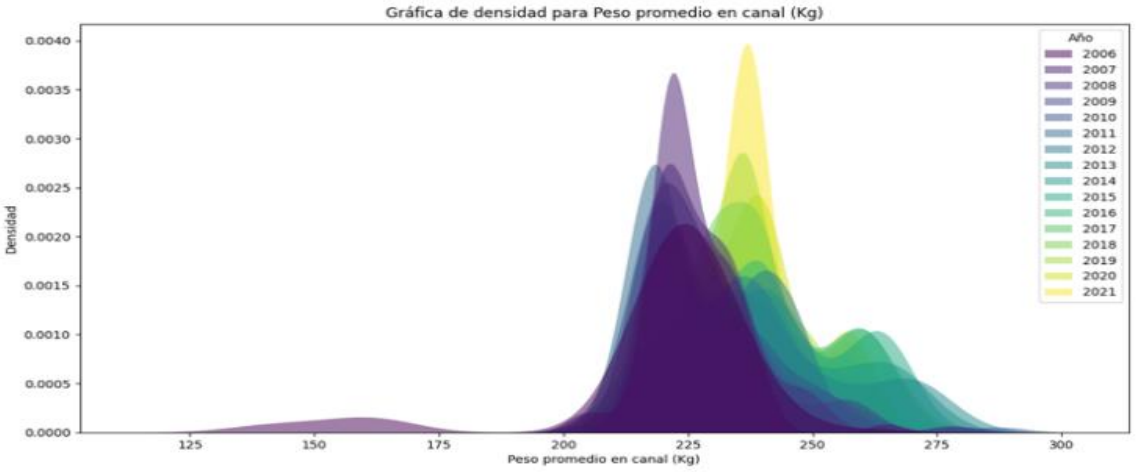
Gráfica 17

Gráfica de densidad para Valor de la producción en pie (Miles $)



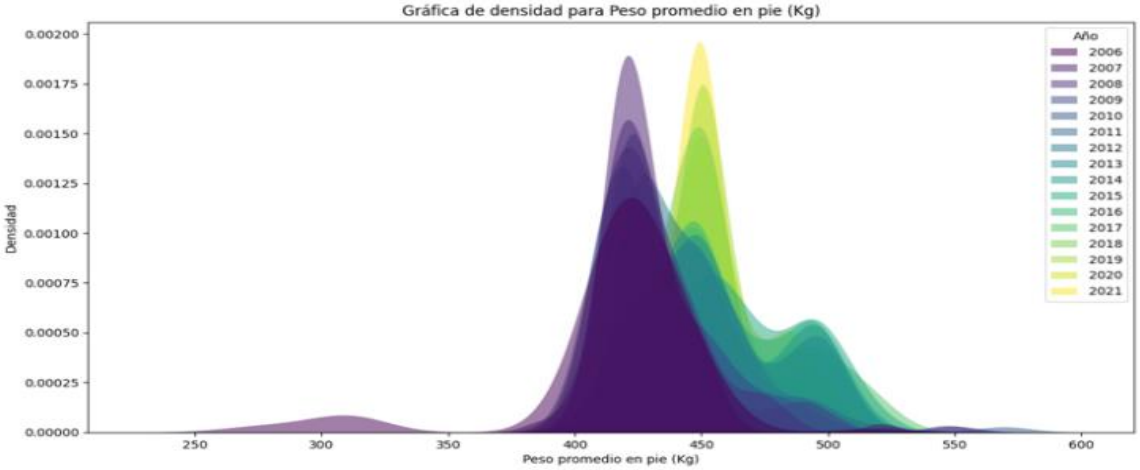
Gráfica 18

Gráfica de densidad para Peso promedio en canal (Kg)



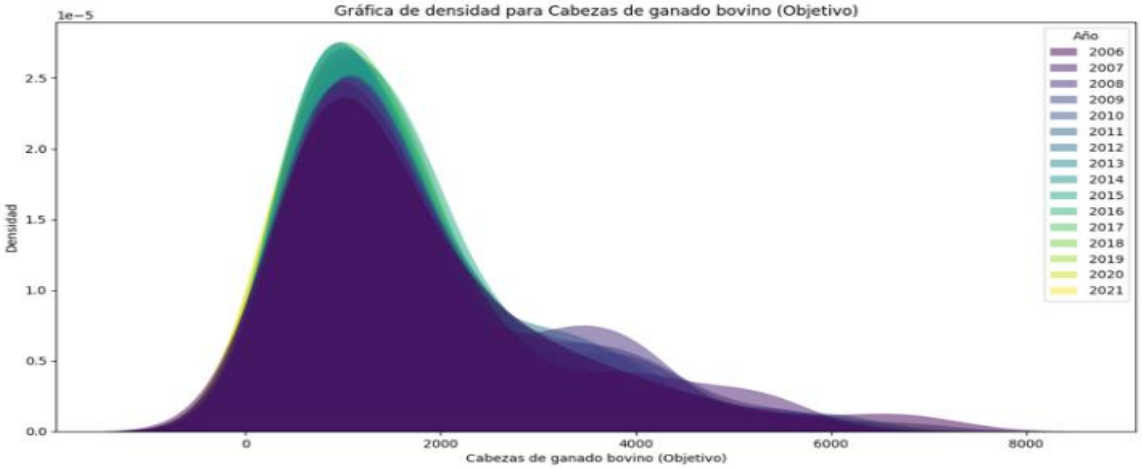
Gráfica 19

Gráfica de densidad para Peso promedio en pie (Kg)



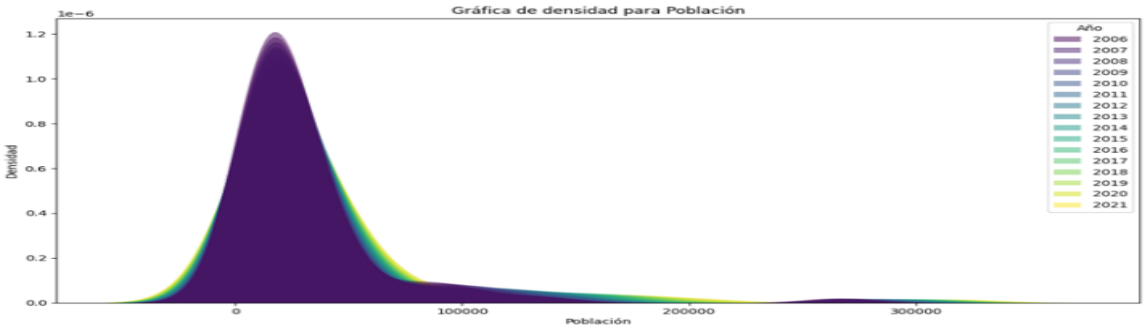
Gráfica 20

Gráfica de densidad para Cabezas de ganado bovino (Objetivo)



Gráfica 21

Gráfica de densidad para Población



Gráfica 22

Gráfica de densidad para Población varón 15-64 años

Estas gráficas ayudan a visualizar cómo los datos han cambiado a lo largo de los años y cómo fluyen, ofreciendo insights para su análisis.

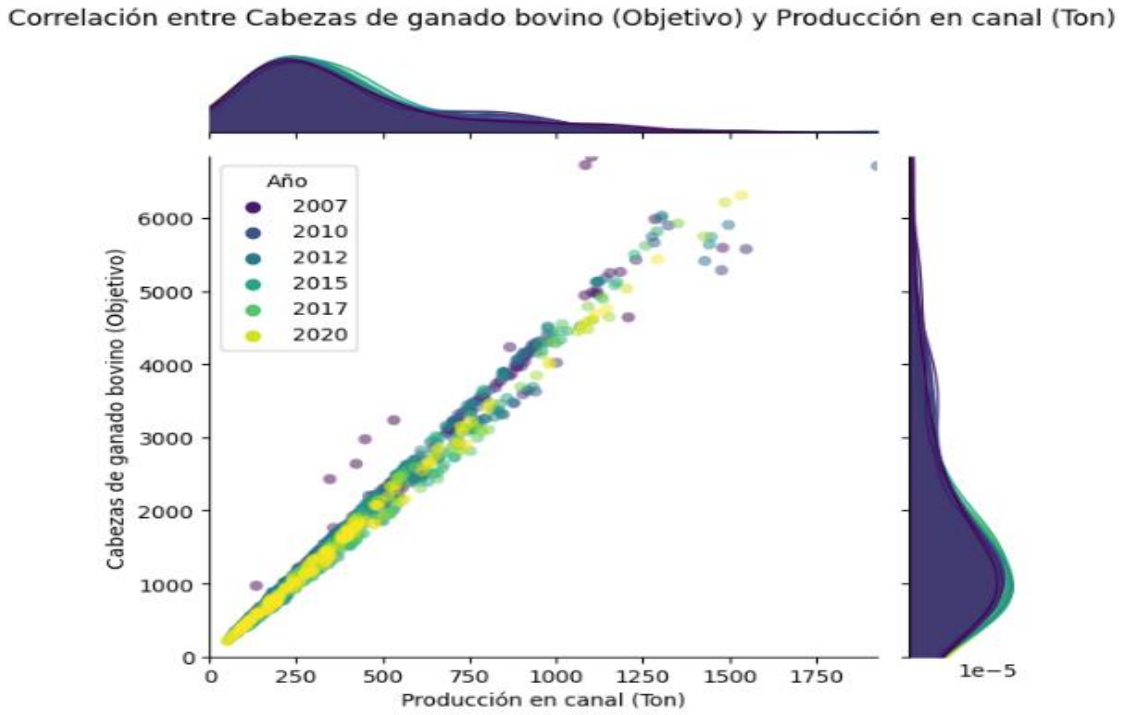
Las gráficas de densidad muestran la probabilidad relativa de que los valores de una variable caigan en diferentes rangos. La forma de la gráfica de densidad puede revelar si los datos siguen una distribución normal, si hay asimetría o sesgo en los datos, o si hay múltiples picos o modas en la distribución. (Wickham, 2016). El análisis de las gráficas de densidad puede ayudar a identificar patrones y características importantes de los datos, como la presencia de valores atípicos, la presencia de diferentes grupos o subpoblaciones, o la presencia de correlaciones entre variables. (Vanderplas, 2016).

4.9 Gráficas de Correlación

Las gráficas de correlación visualizan la relación entre cada variable de producción y la variable objetivo, así como sus distribuciones individuales.

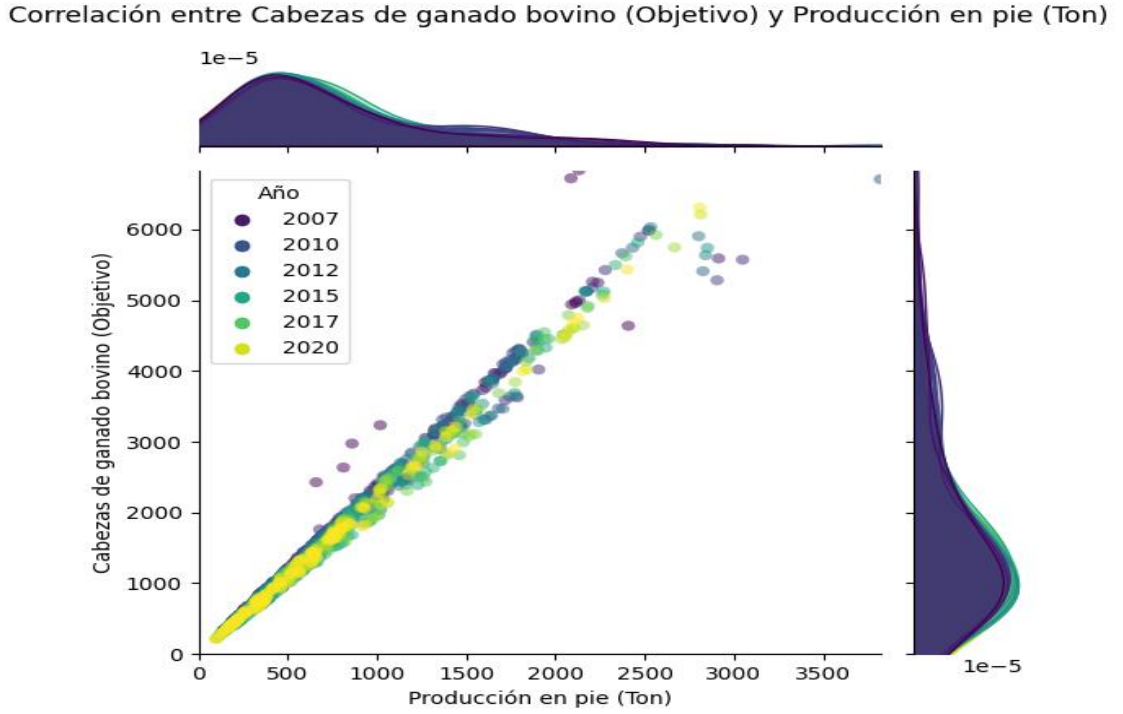
Gráfica 23

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Producción en canal (Ton)



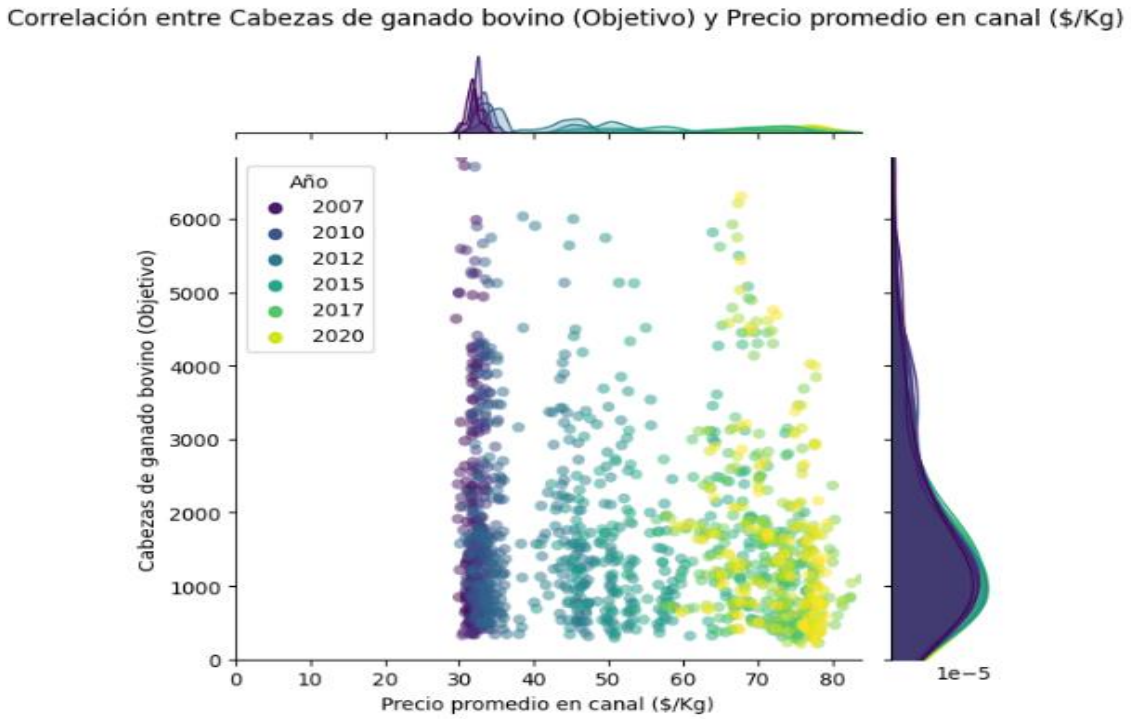
Gráfica 24

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Producción en pie (Ton)



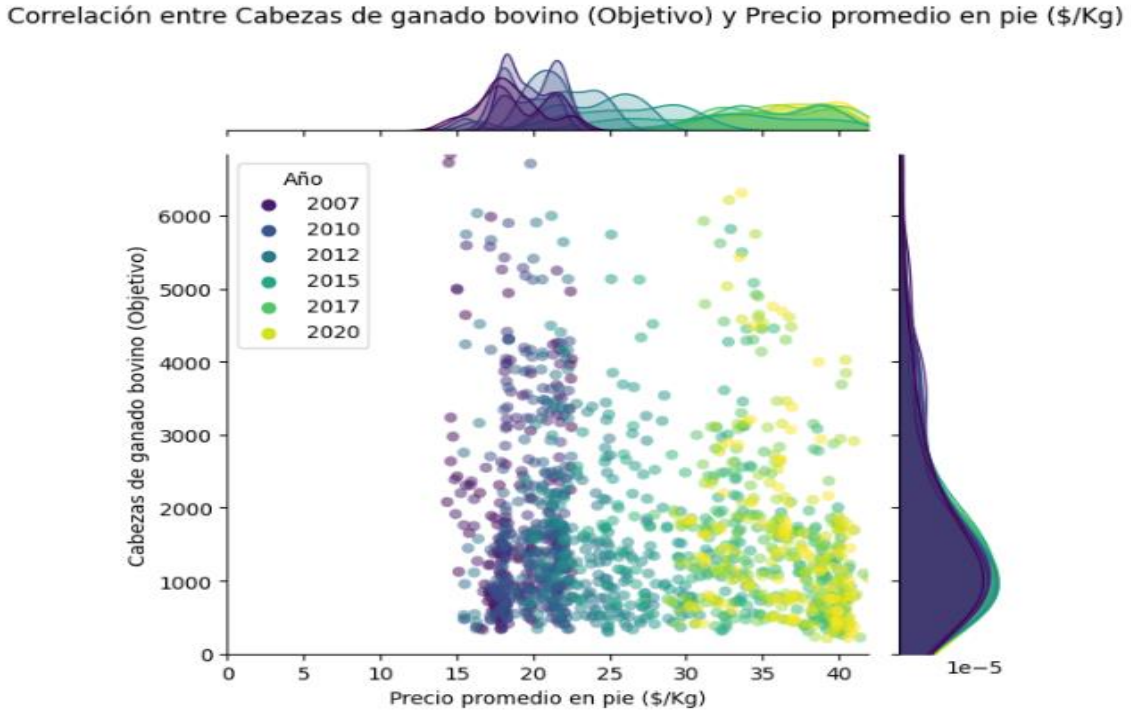
Gráfica 25

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Precio promedio en canal ($/Kg)



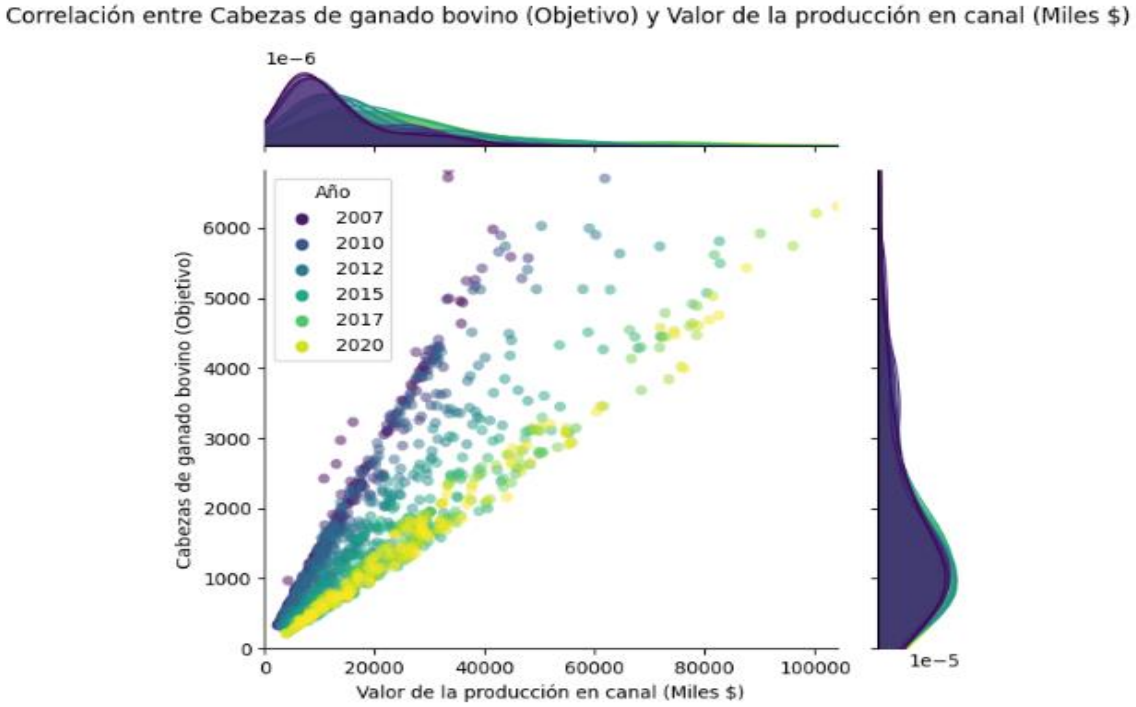
Gráfica 26

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Precio promedio en pie ($/Kg)



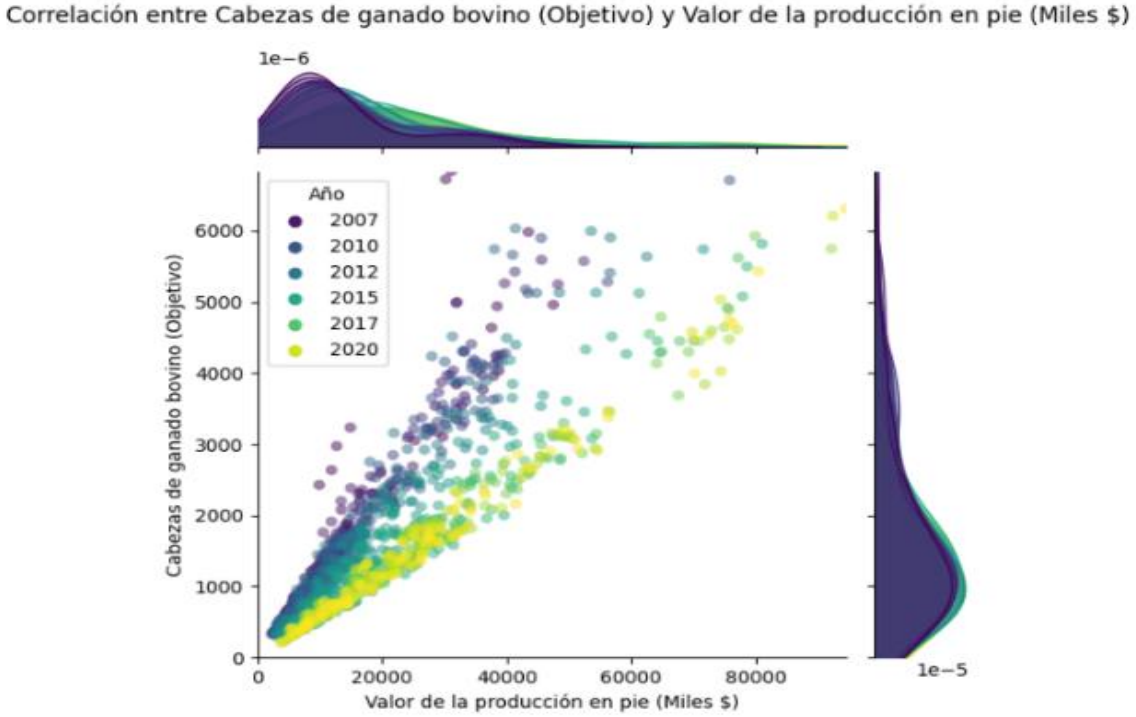
Gráfica 27

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Valor de la producción en canal (Miles $)



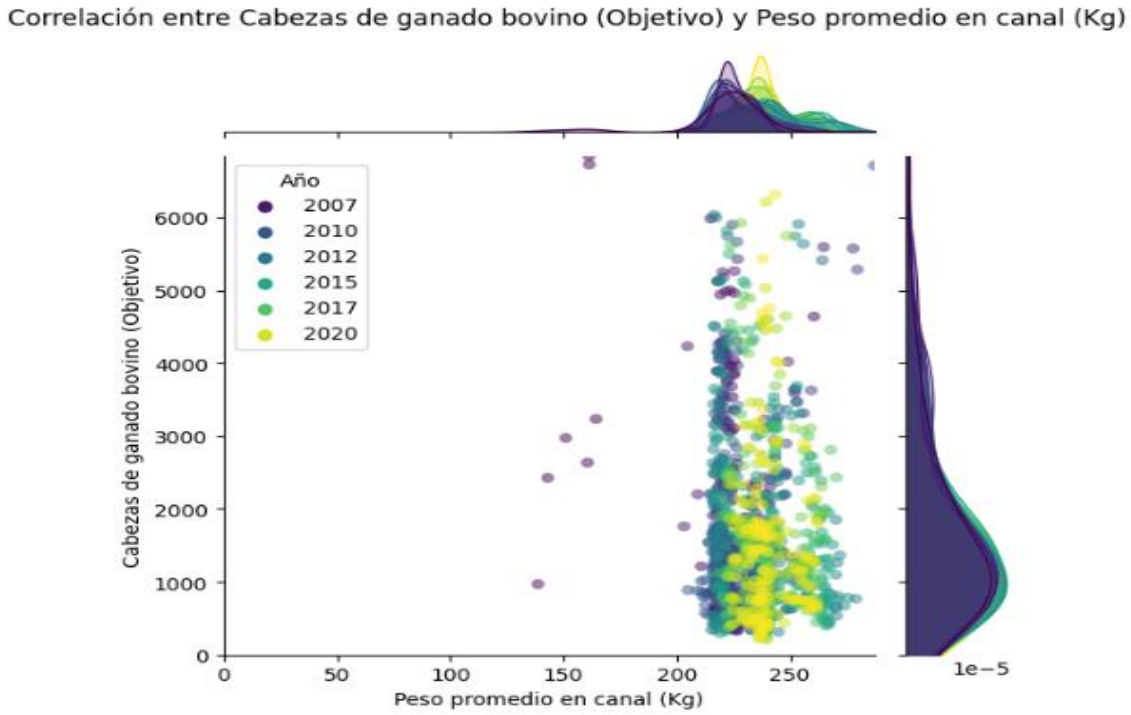
Gráfica 28

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Valor de la producción en pie (Miles $)



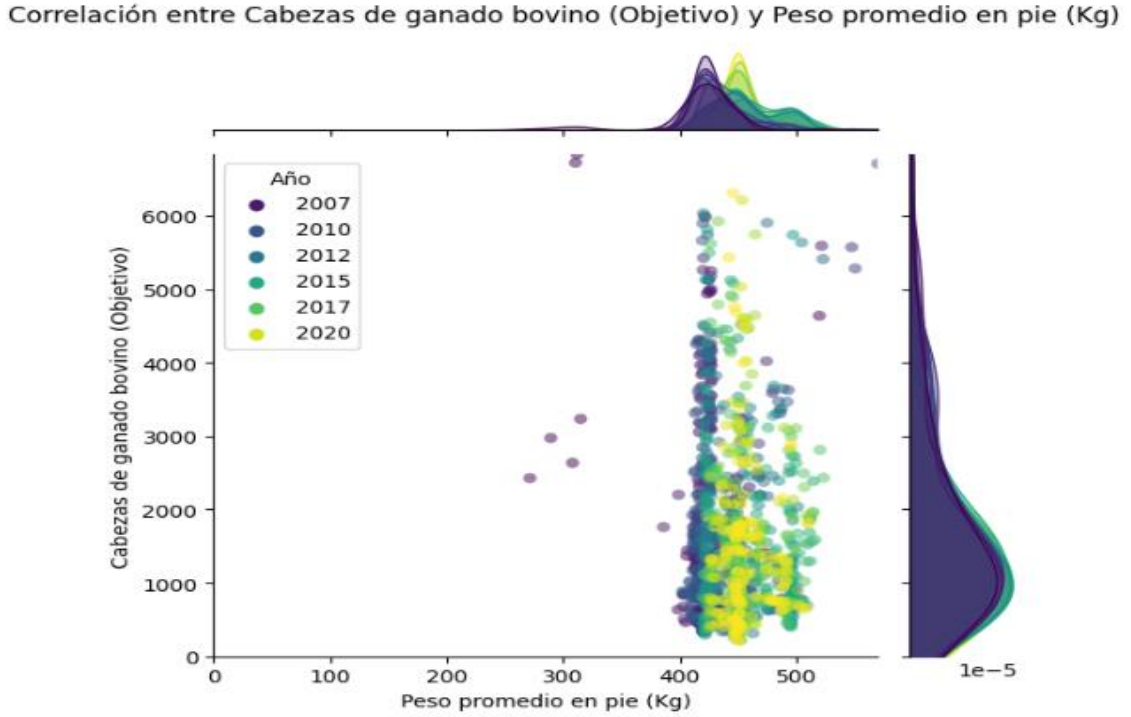
Gráfica 29

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Peso promedio en canal (Kg)



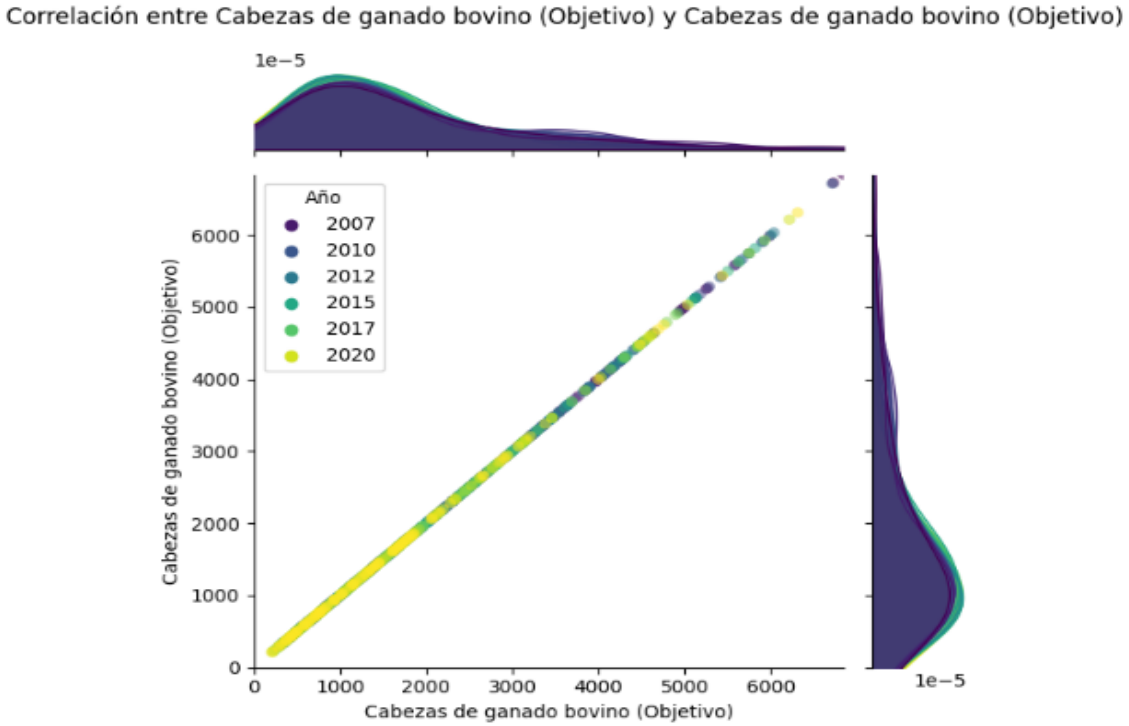
Gráfica 30

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Peso promedio en pie (Kg)



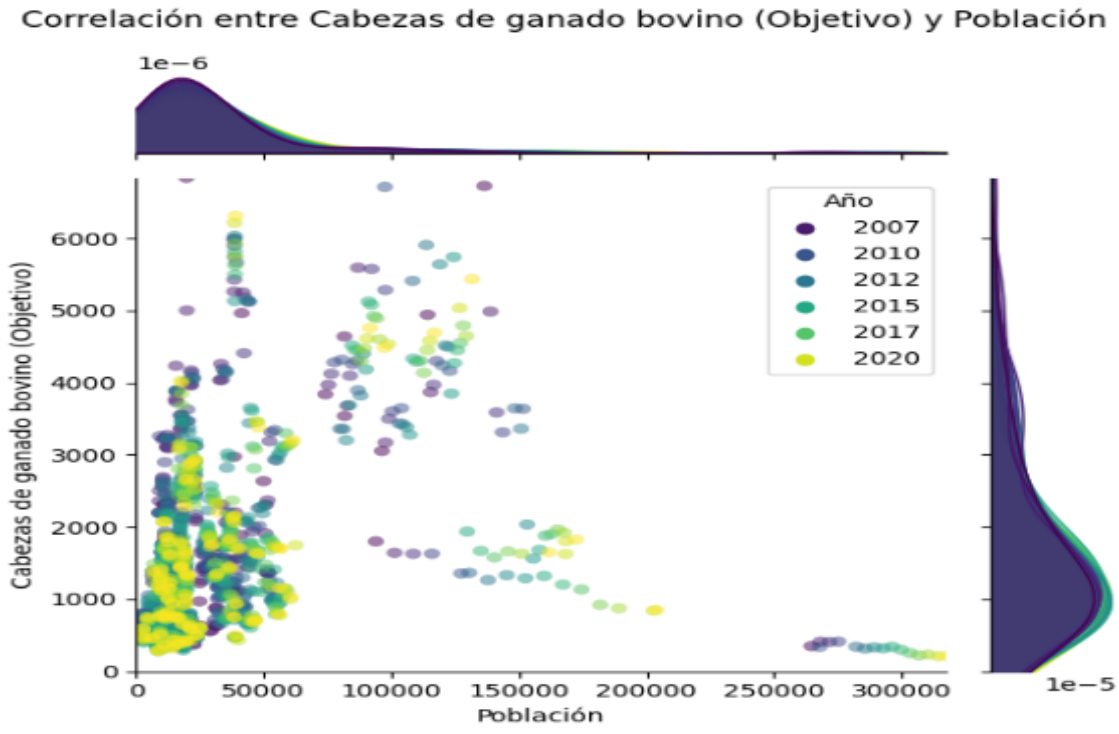
Gráfica 31

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Cabezas de ganado bovino (Objetivo)



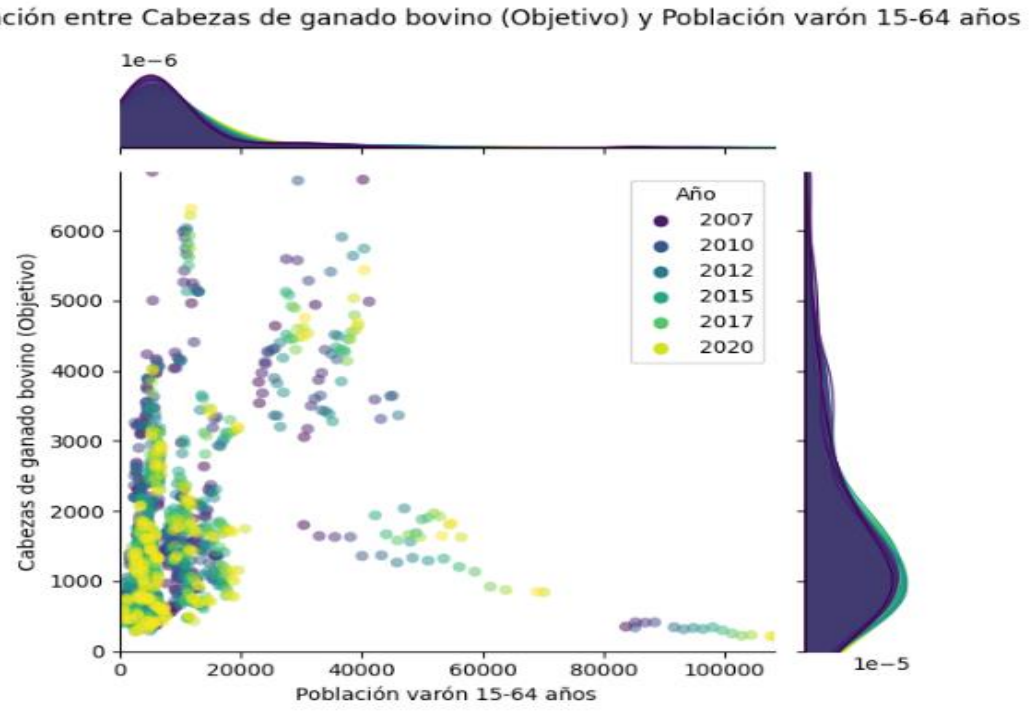
Gráfica 32

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Población



Gráfica 33

Gráfica de correlación entre Cabezas de ganado bovino (Objetivo) y Población varón 15-64 años



Estas gráficas proporcionan insights adicionales:

- Variables con Fuerte Correlación Positiva: Como la "Producción en canal (Ton)", mostrarán una clara tendencia ascendente.
- Variables con Correlación Negativa o Débil: Como los precios y pesos promedio, mostrarán puntos más dispersos sin una tendencia clara.
- Visualización de la Distribución: Además de la relación bivariada, estas gráficas también muestran la distribución de ambas variables en los márgenes.

La combinación de análisis estadístico y visualización gráfica permite una comprensión profunda y holística de los datos. Las gráficas de series de tiempo, densidad y correlación, cada una ofrece una perspectiva única y complementa el análisis correlacional previo, enriqueciendo la comprensión de las relaciones entre las variables en el conjunto de datos. (Wilke, 2019).

4.10 Prueba de Dickey Fuller

La prueba de Dickey-Fuller es esencial para el análisis de series temporales, ya que proporciona información sobre la estacionariedad de las series. La estacionariedad en el contexto de las series temporales se refiere a una propiedad donde las propiedades estadísticas de la serie, como la media y la varianza, permanecen constantes en el tiempo. En una serie estacionaria, no se observan tendencias o patrones a largo plazo que alteren estas propiedades estadísticas básicas. Esto es importante en el análisis de series temporales porque muchos modelos, como los modelos ARIMA, asumen que la serie es estacionaria. Si una serie no es estacionaria, puede requerir transformaciones, como diferenciación o logaritmos, para estabilizar su media y varianza antes de ser utilizada en modelos de series temporales. (Dickey Fuller, 1979).

Resultados de la Prueba:

Resultados de la prueba de Dickey-Fuller para Producción en canal (Ton):
Estadística ADF: -10.105051035527143
p-valor: 1.035203177862712e-17
Valores Críticos:
    1%: -3.435
    5%: -2.864
    10%: -2.568
Resultados de la prueba de Dickey-Fuller para Producción en pie (Ton):
Estadística ADF: -10.107719677317675
p-valor: 1.0194514807007446e-17
Valores Críticos:
    1%: -3.435
    5%: -2.864
    10%: -2.568
Resultados de la prueba de Dickey-Fuller para Precio promedio en canal ($/Kg):
Estadística ADF: -0.5047256837482652
p-valor: 0.8910910071967355

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Precio promedio en pie ($/Kg):

Estadística ADF: -0.5480285413915738

p-valor: 0.8822984420782585

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Valor de la producción en canal (Miles $):

Estadística ADF: -2.823715764405744

p-valor: 0.054965179385304944

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Valor de la producción en pie (Miles $):

Estadística ADF: -3.4020785890910865

p-valor: 0.010888216414034875

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Peso promedio en canal (Kg):

Estadística ADF: -2.553519635801789

p-valor: 0.10298928028693077

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Peso promedio en pie (Kg):

Estadística ADF: -2.4487691195938117

p-valor: 0.12844192393487158

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Cabezas de ganado bovino (Objetivo):

Estadística ADF: -8.605390514638188

p-valor: 6.694085778299656e-14

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Población:

Estadística ADF: -6.927470589073135

p-valor: 1.1058194975765377e-09

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Resultados de la prueba de Dickey-Fuller para Población varón 15-64 años:

Estadística ADF: -6.450894331602224

p-valor: 1.52448433159692e-08

Valores Críticos:

1%: -3.435

5%: -2.864

10%: -2.568

Los resultados incluyen:

- Estadística ADF: Un valor más negativo indica más evidencia de estacionariedad.
- p-valor: Un p-valor bajo (por ejemplo, < 0.05) indica que la serie es estacionaria.
- Valores Críticos: Si la estadística ADF es menor que estos valores, se rechaza la hipótesis nula, indicando estacionariedad.

Interpretación de los Resultados:

Basándonos en los resultados, podemos categorizar las variables en tres grupos:

- Series Estacionarias: Incluyen "Producción en canal (Ton)", "Producción en pie (Ton)", "Población", "Cabezas de ganado bovino (Objetivo)", "Población varón 15-64 años". Estas series tienen p-valores muy bajos y estadísticas ADF muy negativas, lo que significa que hay fuertes evidencias de estacionariedad.

- Series No Estacionarias: Comprenden "Precio promedio en canal ($/Kg)", "Precio promedio en pie ($/Kg)". Estas series tienen p-valores altos, lo que significa que no hay evidencia suficiente para rechazar la hipótesis nula, y las series pueden no ser estacionarias.

- Series con Resultados Intermedios: Incluyen "Valor de la producción en canal (Miles $)", "Valor de la producción en pie (Miles $)", "Peso promedio en canal (Kg)", "Peso promedio en pie (Kg)". Estos tienen p-valores intermedios y deben evaluarse en función de los valores críticos y el nivel de significancia elegido.

4.11 Gráficas de Retraso

Las gráficas de retraso son una herramienta esencial en el análisis de series temporales, ofreciendo una representación visual de la correlación temporal en una serie. Permiten identificar la estructura temporal y revelan cómo los valores de la serie en diferentes puntos en el tiempo están relacionados entre sí. (Brockwell & Davis, 2013).

Componentes:

- Valor en *t*: Representa el valor de la serie en un punto en el tiempo.
- Valor en *t*+1: Representa el valor de la serie en el siguiente punto en el tiempo.

Interpretación:

Las gráficas de retraso pueden revelar diferentes patrones:

- Correlación Positiva: Agrupación de puntos a lo largo de una línea diagonal ascendente, indicando autocorrelación positiva y una posible tendencia o patrón regular.
- Correlación Negativa: Agrupación de puntos a lo largo de una línea diagonal descendente, reflejando una correlación negativa entre los valores en tiempos consecutivos.
- Sin Correlación: Dispersión de puntos sin un patrón claro, sugiriendo que la serie puede ser más cercana a un proceso aleatorio.

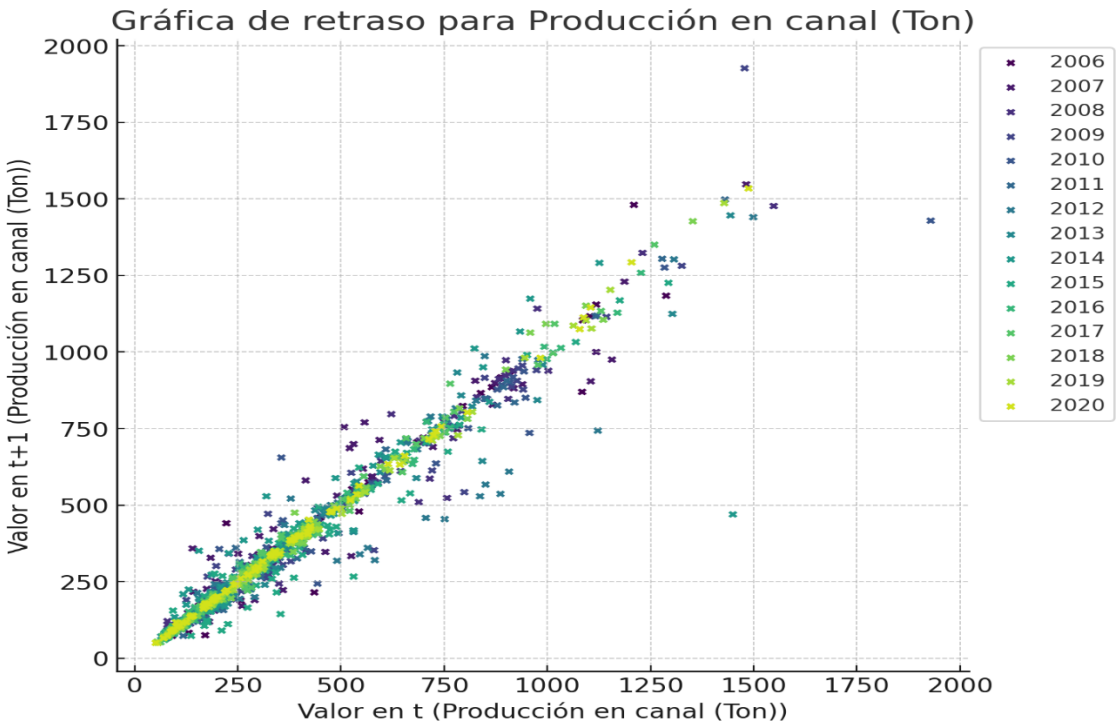
Aplicación:

Las gráficas de retraso son útiles para:

- Identificar la Autocorrelación: Detectar la estructura temporal en los datos, como tendencias o patrones estacionales.
- Seleccionar Modelos: Influenciar la selección de modelos de series temporales, como los modelos ARIMA.
- Diagnosticar Problemas: Ayudar a identificar problemas como la no estacionariedad.

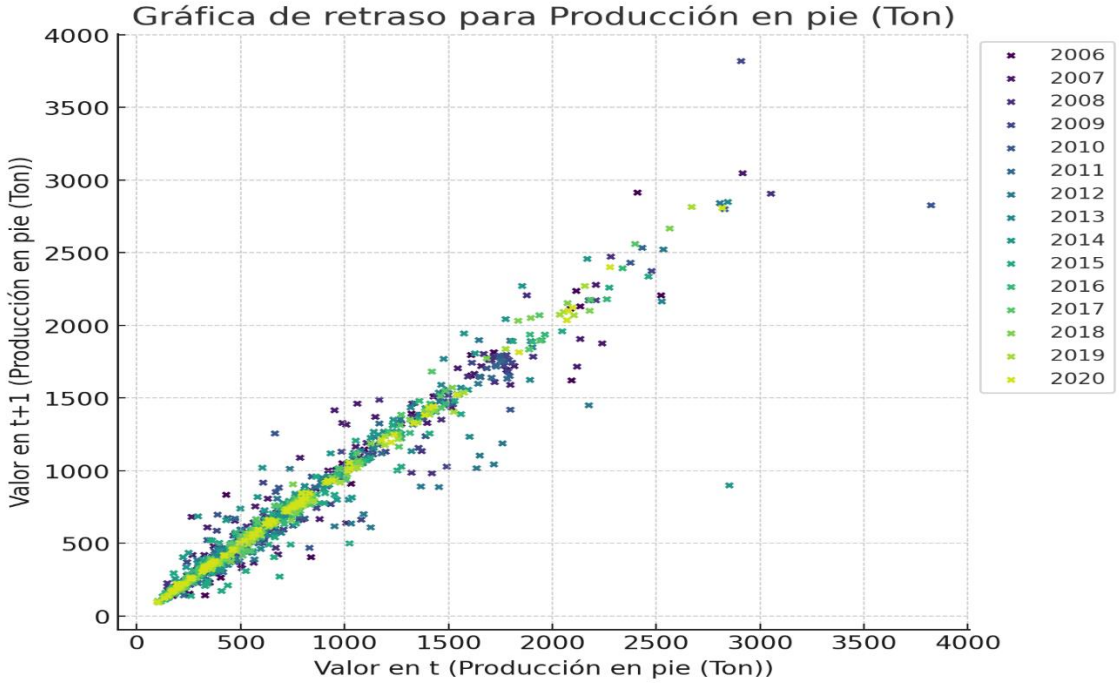
Gráfica 34

Gráfica de retraso para Producción en canal (Ton)



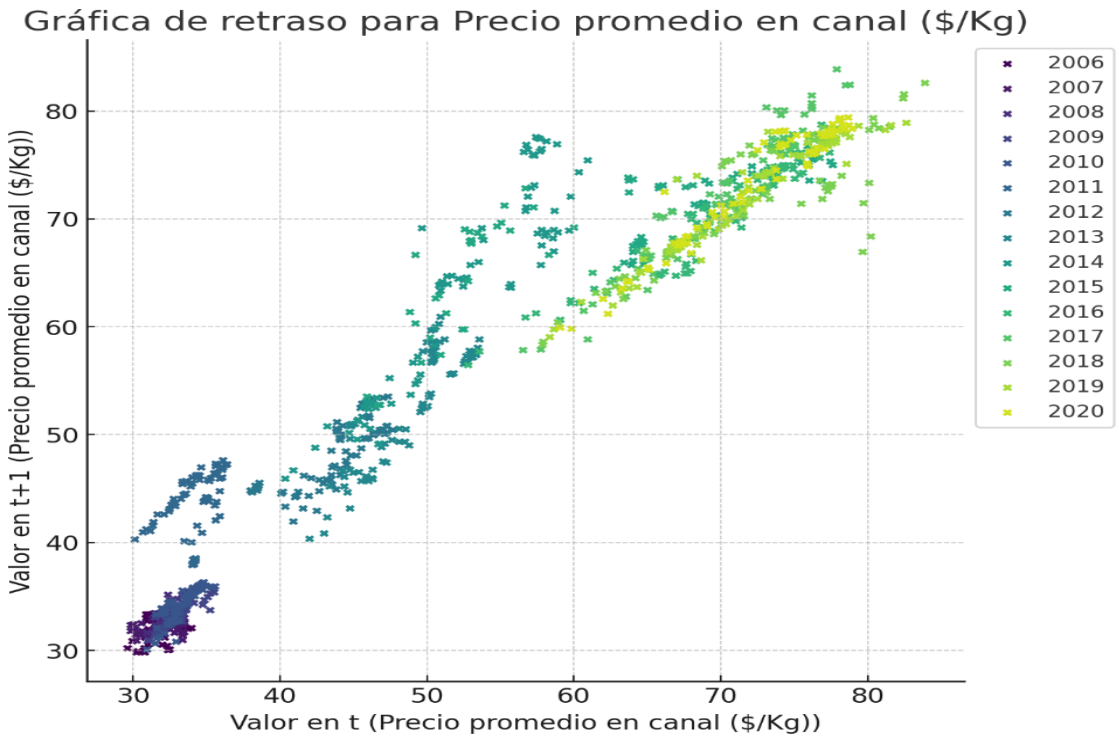
Gráfica 35

Gráfica de retraso para Producción en pie (Ton)



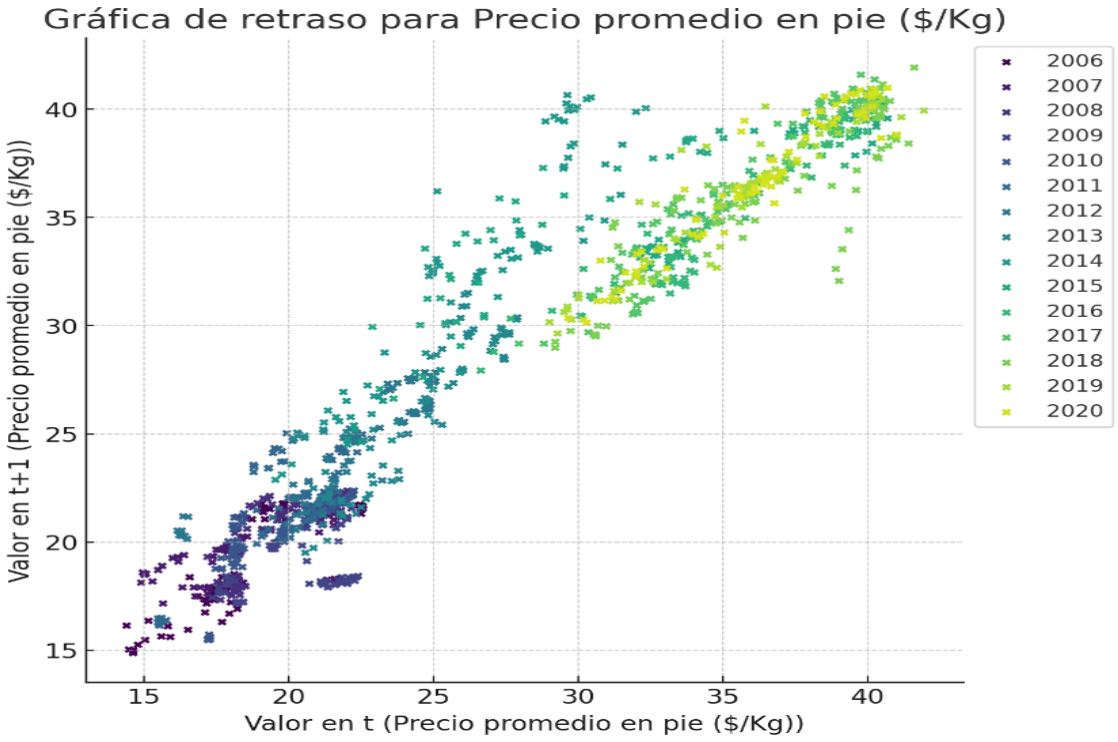
Gráfica 36

Gráfica de retraso para Precio promedio en canal ($/Kg)



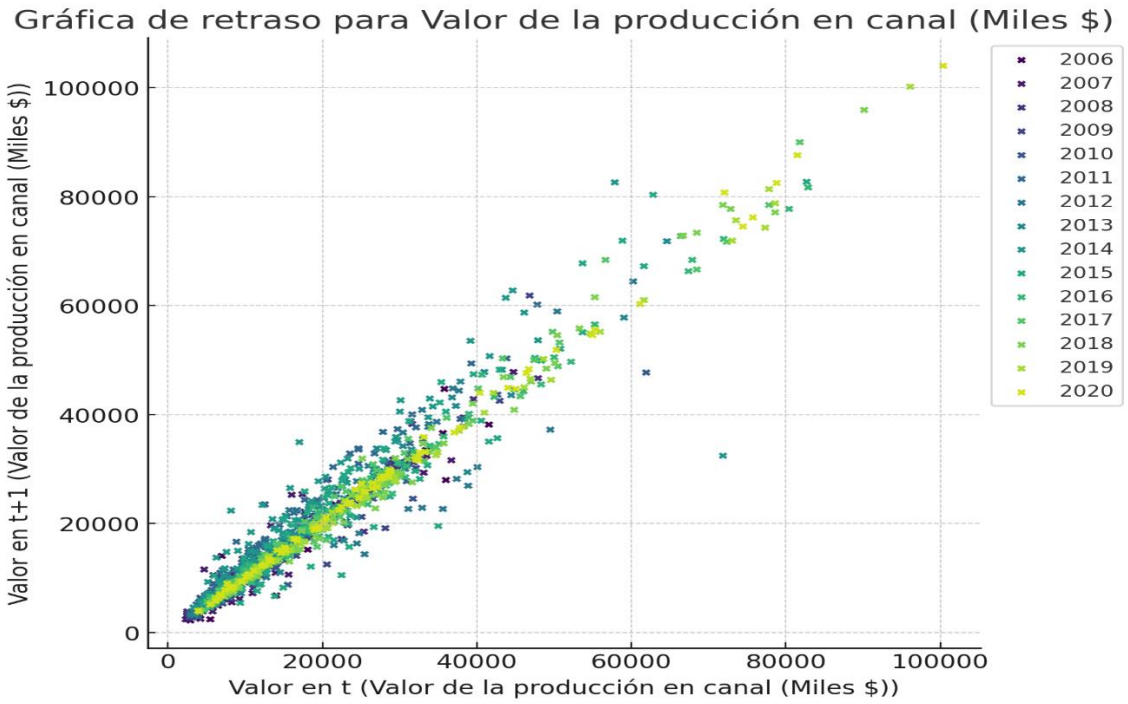
Gráfica 37

Gráfica de retraso para Precio promedio en pie ($/Kg)



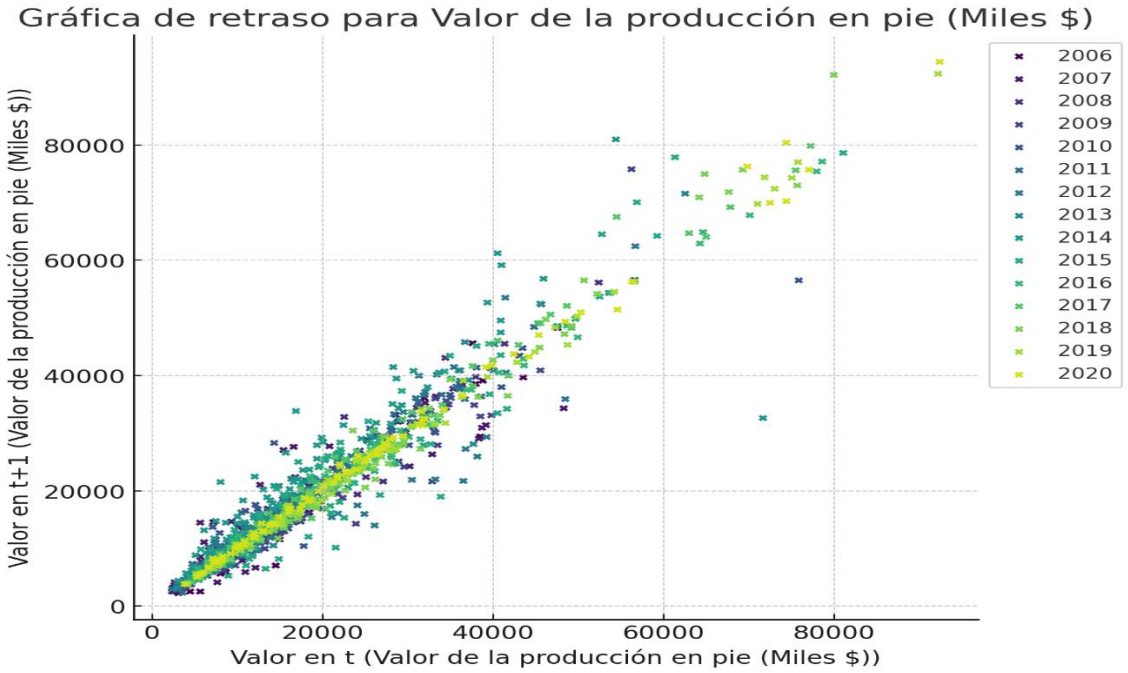
Gráfica 38

Gráfica de retraso para Valor de la producción en canal (Miles $)



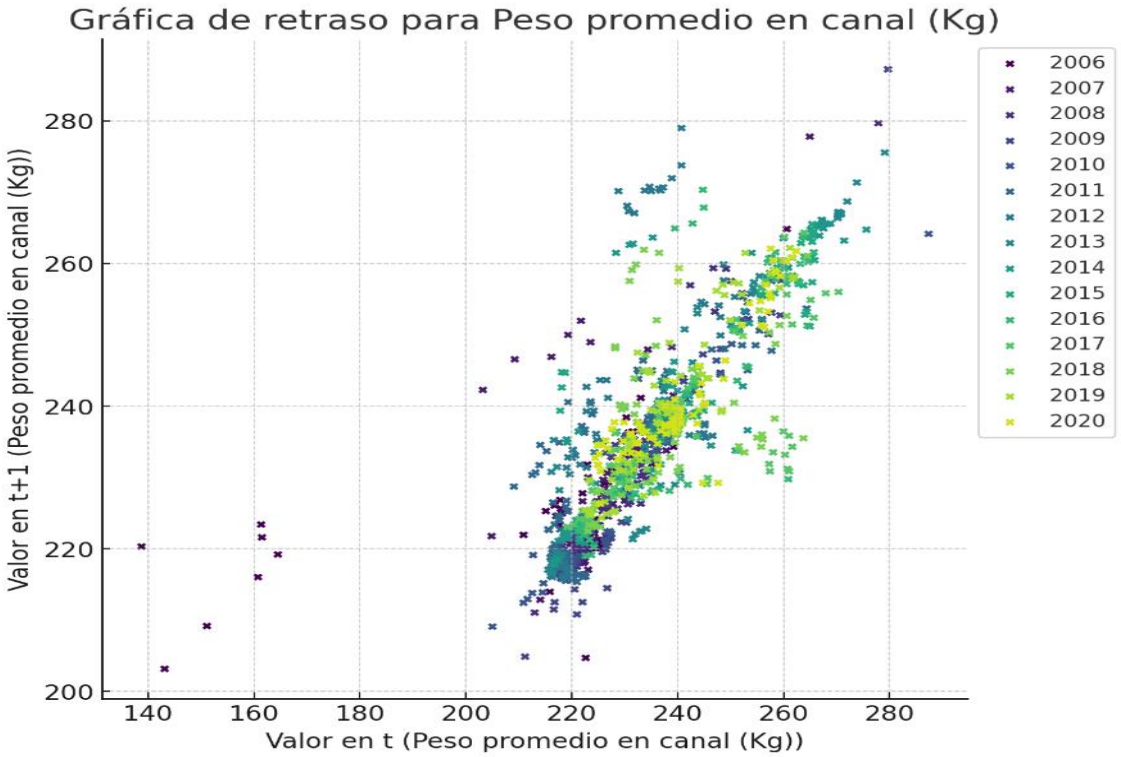
Gráfica 39

Gráfica de retraso para Valor de la producción en pie (Miles $)



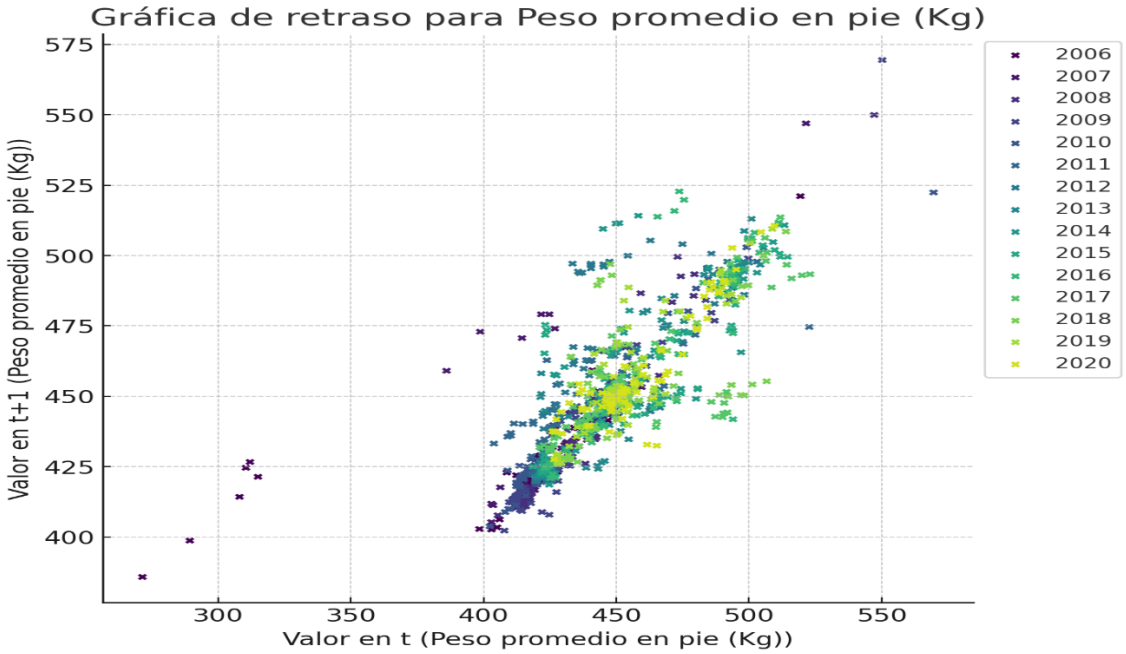
Gráfica 40

Gráfica de retraso para Peso promedio en canal (Kg)



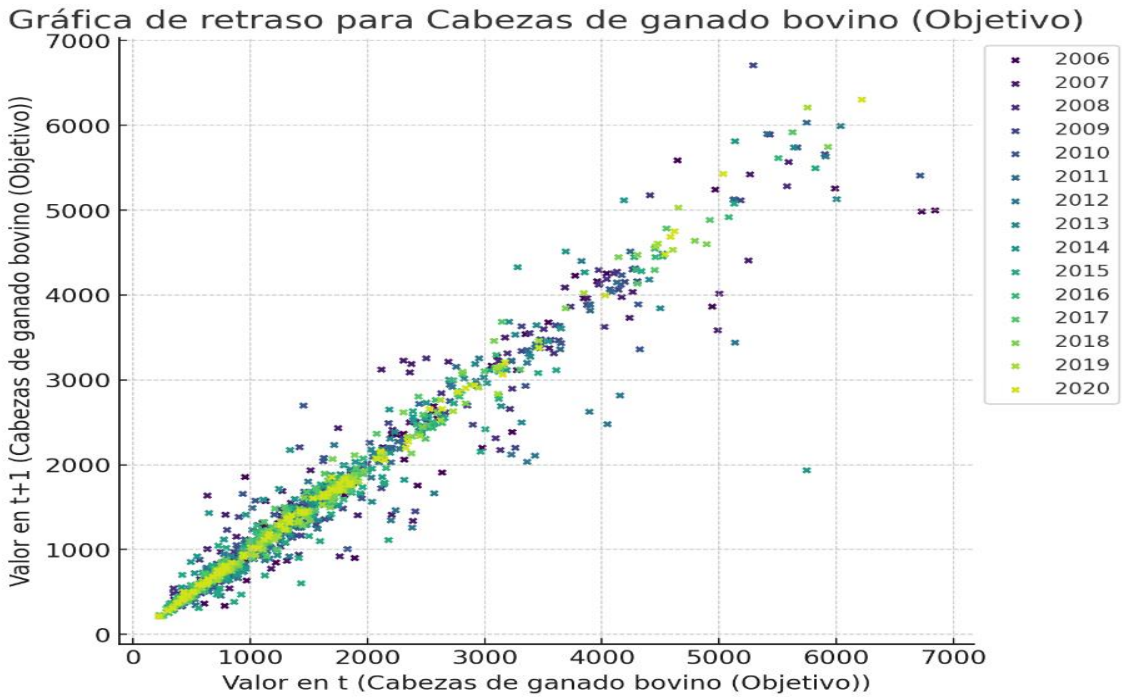
Gráfica 41

Gráfica de retraso para Peso promedio en pie (Kg)



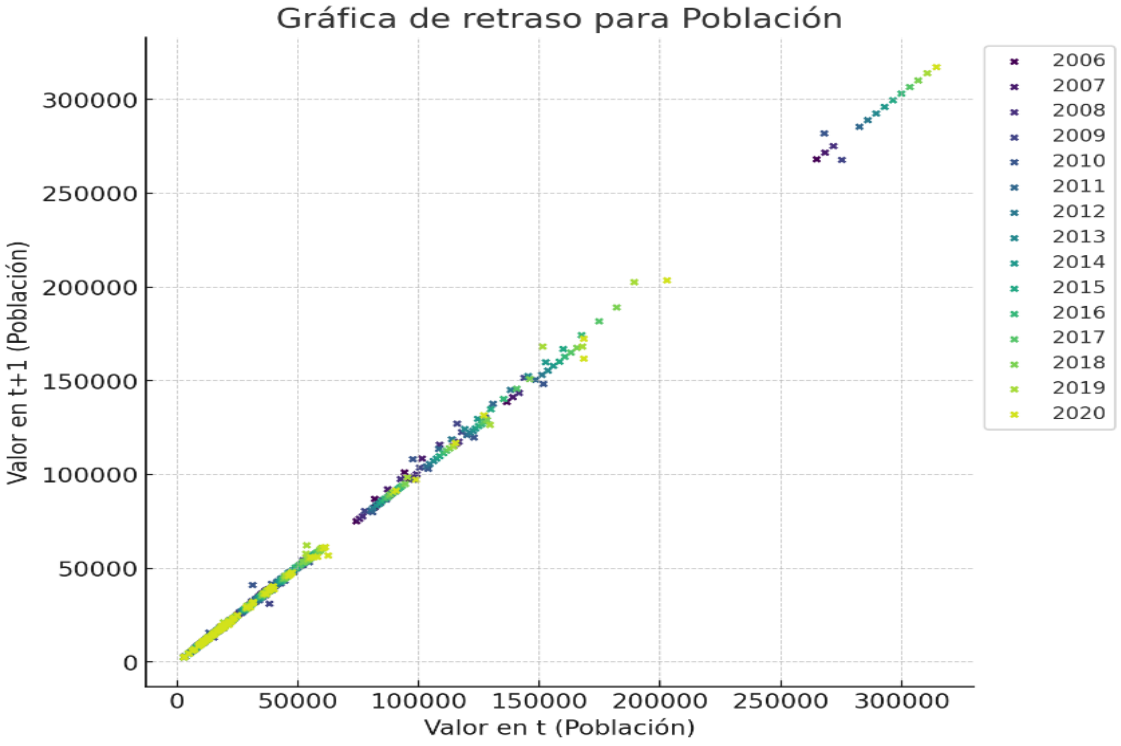
Gráfica 42

Gráfica de retraso para Cabezas de ganado bovino (Objetivo)



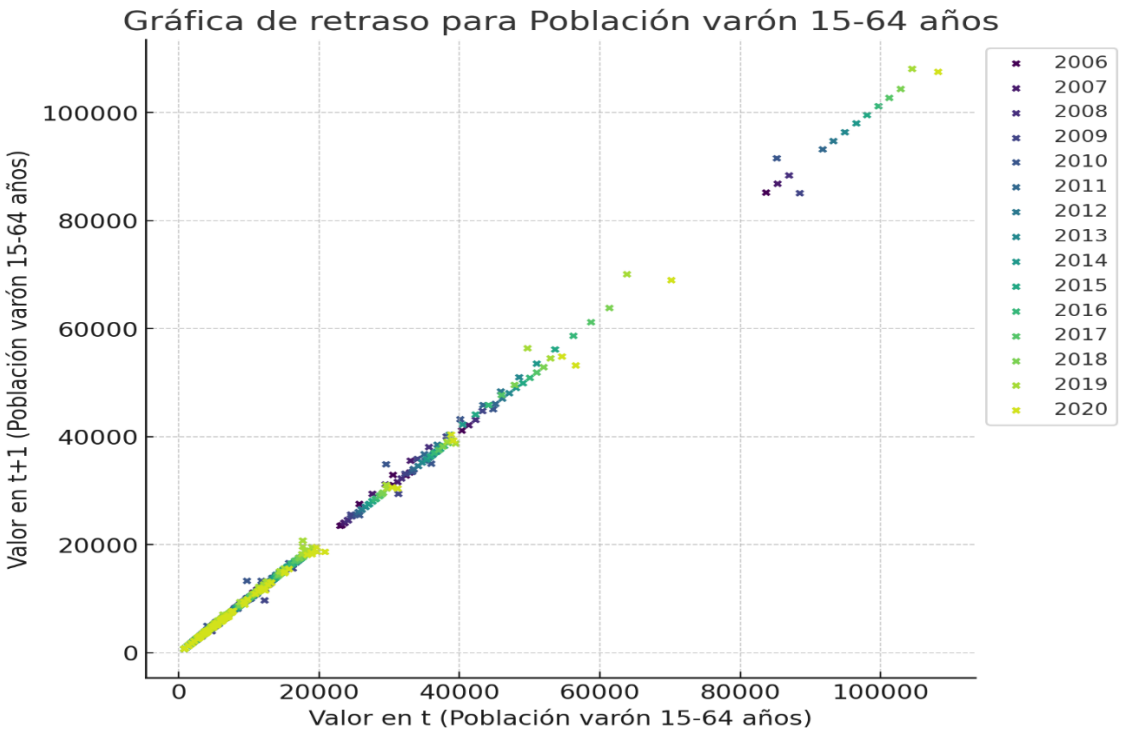
Gráfica 43

Gráfica de retraso para Población



Gráfica 44

Gráfica de retraso para Población varón 15-64 años



Análisis de Variables Específicas

- Variables con Fuerte Correlación Temporal: Por ejemplo, "Producción en canal (Ton)" y "Producción en pie (Ton)" podrían mostrar una fuerte correlación temporal.
- Variables con Correlación Temporal Débil: Algunas variables pueden mostrar una correlación débil o negativa sin una tendencia clara.

Las gráficas de retraso son una herramienta complementaria valiosa en el análisis de series temporales, útiles para identificar la autocorrelación. Junto con las pruebas de Dickey-Fuller, las gráficas de correlación, las gráficas de densidad y las gráficas de series de tiempo, proporcionan una comprensión completa de las propiedades estadísticas y temporales de las variables, siendo fundamentales para el análisis y modelado adecuado de las series temporales.

4.12 Mapa coroplético

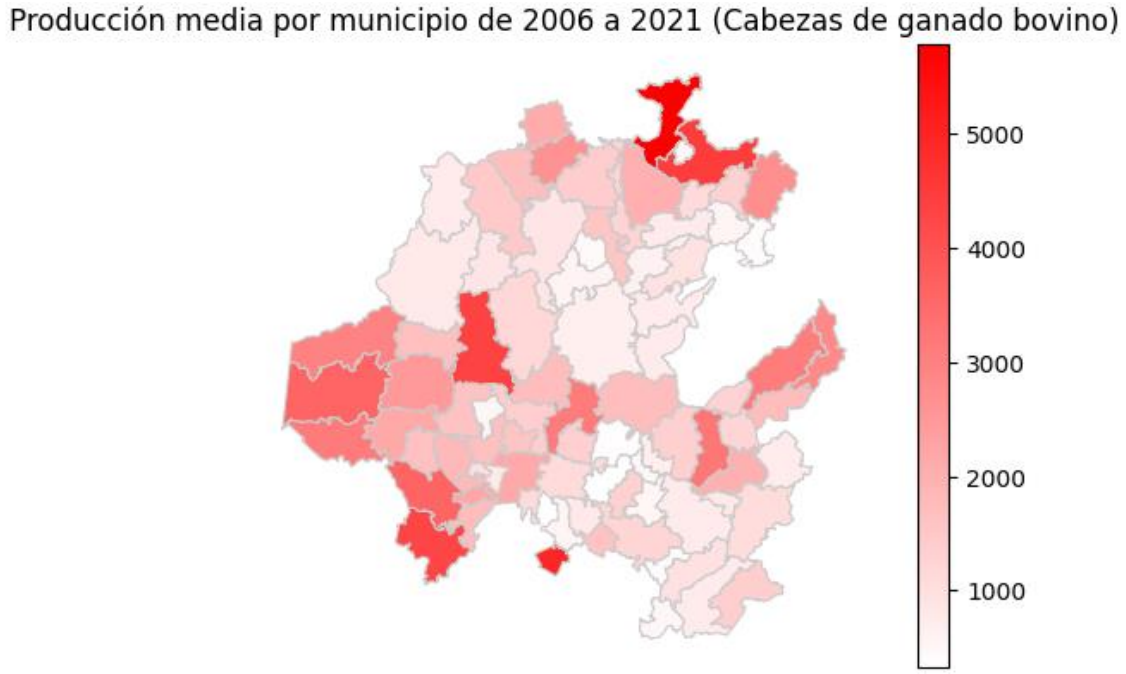
El mapa coroplético es una poderosa herramienta de visualización que representa variables en un mapa geográfico, permitiendo una comprensión más clara de cómo una variable específica, (Slocum, 2005) en este caso, "Cabezas de ganado bovino (Objetivo)", está distribuida geográficamente.

Interpretación

- Visualización Geográfica: El mapa muestra la distribución de "Cabezas de ganado bovino (Objetivo)" en los municipios durante un período específico. Los tonos más oscuros representan mayores cantidades de ganado, mientras que los tonos más claros representan cantidades menores.
- Análisis Regional: Identifica patrones regionales, como áreas con alta o baja producción de ganado, y revela relaciones geográficas.
- Contexto Temporal: Al calcular la mediana para un rango de años, el mapa refleja una vista agregada a lo largo del tiempo, útil para identificar tendencias estables.

Figura 6

Mapa coroplético de la producción media por municipio del 2006 a 2021



Nota: Los municipios con mayor tonalidad roja representan mayor producción.

4.13 Modelo para Pronóstico

La construcción de un modelo de pronóstico para predecir la variable objetivo, en este caso, "Cabezas de ganado bovino," es un proceso complejo que involucra múltiples etapas. A continuación, se resume cada una de estas etapas:

Propósito del Modelo

Objetivo: Predecir la variable objetivo "Cabezas de ganado bovino" utilizando variables seleccionadas basadas en un análisis multivariable previo, considerando la estacionalidad y el análisis de residuos.

Preparación y Limpieza de Datos

- Carga de Datos: Carga de los datos desde un archivo Excel y selección de columnas relevantes.
- Limpieza de Datos: Transformaciones para limpiar y convertir los datos en formato numérico. (Galety et al., 2023).
- División de Datos: División en conjuntos de entrenamiento y prueba, seleccionando municipios aleatorios. (Galety et al., 2023).

Selección y Entrenamiento del Modelo

- Transformaciones: Aplicación de transformaciones como OneHotEncoding, estandarización y Box-Cox.
- Feature Engineering: Se ocupa esta técnica con polinomial features para tener más datos.
- Modelos Ensamblados: Uso de un ensamble de modelos, incluyendo Gradient Boosting, Random Forest, Elastic Net, XGBoost y Bagging con Decision Trees.
- Optimización de Hiperparámetros: Búsqueda Bayesiana para encontrar los mejores hiperparámetros, utilizando validación cruzada y minimizando el MAE, así como

GroupKfold para dividir los datos de entrenamiento y validación cruzada con ayuda de búsqueda bayesiana.

- Ajuste y Predicción: Ajuste del modelo al conjunto de entrenamiento y realización de predicciones en el conjunto de prueba.

Transformaciones

Estas son algunas de las técnicas clave empleadas en el preprocesamiento de datos:

- OneHotEncoding: Esta técnica convierte las variables categóricas en un formato numérico que puede ser proporcionado a los modelos de aprendizaje automático. En este caso, se aplica a la columna 'Municipio' para convertir los nombres de los municipios en vectores binarios, donde cada columna representa un municipio diferente. Esta técnica se utiliza para evitar la asignación de un orden o jerarquía a las categorías y para permitir que los modelos de aprendizaje automático utilicen la información de todas las categorías. (Vanderplas & VanderPlas, 2016)
- Estandarización: La estandarización se realiza para escalar las variables numéricas, de modo que tengan una media de 0 y una desviación estándar de 1. Esto es fundamental para muchos algoritmos de aprendizaje automático que son sensibles a la escala de las características.

  La puntuación estándar de una muestra x se calcula como:

$$z = \frac{(x - u)}{s}$$

  donde u es la media de las muestras de entrenamiento o cero si with_mean=False, y s es la desviación estándar de las muestras de entrenamiento o uno si with_std=False. (Sklearn.preprocessing.StandardScaler, s. f.)
- Transformación Box-Cox: Esta transformación se aplica a ciertas características numéricas para hacer que los datos sean más cercanos a la normalidad. Esto puede mejorar el rendimiento de algunos modelos al estabilizar la varianza y hacer que los datos sean más simétricos.

  La transformada de Box-Cox viene dada por:

$$y(\lambda) = \begin{cases} \left(\frac{x^{\lambda} - 1}{\lambda}\right), & si\ \lambda \neq 0 \\ log(x), & si\ \lambda = 0 \end{cases}$$

boxcox requiere que los datos de entrada sean positivos. A veces, una transformación Box-Cox proporciona un parámetro de desplazamiento para conseguirlo; boxcox no lo hace. Dicho parámetro de desplazamiento equivale a añadir una constante positiva a x antes de llamar a boxcox.

Los límites de confianza devueltos cuando se proporciona alfa dan el intervalo donde:

$$llf(\hat{\lambda}) - llf(\lambda) < \frac{1}{2} x^2(1 - \alpha, 1),$$

con llf la función log-verosimilitud y x^2 la función chi-cuadrado. (G. E. P. Box & D. R. Cox, 1964)

Feature Engineering

Polynomial Features: La creación de características polinomiales es una forma de aumentar la complejidad del modelo al añadir potencias de las características existentes. (Kuhn & Johnson, 2021) En este caso, se utiliza un grado de 1, lo que significa que no se están añadiendo características polinómicas adicionales. Sin embargo, el código permite la flexibilidad de experimentar con grados polinómicos más altos si se desea.

Modelos Ensamblados

- Ensamblaje de Modelos: En lugar de utilizar un solo modelo, el código utiliza una combinación de varios modelos, incluyendo Gradient Boosting, Random Forest, Elastic Net, XGBoost y Bagging con Decision Trees. Cada uno de estos modelos tiene sus propias fortalezas y debilidades, y al combinarlos, el modelo ensamblado puede aprovechar lo mejor de cada uno.
- Stacking: En este caso, se utiliza el ensamble de Stacking, donde las predicciones de varios modelos base se utilizan como entrada para un modelo meta que realiza la predicción final. (Y. Cai et al., 2023)

Validación Cruzada (Cross Validation)

La validación cruzada es una técnica esencial para evaluar el rendimiento de un modelo de aprendizaje automático. Ayuda a entender cómo el modelo se comportará en datos no vistos y permite una evaluación más robusta comparada con una simple división de entrenamiento/prueba. (Comp.ai.neural-nets FAQ, Part 3 of 7: GeneralizationSection - What are cross-validation and bootstrapping?, s. f.)

Optimización de Hiperparámetros

- Búsqueda Bayesiana: A diferencia de una búsqueda en cuadrícula que prueba todas las combinaciones posibles de hiperparámetros, la búsqueda Bayesiana utiliza la información de las iteraciones anteriores para elegir los siguientes hiperparámetros a probar. Esto puede llevar a una búsqueda más eficiente y rápida de los mejores hiperparámetros, la estrategia basada en el teorema de bayes dónde:

  $$P(H \mid \text{datos}) = \left(\frac{P(datos \mid \text{H})\, P(H)}{P\ datos}\right)$$

  Esto da como resultado P(H | datos) . La función "P()" es una forma de indicar la probabilidad y "|" quiere decir "dado que". Por lo tanto, P(H | datos) indica la probabilidad de que una hipótesis sea verdadera a partir de los datos que hemos observado.. (Google, 2023).

- GroupKFold: En este caso, se utiliza una variante de la validación cruzada conocida como GroupKFold. La idea es dividir los datos en k grupos (o "folds") y entrenar el modelo en k-1 grupos mientras se valida en el grupo restante. (Hastie et al., 2013) Esto se repite k veces, con un grupo diferente como conjunto de validación cada vez. La métrica de rendimiento (en este caso, MAE) se calcula en cada iteración y luego se promedia para obtener una estimación final del rendimiento del modelo. Esta técnica de validación cruzada asegura que los mismos grupos no aparezcan en los conjuntos de entrenamiento y prueba. En este caso, se utiliza para dividir los

datos por municipio, asegurando que los datos de un municipio no estén presentes tanto en el entrenamiento como en la validación. (Breiman, L. & Spector P., 1992).

- Grupos en la Validación Cruzada: En el contexto de estos datos, los municipios se utilizan como grupos. Esto significa que todos los datos de un municipio en particular estarán en el mismo grupo, ya sea en el conjunto de entrenamiento o en el conjunto de validación, pero no en ambos. (Rao R. Fung G. & Rosales R., 2008). Esto es útil cuando hay una estructura de grupo en los datos que se quiere respetar durante la validación cruzada, como en este caso, donde los datos están agrupados por municipio. (Kohavi R., 1995)

- La validación cruzada con GroupKFold es particularmente útil en este contexto porque asegura que el modelo se evalúa de manera justa, teniendo en cuenta la estructura de grupo en los datos. Además, como se realiza dentro de la búsqueda Bayesiana de hiperparámetros, también ayuda a seleccionar los hiperparámetros que producen un modelo que generaliza bien a nuevos datos. (James et al., 2013).

Ajuste y Predicción

Entrenamiento y Prueba: Después de seleccionar los mejores hiperparámetros, el modelo se entrena en todo el conjunto de entrenamiento y se realiza una predicción en el conjunto de prueba. Las métricas como el MAE son útiles para evaluar qué tan bien el modelo está realizando predicciones en datos no vistos. (Breiman L. & Spector P., 1992).

Dónde se ocupan los modelos:

Gradient Boosting

- ¿Por qué?: Gradient Boosting es un poderoso algoritmo de boosting que construye árboles en una etapa a la vez, donde cada árbol corrige los errores del anterior.

- Ventajas: Capacidad para capturar interacciones no lineales y manejar diferentes tipos de datos. (Natekin A. & Knoll A., 2013).

Random Forest

- ¿Por qué?: Random Forest es un algoritmo de bagging que construye múltiples árboles de decisión y promedia sus predicciones.
- Ventajas: Reduce la varianza en comparación con un solo árbol de decisión, lo que lo hace más robusto a los datos ruidosos y outliers. (Breiman, 1984).

Elastic Net

- ¿Por qué?: Elastic Net es una técnica de regresión regularizada que combina las penalizaciones L1 y L2.
- Ventajas: Puede seleccionar automáticamente características útiles y manejar multicolinealidad. (Zou & Hastie, 2005)

XGBoost

- ¿Por qué?: XGBoost es una implementación optimizada del algoritmo Gradient Boosting con regularización adicional.
- Ventajas: Rápido, escalable y capaz de modelar relaciones complejas. (Mitchell R & Frank E., 2017)

Bagging con Decision Trees

- ¿Por qué?: Bagging es una técnica de ensamblado que reduce la varianza al entrenar múltiples modelos en subconjuntos aleatorios de datos y promediar sus predicciones.
- Ventajas: Mejora la robustez y reduce el sobreajuste. (Leo Breiman, 1996).

Estrategia de Ensamblado

- ¿Por qué Ensamblar?: El ensamblaje combina las predicciones de múltiples modelos para crear un modelo compuesto que a menudo tiene un rendimiento superior a cualquiera de los modelos individuales.
- Ventajas del Ensamblado: La combinación de diferentes modelos puede capturar diferentes aspectos de los datos, lo que aumenta la robustez y mejora la generalización. Los modelos ensamblados suelen ser menos propensos al sobreajuste y pueden proporcionar una mejor precisión.

El uso de estos modelos y su ensamblaje representa una estrategia sofisticada y bien pensada para crear un modelo que es capaz de capturar relaciones complejas en los datos, mientras se mantiene robusto y preciso. La diversidad de los modelos en el ensamblaje permite una representación más rica de los datos y aumenta la probabilidad de un buen rendimiento en datos no vistos. (Y. Cai et al., 2023)

En este caso se ocupan los siguientes hiperparámetros para justar el modelo dónde bagging trabaja con valores predeterminados:

Gradient Boosting (model__gb__)

- max_depth: Controla la profundidad máxima de los árboles individuales. Afecta la complejidad del modelo y la capacidad de ajustarse a los datos.
- min_samples_split: Número mínimo de muestras requeridas para dividir un nodo interno. Ayuda a prevenir el sobreajuste.

Random Forest (model__rf__)

- n_estimators: Número de árboles en el bosque. Más árboles pueden aumentar la precisión, pero también el tiempo de cómputo.

- max_features: Número de características a considerar en cada división. Controla la diversidad entre árboles y, por ende, la robustez del modelo.

Elastic Net (model__en__)

- alpha: Parámetro de regularización que combina las penalizaciones L1 y L2. Ayuda a prevenir el sobreajuste.
- l1_ratio: Proporción de la penalización L1 en la regularización total. Controla la cantidad de características seleccionadas (sparse).

XGBoost (model__xgb__)

- learning_rate: Tasa de aprendizaje que controla el aporte de cada árbol. Valores bajos pueden mejorar la precisión, pero requieren más árboles.
- max_depth: Profundidad máxima de los árboles. Similar al Gradient Boosting, controla la complejidad.
- n_estimators: Número de árboles a construir.
- min_child_weight: Controla el sobreajuste al regular la creación de nodos hijo.
- gamma: Parámetro de regularización que controla el ajuste de los árboles.
- subsample: Proporción de datos de entrenamiento utilizados en cada árbol. Ayuda a prevenir el sobreajuste.
- colsample_bytree: Proporción de características utilizadas en cada árbol. Aumenta la diversidad entre árboles.

La elección de estos hiperparámetros está alineada con el objetivo de crear un modelo que pueda capturar las relaciones en los datos sin sobreajustar. La sintonización de estos parámetros mediante una búsqueda Bayesiana permite encontrar la combinación óptima que minimice el error en las predicciones, asegurando que el modelo sea lo suficientemente complejo para ajustarse a los datos, pero no tanto como para perder su capacidad de generalización. (Pedregosa F et al., 2011).

Código: Ver ANEXO A: CÓDIGO 1.

Resultado:

Mejor parámetro encontrado	Valor
model__en__alpha	**0.0100018**
model__en__l1_ratio	**1**
model__gb__max_depth	**2**
model__gb__min_samples_split	**5**
model__rf__max_features	**log2**
model__rf__n_estimators	**50**
model__xgb__colsample_bytree	**0.9**
model__xgb__gamma	**0.2**
model__xgb__learning_rate	**0.046416**
model__xgb__max_depth	**3**
model__xgb__min_child_weight	**3**
model__xgb__n_estimators	**100**
model__xgb__subsample	**0.5**

Lowest MAE found**: 82.56182417338664**

MAE**: 35.51917832583811**

Interpretación de los Resultados

- Mejores Parámetros Encontrados: Parámetros que ofrecen el mejor rendimiento en la validación cruzada.
- Error Absoluto Medio (MAE): Representa la diferencia promedio entre las predicciones y los valores reales.
- Predicciones para Cada Municipio: Predicciones para cada municipio, considerando una ventana temporal.
- Exportación de Predicciones: Las predicciones se exportan a un archivo Excel para análisis o visualización adicionales.
- Lowest MAE found: Este valor se obtiene después de realizar una búsqueda Bayesiana para encontrar el mejor conjunto de hiperparámetros para un modelo apilado (StackingRegressor) que contiene GradientBoostingRegressor, RandomForestRegressor, ElasticNet, XGBRegressor y BaggingRegressor. La búsqueda Bayesiana utiliza validación cruzada con GroupKFold en 10 divisiones, y el MAE (Mean Absolute Error) es la métrica de evaluación. El "Lowest MAE

found" representa el mejor (menor) error absoluto medio negativo encontrado en el proceso de búsqueda. Es el error en el conjunto de validación para el mejor conjunto de hiperparámetros y refleja cómo se desempeña el modelo en datos no vistos con ese conjunto particular de hiperparámetros.

La interpretación de los resultados, incluyendo la evaluación del MAE y la exportación de las predicciones, proporciona una visión clara del rendimiento del modelo y su aplicabilidad para tomar decisiones basadas en datos en el contexto de la producción de ganado bovino.

Este modelo podría utilizarse para guiar estrategias de producción, planificación de recursos y toma de decisiones en la industria ganadera, aprovechando la rica información contenida en las variables seleccionadas y las técnicas de modelado aplicadas.

4.14 Análisis de Residuos y Coeficiente de Determinación *R*2

En el proceso de modelado, es esencial evaluar no solo la precisión de las predicciones sino también la estructura de los errores o residuos. En este contexto, se lleva a cabo un análisis específico para evaluar si hay alguna estructura sistemática en los errores de predicción, lo cual podría indicar que el modelo no está capturando completamente la información en los datos. (Glantz & Slinker, 2000)

Cálculo del Coeficiente de Determinación *R*2

- Residuos: Diferencia entre los valores observados y los predichos por el modelo.
- Valores Predichos: Los valores predichos por el modelo.
- Regresión Lineal: Ajuste de una regresión lineal entre los residuos y los valores predichos.
- Coeficiente de Determinación *R*2: Medida que indica cuánta variación en los residuos es explicada por los valores predichos.

El coeficiente de determinación R^2 es: 0.0003037315900558113

Interpretación

- Resultado Cercano a Cero: El coeficiente de determinación $R2$ es muy cercano a cero, lo cual indica que los valores predichos no explican prácticamente ninguna variación en los residuos.
- Ausencia de Estructura Sistemática: La falta de correlación entre los residuos y los valores predichos sugiere que los errores de predicción no tienen una estructura sistemática.
- Ausencia de Heterocedasticidad: La heterocedasticidad se refiere a la variabilidad cambiante de los errores de un modelo a lo largo de los niveles de alguna variable explicativa. La ausencia de heterocedasticidad es una buena señal, ya que significa que la varianza de los residuos es aproximadamente constante en relación con los valores predichos.

La evaluación de los residuos y el cálculo del coeficiente de determinación $R2$ proporcionan evidencia importante sobre la calidad y confiabilidad del modelo. (Glantz & Slinker, 2000)

En este caso, el análisis mostró que no hay evidencia significativa de heterocedasticidad, y los errores de predicción no tienen una estructura sistemática que el modelo no esté capturando. Esto fortalece la confianza en el modelo, cumpliendo con uno de los supuestos clásicos de la regresión lineal.

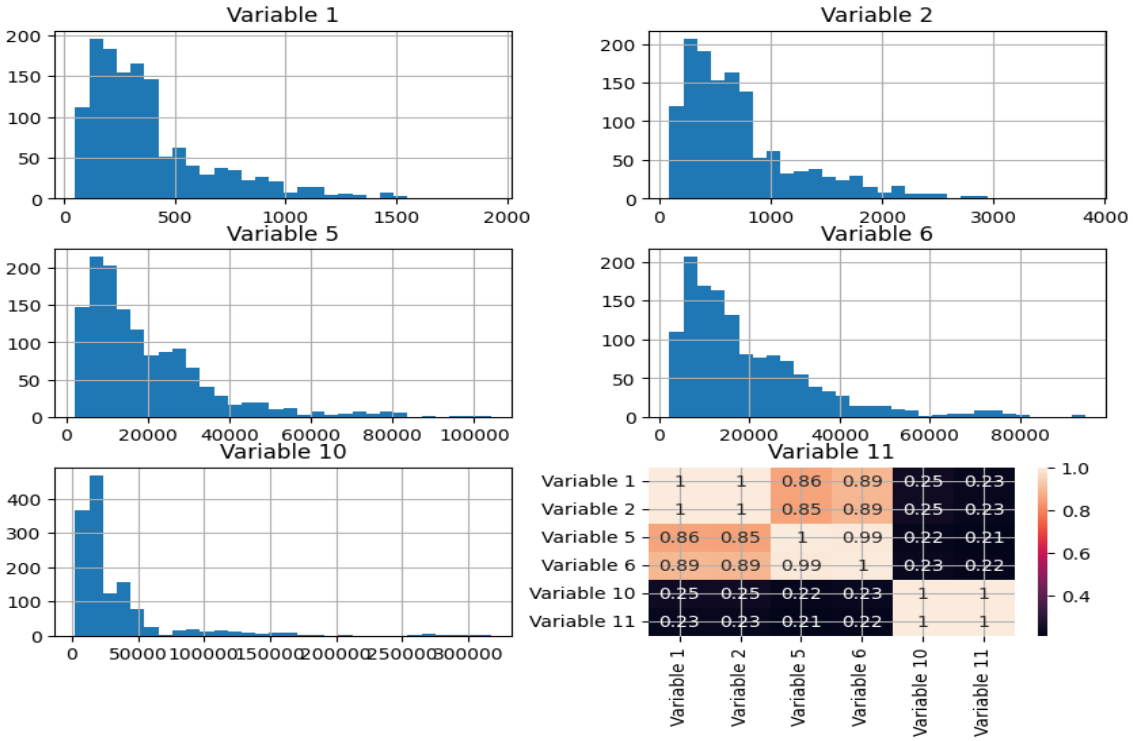
La ausencia de sesgo en los errores y la conformidad con los supuestos de la regresión sugieren que el modelo es sólido y confiable para el propósito de pronosticar la variable objetivo, en este caso, "Cabezas de ganado bovino." La integridad del modelo es crucial para garantizar que las predicciones sean válidas y útiles en el contexto de toma de decisiones en la industria ganadera. (Glantz & Slinker, 2000)

4.15 Verificación

Para verificar que efectivamente se cumplen el supuesto de que las variables trabajadas afectan a la variable objetivo se toma en cuenta los histogramas de las variables numéricas y su matriz de correlación, verificando que como ya se había descrito en lo anterior dónde se hace la matriz de covarianzas y la parte que se presenta a continuación concuerdan.

Figura 7

Gráficos y matriz de resultados



Nota: las gráficas descritas en la figura muestran que los resultados de salida concuerdan con los datos de entrada, como se muestra en la matriz de covarianzas y en los gráficos posteriores.

4.16 Análisis de Importancia de Características

La importancia de las características es una medida que describe el impacto de cada característica en la predicción de la variable objetivo en un modelo de aprendizaje automático. Permite entender cuáles son las variables más relevantes en el modelo y cómo contribuyen a las predicciones.

En este caso, se utiliza un RandomForestRegressor para entrenar el modelo y obtener las importancias de las características después de aplicar transformaciones y preprocesamientos.

Proceso de Análisis:

1. Entrenamiento del Pipeline: Entrena el pipeline que incluye todas las transformaciones y preprocesamientos necesarios.
2. Obtención de Nombres de Características: Extrae los nombres de las características después del preprocesamiento, incluyendo las categóricas y numéricas transformadas.
3. Entrenamiento de RandomForestRegressor: Entrena un RandomForestRegressor utilizando los datos preprocesados.
4. Obtención de Importancias de Características: Extrae las importancias de las características del modelo entrenado.
5. Impresión de Importancias: Imprime las importancias para cada característica.

Resultados:

Municipio	Importancia
Acaxochitlán	**0.7545**
Actopan	**0.2408**
Agua Blanca de Iturbide	**0.0002**
Ajacuba	**0.0007**
Alfajayucan	**0.0006**
Almoloya	**0.0031**

Interpretación:

- Variabilidad en Importancias: Las importancias son significativamente diferentes entre las características, lo que refleja cómo el preprocesamiento y la transformación adecuados pueden cambiar la interpretación de la importancia en el modelo.

- Municipio_Acaxochitlán: Tiene una importancia muy alta, lo que indica que esta variable categórica transformada juega un papel crucial en la predicción de la variable objetivo.
- Municipio_Agua Blanca de Iturbide: Tiene una importancia relativamente baja, lo que sugiere que esta categoría específica no tiene un impacto significativo en la predicción.

El análisis de la importancia de las características es una herramienta poderosa para comprender el funcionamiento interno de un modelo de aprendizaje automático. En este caso, revela cómo diferentes municipios tienen diferentes niveles de importancia en el modelo, lo que podría ser útil para tomar decisiones estratégicas en el contexto de la producción de ganado bovino.

La capacidad de identificar y comprender la importancia de diferentes características puede guiar la toma de decisiones, la selección de características, y proporcionar intuiciones útiles para mejorar y afinar el modelo.

4.17 Gráfica de Residuales

La gráfica de residuales es una herramienta importante para diagnosticar y comprender el comportamiento de los errores en un modelo de regresión. Permite visualizar la heterocedasticidad en los datos, es decir, la constancia o no constancia de la variabilidad de los errores en diferentes niveles de las variables explicativas, o que estos se ajusten o no a una distribución normal. (Kutner et al., 2004).

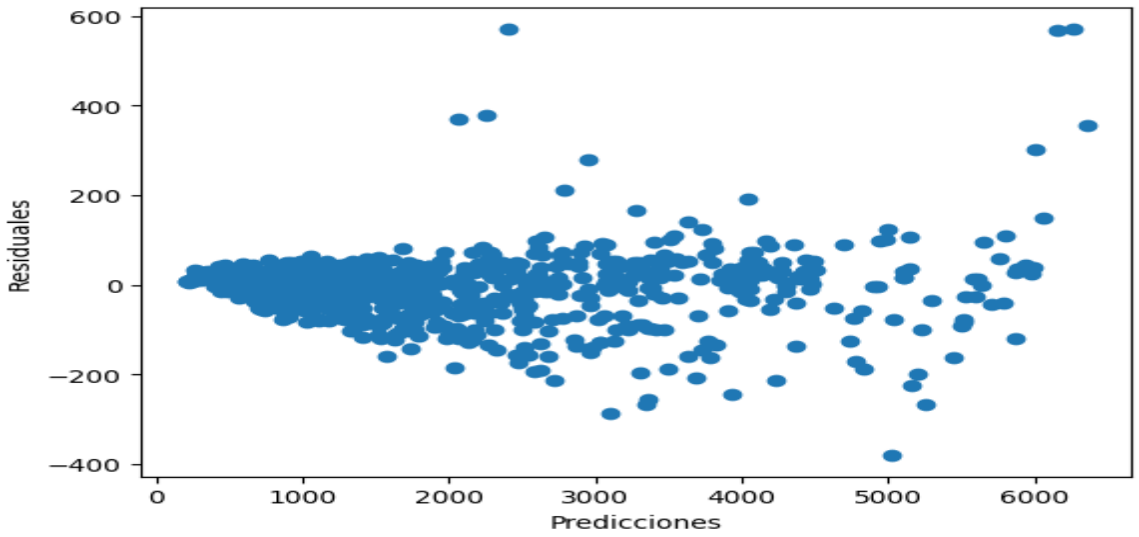
La heterocedasticidad se refiere a una situación en la que la variabilidad de los errores (o residuales) no es constante en todos los niveles de las variables explicativas. Si la variabilidad de los errores es constante en diferentes niveles de las variables explicativas, entonces se dice que los errores son homocedásticos. Si la variabilidad de los errores no es constante en diferentes niveles de las variables explicativas, entonces se dice que los errores son heterocedásticos. La heterocedasticidad puede ser un problema en la regresión

lineal, ya que puede afectar la precisión de las estimaciones de los parámetros del modelo. (Greene, 2017).

La distribución normal es una distribución de probabilidad continua que se utiliza comúnmente en estadística. Muchos modelos estadísticos, incluyendo la regresión lineal, asumen que los errores siguen una distribución normal. Si los errores no siguen una distribución normal, entonces los resultados del modelo pueden ser sesgados o inexactos. (Montgomery et al., 2021).

Primero se calcula las predicciones usando el modelo ajustado (pipe) y luego calcula los residuales restando las predicciones de los valores observados de la variable objetivo. Posteriormente, se trazan los residuales frente a las predicciones.

Gráfica 45
Gráfica de residuos



Interpretación

- Eje X (Predicciones): Representa los valores predichos por el modelo para la variable objetivo.
- Eje Y (Residuales): Representa la diferencia entre los valores observados y predichos (residuales).

Análisis

- Sin Heterocedasticidad: Si los puntos en la gráfica están dispersos de manera aleatoria y no muestran un patrón claro, esto indica que no hay heterocedasticidad significativa. En otras palabras, la variabilidad de los errores es constante en todos los niveles de las variables explicativas.
- Presencia de Heterocedasticidad: Si los puntos en la gráfica muestran un patrón (por ejemplo, una forma de embudo), esto podría indicar la presencia de heterocedasticidad. En ese caso, la variabilidad de los errores cambia en diferentes niveles de las variables explicativas.

Para este caso observamos una posible heterocedasticidad en los datos, sin embargo, con el r2 y el trabajo con bagging, que es un modelo que trabaja muy bien con las colas, esto puede que no sea un problema como ya lo seña el r2. (Koenker R., 1981).

Para poder observar mejor si la heterocedasticidad se cumple podemos ocupar otro test y después visualizar nuestros datos en una gráfica de caja, y ver cómo se están haciendo las transformaciones para, ello empezamos con la prueba de Breusch Pagan:

4.18 Prueba de Breusch-Pagan para Heterocedasticidad

La prueba de Breusch-Pagan es una herramienta estadística diseñada para detectar la presencia de heterocedasticidad en un modelo de regresión. (Breusch T. S. & Pagan A. R., 1979).

Resultados:

Métrica	Valor
LM Statistic	**311.6424702**
LM-Test p-value	**2.54E-27**
F-Statistic	**4.464397914**
F-Test p-value	**1.34E-32**

- Estadística LM: 311.6425. Es una medida que cuantifica la evidencia en contra de la hipótesis nula de homocedasticidad (variabilidad constante de los errores).
- p-valor de la Prueba LM: $2.54\times10-272.54\times10-27$. Es un valor extremadamente pequeño, lo que indica un fuerte rechazo de la hipótesis nula.
- Estadística F: 4.4644. Similar a la estadística LM, pero basada en una prueba F en lugar de una prueba Chi-cuadrado.
- p-valor de la Prueba F: $1.34\times10-321.34\times10-32$. También un valor extremadamente pequeño, lo que refuerza el rechazo de la hipótesis nula.

Interpretación:

Los resultados de la prueba de Breusch-Pagan indican la presencia de heterocedasticidad en el modelo. Los p-valores extremadamente bajos para ambas pruebas (LM y F) sugieren que hay suficiente evidencia para rechazar la hipótesis nula de homocedasticidad. (Greene W. H., 2002).

La heterocedasticidad detectada por la prueba de Breusch-Pagan podría ser una preocupación, ya que viola uno de los supuestos clave de los modelos lineales. La presencia de heterocedasticidad puede afectar la eficiencia y la validez de las inferencias estadísticas realizadas a partir del modelo. (Greene W. H., 2002).

4.19 Análisis de Transformaciones y Heterocedasticidad

Ahora bien, esto entra en contradicción con la prueba r2, para ello ahora se verá cómo se están haciendo las trasformaciones:

Para ello se imprimen y se describen los variables antes y después de la transformación standarscaler y box cox, algoritmos que ayudan a normalizar la varianza y ayudan a reducir el impacto de la heterocedasticidad:

Primero observamos si no existen valores infinitos, ya que esto es necesario para que box cox y standarscaler funcionen o nulos con:

```
# Valores NaN
print(data_train_transformed.isnull().sum())
# Valores Infinitos
print(np.isinf(data_train_transformed).sum())
```

Valores NaN

Variable 1	**0**
Variable 2	**0**
Variable 5	**0**
Variable 6	**0**
Variable 9 (Objetivo)	**0**
Variable 10	**0**
Variable 11	**0**

Valores infinitos

Variable 1	**0**
Variable 2	**0**
Variable 5	**0**
Variable 6	**0**
Variable 9 (Objetivo)	**0**
Variable 10	**0**
Variable 11	**0**

Dónde claramente no existen datos de esta manera para proceder a ver si estos tienen valores negativos con:

```
# Verificar si hay valores no positivos
print((data_train <= 0).sum())
```

Valores Negativos

Variable 1	**0**
Variable 2	**0**
Variable 5	**0**
Variable 6	**0**
Variable 9 (Objetivo)	**0**
Variable 10	**0**
Variable 11	**0**

Y se observa que de igual forma no se tienen, ahora bien otra forma sería ver si existen datos extra que no se están contemplando con:

```
# Verificar si hay valores atípicos
for feature in num_features:
	print(f"{feature}  tiene  {data_train[feature].nunique()}  valores únicos.")
```

Variable 1 tiene 1267 valores únicos.

Variable 2 tiene 1268 valores únicos.

Variable 5 tiene 1279 valores únicos.

Variable 6 tiene 1280 valores únicos.

Variable 10 tiene 1279 valores únicos.

Variable 11 tiene 1278 valores únicos.

Dónde se sigue viendo que los datos siguen estando bien para lo que ahora si se procede con la impresión de los datos transformados con:

```
# Imprime valores antes de la transformación
print(data_train.head())
# Imprime valores despues de la transformación
print(data_train_transformed.head())
```

Datos antes de la transformación

Variable 1	Variable 2	Variable 5	Variable 6	Variable 9 (Objetivo)	Variable 10	Variable 11
1102.68	2131.75	33403.93	31127.71	6837	19685	5438.6
449.08	860.06	13798.09	12702.5	2974	38345.2	10213.8
514.59	961.39	15961.12	18422.73	2307	50810.4	14776.4
526.45	989.18	16635.43	16340.01	2202	9044.8	2477.8
287.01	536.5	8862.57	10213.51	1298	16107.8	4748.6

Datos después de la transformación

Variable 1	Variable 2	Variable 5	Variable 6	Variable 9 (Objetivo)	Variable 10	Variable 11
1.860398	2.157483	3.770584	3.72035	6837	3.406803	2.633583
1.500308	1.755846	3.178088	3.126563	2974	3.87033	2.993783
1.551794	1.802223	3.270351	3.363215	2307	4.080356	3.221237
1.560514	1.814214	3.296937	3.285405	2202	2.921634	2.229507
1.338359	1.567774	2.909681	2.993766	1298	3.276214	2.560398

Y ahora se describen estos datos con:

```
# Imprime la descripción de los datos transformados
print(data_train_transformed.describe())
```

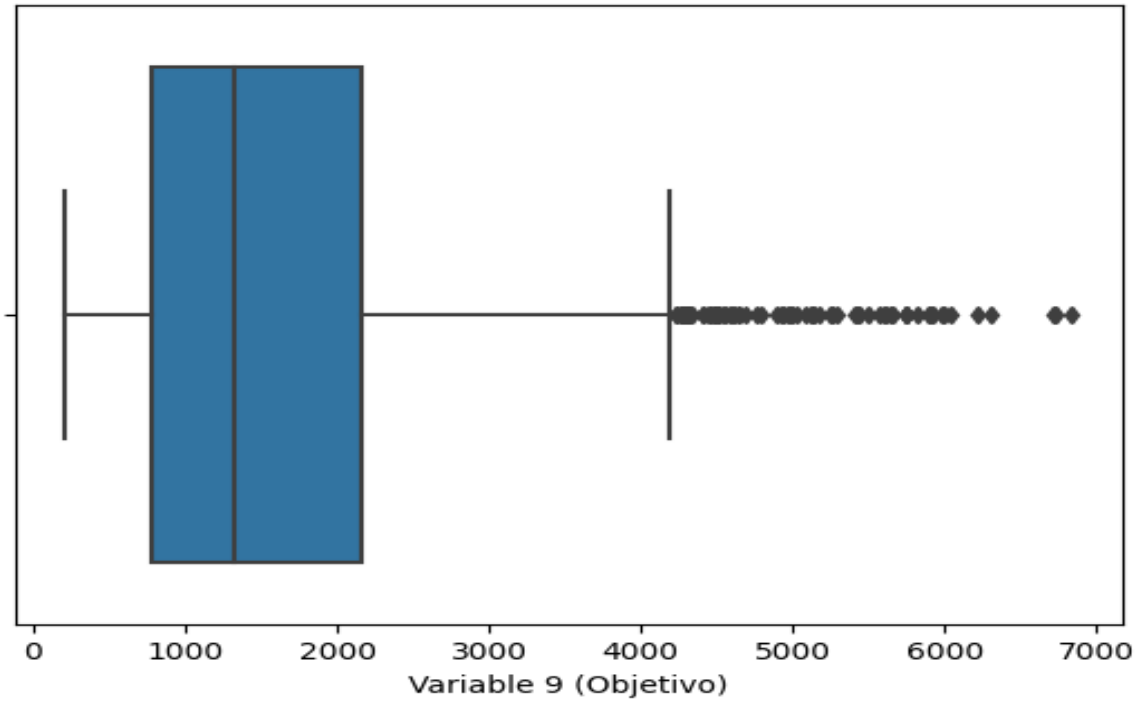
Estadísticas	Variable 1	Variable 2	Variable 5	Variable 6	Variable 9 (Objetivo)	Variable 10	Variable 11
count	1280	1280	1280	1280	1280	1280	1280
mean	1.377041	1.617494	3.264373	3.273078	1682.54297	3.530834	2.784497
std	0.248489	0.275471	0.483845	0.471107	1238.32452	0.601945	0.518985
min	0.807482	0.984597	2.189124	2.197467	212	2.247895	1.686272
25%	1.187051	1.410871	2.898114	2.921754	774.75	3.114774	2.439247
50%	1.368804	1.607561	3.241246	3.245211	1323	3.378275	2.660799
75%	1.542611	1.802174	3.617275	3.623283	2156.25	3.863925	3.050809
max	2.110166	2.446299	4.657727	4.575742	6837	5.687126	4.68967

Para lo que se llega observar que los datos de la variable objetivo tienen un problema y es que tienen una gran variación entre sus datos para observar esto podemos graficar un gráfico de caja:

4.20 Grafica de caja de la variable objetivo

Gráfica 46

Gráfica de caja de la variable objetivo



Nota: Muestra los valores atípicos en el tercer cuartil.

Interpretación:

- Mediana: La línea central en el cuadro representa la mediana de la variable objetivo, que es el valor medio de la distribución.
- Cuartiles: Las aristas superior e inferior del cuadro representan el tercer y primer cuartil de la distribución, respectivamente. La longitud del cuadro (la distancia entre estos dos cuartiles) se denomina rango intercuartílico y da una idea de la dispersión de los datos.
- Bigotes: Los "bigotes" que se extienden desde el cuadro hasta los puntos extremos representan el rango dentro del cual caen la mayoría de los datos.

- Valores Atípicos: Los puntos fuera de los bigotes son considerados valores atípicos y representan observaciones que caen fuera del patrón general de distribución.

Dónde se ve que efectivamente estos datos varían bastante con respecto a su media, explicando la heterocedasticidad de la prueba antes hecha, pero ahora bien haciendo el análisis de los datos se ve que estos tienen una gran variabilidad debido a que son muy pocos datos con respecto a cada municipio, ya que se tienen 84 municipios, con 16 datos (uno por año) y 7 variables diferentes dándonos, 112 datos por municipio, estos son pocos datos en este caso, ya que se analizan un total 9408 datos, y existe un gran variabilidad, cosa que en la vida real, es cierta debido a que cada zona y municipio tiene características diferentes para la producción ganadera, cosa que explica estos resultados, sin embargo es importante recalcar que se ocupan modelos robustos resistentes a la heterocedasticidad, o modelos que trabajan bien con las colas como es bagging, de tal modo que cómo se puede ver en r2 esto no afecta al modelo.

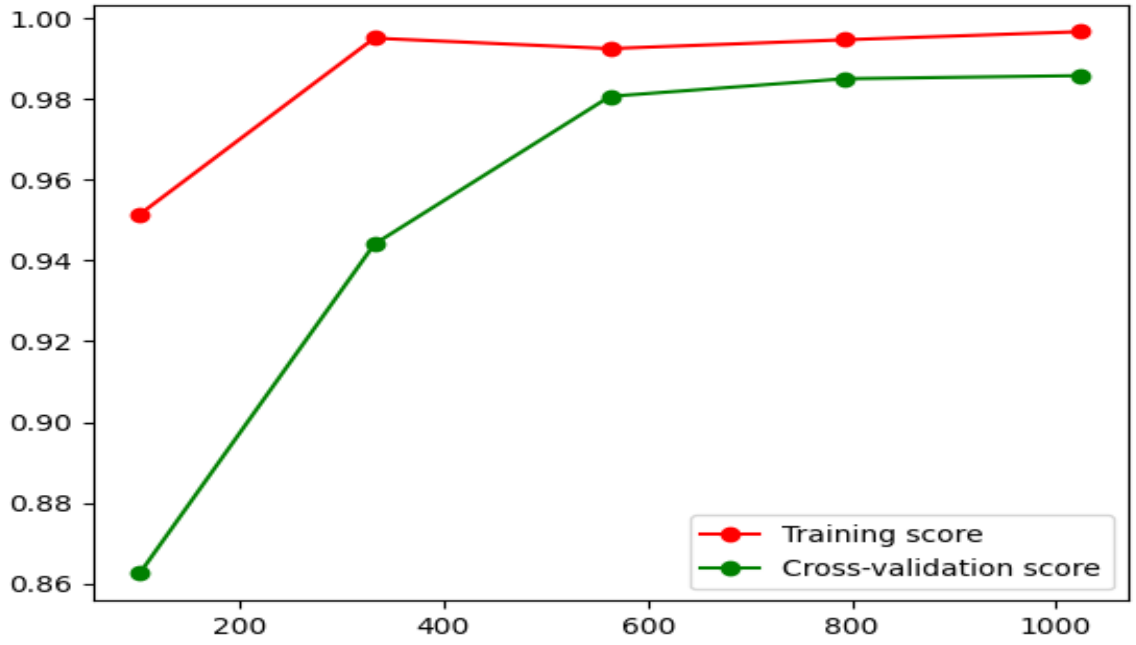
4.21 Evaluación del sobreajuste (Overfitting)

Ahora para tener en cuenta también otro de los problemas verificamos si existe overfitting en las predicciones o bien que el modelo no esté sobre ajustando los datos para esto primero se compara el MAE promedio y lowest mae obtenido por los hiperparámetros del modelo al ocupar búsqueda bayesiana, entonces para el lowest mae encontrado se observa que es mayor que el mae, para ser exactos nos da **Lowest MAE found: 82.56182417338664 y MAE: 35.51917832583811**, lo que nos lleva a suponer que este no existe sobre ajuste, sin embargo es importante tener en cuenta que el "Lowest MAE found" es el MAE en el conjunto de validación, mientras que el MAE es en el conjunto de prueba. (Montesinos López et al., 2022). Estos conjuntos pueden tener características diferentes, lo que podría explicar la discrepancia en los valores, para darnos certeza de que no existe sobre ajuste se grafica la curva de aprendizaje:

4.22 Gráfica de la curva de aprendizaje

Gráfica 47

Gráfica de la curva de aprendizaje del modelo



Interpretación:

- Puntuación de Entrenamiento (línea roja): Esta línea muestra cómo se desempeña el modelo en el conjunto de entrenamiento a medida que aumenta el tamaño del conjunto de entrenamiento.
- Puntuación de Validación Cruzada (línea verde): Esta línea muestra cómo se desempeña el modelo en el conjunto de validación a medida que aumenta el tamaño del conjunto de entrenamiento.
- Convergencia de las Líneas: Si las dos líneas convergen y la brecha entre ellas es pequeña, indica que el modelo tiene un buen equilibrio entre sesgo y varianza, y que no hay sobreajuste.
- Divergencia de las Líneas: Si las líneas no convergen o hay una gran brecha entre ellas, podría indicar sobreajuste. En este caso, el modelo estaría aprendiendo el "ruido" en los datos de entrenamiento y no generalizando bien a datos no vistos.

En este caso se observa que existe una pequeña brecha, pero no existe una convergencia de tal modo que puede haber existencia de sobre ajuste por lo que se procedería a imprimir el mae de los datos entrenados y el mae de los datos de prueba:
Cálculo del MAE en el Conjunto de Entrenamiento:

- Predicciones en el Conjunto de Entrenamiento: Se utilizan los datos de entrenamiento para hacer predicciones utilizando el pipeline previamente entrenado (modelo).
- Cálculo del MAE en el Conjunto de Entrenamiento: Se calcula el MAE comparando las predicciones con los valores verdaderos en el conjunto de entrenamiento.
- Resultado:
  Test MAE: 35.52917832583811

Cálculo del MAE en el Conjunto de Prueba:

- Predicciones en el Conjunto de Prueba: Se utilizan los datos de prueba para hacer predicciones utilizando el pipeline previamente entrenado.
- Cálculo del MAE en el Conjunto de Prueba: Se calcula el MAE comparando las predicciones con los valores verdaderos en el conjunto de prueba.
- Resultado:
  Train MAE: 36.95793384371643

Interpretación:

- Pequeña Diferencia entre MAE de Entrenamiento y Prueba: La diferencia entre el MAE en el conjunto de entrenamiento y el MAE en el conjunto de prueba es pequeña. Esto podría indicar que el modelo no está sobre ajustando los datos, ya que el rendimiento en los datos de entrenamiento y prueba es similar.
- Consistencia con la Observación Previa: Aunque la curva de aprendizaje anterior sugirió una posible existencia de sobreajuste debido a la falta de convergencia, esta

comparación de MAE proporciona evidencia adicional que podría refutar esa preocupación. La consistencia en el rendimiento entre los conjuntos de entrenamiento y prueba apoya la idea de que el modelo está generalizando bien a datos no vistos.

4.23 Prueba de Dickey-Fuller en los datos de entrada y salida

Teniendo en cuenta que se tienen algunas dudas de cómo se tiene manejado los datos en las predicciones se procede a observar cómo se tienen las salidas de los datos; por ejemplo, en primera instancia se hará una prueba de Dickey Fuller para comprobar si los datos de entrada y de salida son o no estacionarios dónde para los datos de entrada se obtiene:

Métrica	Valor
ADF Statistic	**-8.61167809**
p-value	**6.45E-14**
Critical Values (1%)	**-3.43556713**
Critical Values (5%)	**-2.8638439**
Critical Values (10%)	**-2.56799662**

Interpretación:

La estadística ADF es muy negativa y el p-value es extremadamente bajo, lo cual rechaza la hipótesis nula de que la serie tiene una raíz unitaria (no estacionaria). Esto indica que la serie de entrada es estacionaria.

Y para los datos de salida se obtiene:

Métrica	Valor
ADF Statistic	**-6.82448556**
p-value	**1.96E-09**
Critical Values (1%)	**-3.43350843**
Critical Values (5%)	**-2.86293525**
Critical Values (10%)	**-2.56751277**

Interpretación:

Similar a los datos de entrada, la estadística ADF en los datos de salida es negativa y el p-value es muy bajo. Esto también rechaza la hipótesis nula, lo cual indica que las predicciones también son estacionarias.

Conclusión

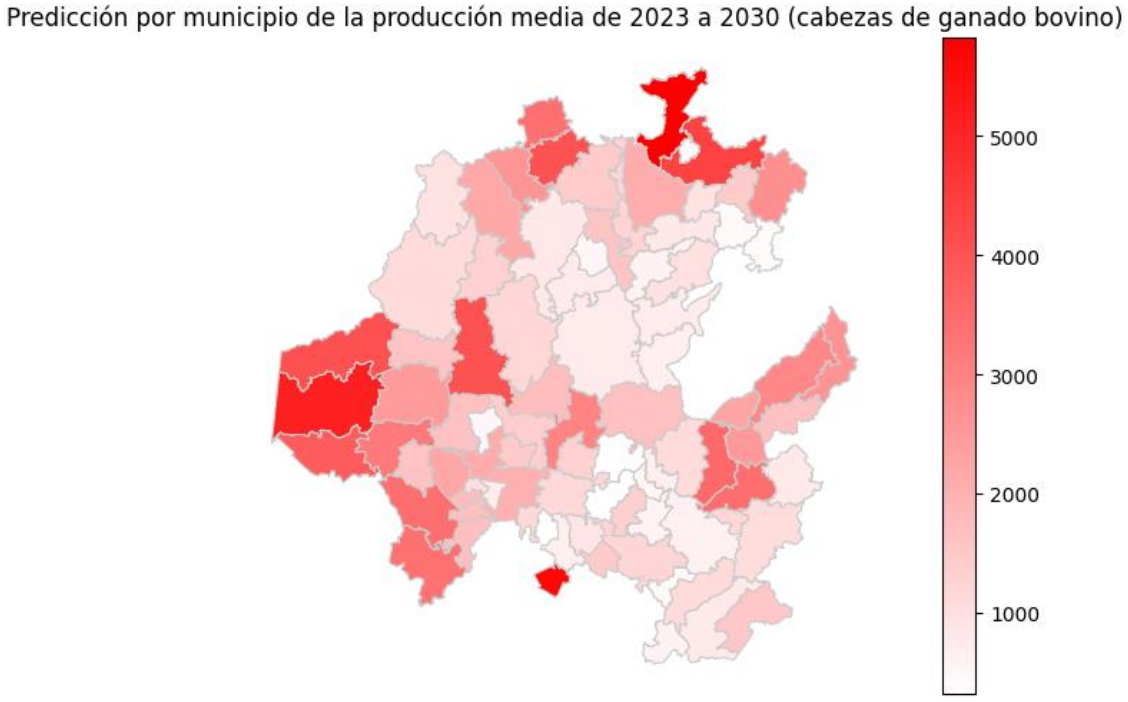
- Estacionariedad en Entrada y Salida: Tanto los datos de entrada como las predicciones son estacionarias. Esto significa que las propiedades estadísticas de estas series no cambian significativamente con el tiempo.
- Implicación en el Modelado: La estacionariedad en los datos es una propiedad deseable en muchos modelos de series temporales, ya que facilita la modelización y la interpretación de los resultados. En este caso, la estacionariedad tanto en los datos de entrada como en las predicciones podría ser un indicador positivo de la calidad del modelo.
- Consistencia con el Análisis Previo: Estos resultados están en línea con el análisis previo realizado en esta conversación, proporcionando una visión adicional sobre la naturaleza de los datos y el rendimiento del modelo.

4.24 Mapa coroplético de la producción media 2023 - 2030

Ahora se imprime el mapa coroplético de la media de producción de los datos pronosticados, que nos ayudará a ver cómo han cambiado los datos en torno a los datos antes pronosticados:

Figura 8

Mapa coroplético de la predicción por municipio de la producción media de 2023 a 2030



4.25 Análisis de los datos pronosticados versus los datos reales

Ahora se imprimirán los datos pronosticados, en este caso se le agregaron los datos de longitud y latitud para los análisis y mapas posteriores

Tabla 4

Tabla de la comparación de datos pronosticados y datos reales

	Municipio	Año	Valor_Predicho	Latitud	Longitud	Valor_Real
0	Acatlán	2006	6266.175445	20.145998	-98.438353	6837
1	Acatlán	2007	5198.509671	20.145998	-98.438353	5000
2	Acatlán	2008	4233.374454	20.145998	-98.438353	4020
3	Acatlán	2009	3788.374423	20.145998	-98.438353	3627
4	Acatlán	2010	3680.091953	20.145998	-98.438353	3472

Se hace el cálculo de las comparaciones dándonos como resultado:

Métrica	Valor
Cálculo del Error Absoluto Medio (MAE)	**287.57**
Cálculo del Error Cuadrado Medio (MSE)	**342727.11**
Cálculo del Error Porcentual Absoluto Medio (MAPE)	**21.02%**
Cálculo del Coeficiente de Determinación (R^2)	**0.7692**

Dónde:

- Cálculo del Error Absoluto Medio (MAE): Mide la diferencia promedio en magnitud entre los valores predichos y reales. Se obtiene un valor de 287.57.
- Cálculo del Error Cuadrado Medio (MSE): Mide la diferencia cuadrada promedio entre los valores predichos y reales. Se obtiene un valor de 342727.11.
- Cálculo del Error Porcentual Absoluto Medio (MAPE): Mide la diferencia porcentual promedio entre los valores predichos y reales. Se obtiene un valor de 21.02%.
- Cálculo del Coeficiente de Determinación (R^2): Mide la proporción de la varianza en la variable dependiente que es predecible a partir de las variables independientes. Se obtiene un valor de 0.7692.

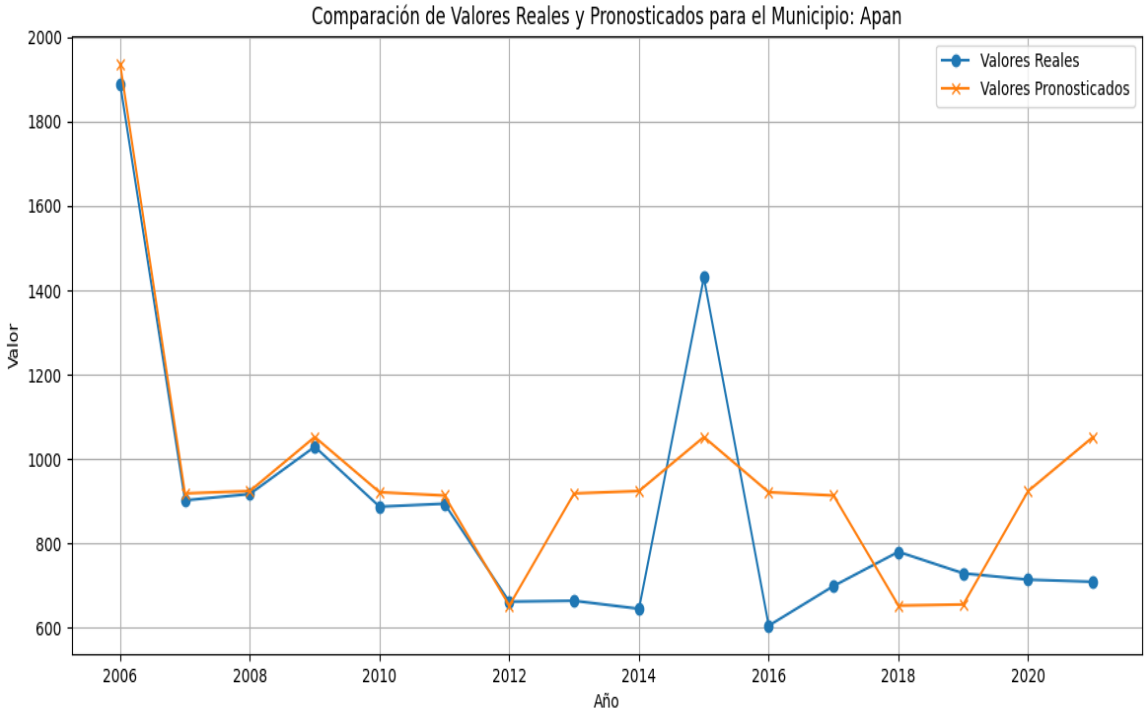
Interpretación:

- MAE y MSE: Los valores de MAE y MSE proporcionan una medida de la precisión de las predicciones en términos absolutos y cuadrados, respectivamente. Cuanto más bajos sean estos valores, mejor será la precisión del modelo.
- MAPE: Un MAPE del 21.02% significa que el modelo tiene un error promedio del 21.02% en sus predicciones. Esto puede considerarse aceptable o no, dependiendo del contexto y los requisitos del análisis.
- R^2: Un valor de R^2 de 0.7692 indica que aproximadamente el 76.92% de la variabilidad en los datos reales puede ser explicada por el modelo. Esto sugiere una buena calidad de ajuste, aunque podría haber margen para mejorar.

Para ver que tan bien está tomando las tendencias el modelo podemos ocupar esta comparación con este gráfico dónde se compara un municipio aleatorio con sus datos pronosticados y sus datos reales que en este caso es Apan:

Gráfica 48

Gráfica de la comparación de los valores reales y pronosticados para un municipio aleatorio, que en este caso es Apan

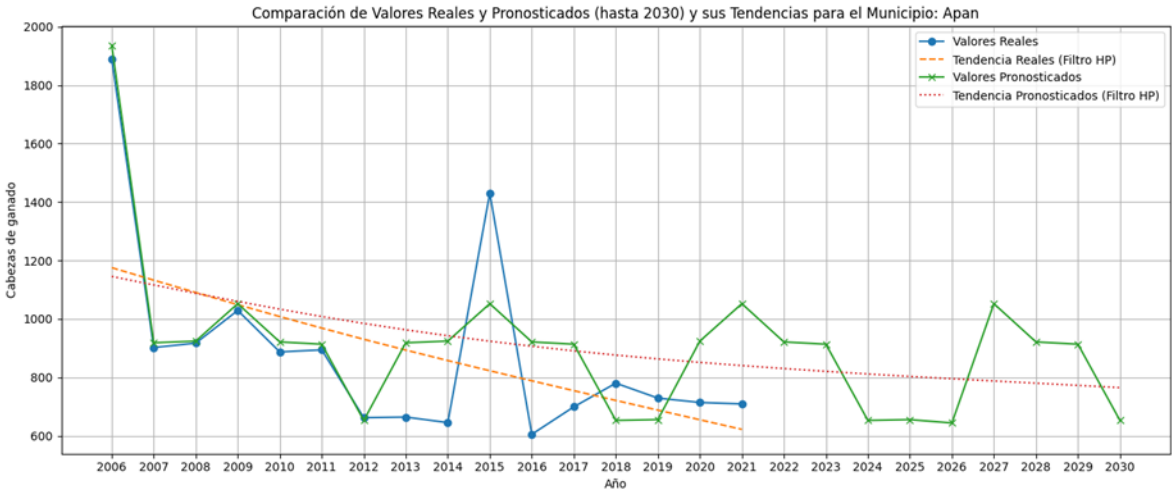


Dónde se observa perfectamente que el modelo está capturando las tendencias en los datos. Concluyendo que este modelo es bastante bueno para pronosticar los datos, ya que junto con el r2 de .7692 nos podemos dar cuenta que tiene una gran calidad de ajuste en el supuesto de que cualquier modelo que contenga en comparación entre sus valores reales y pronosticados mayor que .5.

El análisis de los datos pronosticados versus los datos reales muestra que el modelo actual es bastante bueno para pronosticar los datos y capta adecuadamente las tendencias en los datos. La evaluación cuantitativa y visual ofrece una comprensión completa del rendimiento del modelo y puede guiar futuros esfuerzos de mejora y ajuste.

Gráfica 49

Gráfica del análisis del municipio por medio de filtro Hodrick Prescott para la visualización de la tendencia



El filtro Hodrick-Prescott (HP) es una herramienta estadística utilizada para descomponer una serie temporal en dos componentes: una tendencia (o componente de ciclo de largo plazo) y un componente cíclico (o fluctuaciones de corto plazo alrededor de esta tendencia). (Hodrick & Prescott, 1981).

Comparación Visual en una Gráfica: Una vez que tenemos las tendencias suavizadas, las trazamos junto con los valores originales en un gráfico. Esto nos permite comparar visualmente:

Cómo evoluciona la tendencia real con respecto a la tendencia pronosticada.
Si el modelo de predicción está capturando correctamente la tendencia subyacente de los datos reales.

Evaluación de Diferencias: Al observar la diferencia (por ejemplo, la diferencia porcentual) entre la tendencia real y la tendencia pronosticada, podemos cuantificar qué tan bien se ajusta el modelo a los datos. Si las diferencias son pequeñas, esto sugiere que el modelo se está ajustando bien a la tendencia subyacente de los datos reales. Si las diferencias son grandes, indica que hay áreas de mejora en el modelo.

4.26 Análisis de posible exportación o importación de carne bovina para ubicación de rastro TIF

Ahora lo siguiente, con los datos obtenidos del pronóstico se prosigue a sacar una posible exportación o importación de carne bovino de los municipios, esto con la finalidad de generar una posible localización del rastro TIF ya que como antes habíamos mencionado todo este procedimiento se hizo con la finalidad de tener datos fiables pronosticados que nos ayuden a ver dónde se podría ubicar el rastro y si es una buena opción o no, por lo tanto se acomodarían los datos puesto que se tenga una aproximación a esas exportaciones e importaciones siguiendo la lógica de que este supuesto pueda darse de la producción de cabezas de bovino menos las cabezas de bovino consumidas.

Tabla 5

Tabla del análisis de posible exportación e importación de carne bovina

Municipio	Población de reses promedio 2023-2030	Población de personas promedio 2023-2030	Consumo de carne promedio en Kg. De 2023-2030	Cabezas de ganado consumidas en promedio 2023-2030	Relación entre oferta y demanda
Acatlán	3449.277925	22312.04730	341374.3236	1501.762480	1947.515445
Acaxochitlán	871.817114	48605.94865	743671.0143	3271.532594	-2399.715480
Actopan	3050.521069	65226.49865	997965.4293	4390.216059	-1339.694990
Agua Blanca de Iturbide	2180.272540	10279.01081	157268.8654	691.851920	1488.420620
Ajacuba	1929.933842	20282.09865	310316.1093	1365.132224	564.801618
...	...	...	...	...	...

Primero se imprime la tabla, recordando que para sacar las cabezas de consumo promedio se colocó primero el promedio de población por localidad; sacada por la tendencia promediada de 2023 a 2030 misma que ya fue explicada en los datos anteriores, para después ser multiplicada por el consumo último registrado de carne de res promedio por habitante en México; que es de “15.3kg” (esto según el compendio estadístico de carne 2022 de la página comecarne.org), así como el resultado ser dividido entre el promedio del

peso en canal de todos los municipios de los datos ya antes mencionados, en kilogramos; que es de “227,3157895Kg”.

$$\text{Cabezas de consumo promedio} = (\text{Promedio de población por localidad} \times 15.3)/227.3157895$$

Donde:

El "Promedio de población por localidad" es la tendencia promediada de 2023 a 2030.
El número 15.3 representa el consumo último registrado de carne de res promedio por habitante en México (en kg).
El número 227.3157895 representa el promedio del peso en canal de todos los municipios (en kg).

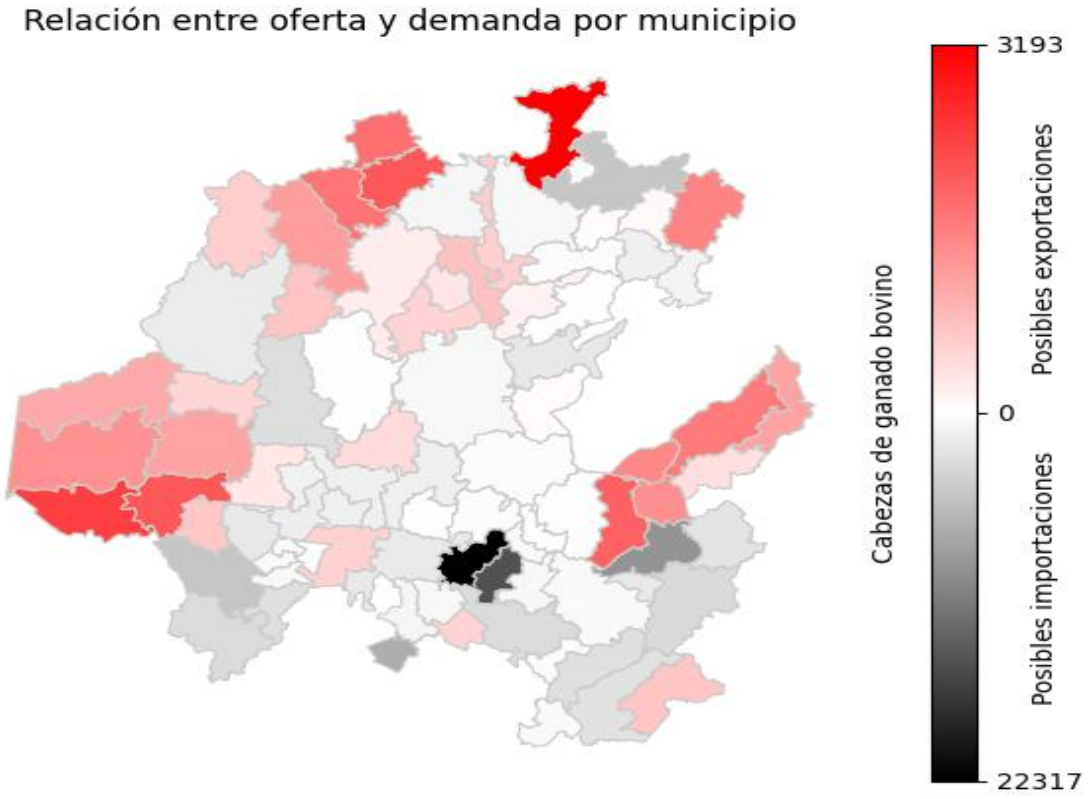
Y la definición de la Relación entre oferta y demanda; es la población de reses promedio de 2023 a 2030 menos las cabezas de ganado consumidas en promedio 2022 a 2030.

$$\text{Relación entre oferta y demanda} = \text{Población de reses promedio de } 2023 - 2030 - \text{Cabezas de ganado consumidas en promedio } 2023 - 2030$$

El análisis presentado proporciona una evaluación detallada de la producción y consumo de carne bovina en diferentes municipios. A través de la "Relación entre oferta y demanda," se puede identificar áreas donde puede haber un excedente o déficit en la producción de carne. Esta información es vital para tomar decisiones informadas sobre la ubicación de un rastro TIF, asegurando que se coloque en una región donde pueda servir de manera efectiva a las necesidades locales y contribuir al abastecimiento y procesamiento de carne de manera eficiente.

Figura 9

Mapa coroplético de la relación entre oferta y demanda por municipio



Nota: Las partes más obscuras representan una mayor importación mientras que los tonos más rojos representan mayor exportación bovina.

Ahora se imprime el mapa coroplético que nos ayudará a ver cómo están las exportaciones e importaciones de cada municipio dónde los colores más oscuros representan las posibles importaciones en cabezas de ganado y los rojos como posibles exportaciones en cabezas de ganado

4.27 Análisis de agrupamiento para identificar ubicaciones para centros de distribución y acopio

El análisis de agrupamiento, o clustering, es una técnica de aprendizaje no supervisada que se utiliza para identificar grupos homogéneos dentro de un conjunto de datos. Ahora se realiza un análisis de agrupamiento (clustering) para identificar posibles ubicaciones para los centros de distribución y acopio en diferentes municipios los cuales serían los

municipios que posiblemente exportan carne, dado del análisis anterior. Esto se hace con la intención de disminuir los costos relacionados con la entrega de carne por parte de los ganaderos y mejorar las prácticas en el manejo del ganado.

Contexto y Objetivo:

La finalidad de este análisis es determinar ubicaciones óptimas para centros de distribución y acopio, que servirán como puntos de recolección de ganado y centros de revisado preliminar para el rastro TIF. Esto permitirá una mejor gestión del ganado y una reducción de los costos, beneficiando a los ganaderos y contribuyendo a la calidad TIF para la exportación.

Tabla 6

Tabla de los pesos y ubicación de los municipios en el entorno geográfico

Municipio	Latitud	Longitud	Peso
Acatlán	20.14599753	-98.43835281	1947.515445
Agua Blanca de Iturbide	20.34998156	-98.35651984	1488.420620
Ajacuba	20.09268030	-99.12210399	564.8016176
Alfajayucan	20.41015828	-99.34946458	1126.671252
Almoloya	19.70312738	-98.40328030	715.7856113
Atlapexco	21.01745106	-98.34829642	70.18432667
Chapantongo	20.28559483	-99.41319277	2075.752896
Chapulhuacán	21.15787458	-98.90435814	2076.831214
Chilcuautla	20.33143910	-99.23167454	286.4689418
Eloxochitlán	20.74687461	-98.80930520	535.8706135
Huautla	21.03188243	-98.28668517	1531.505859
Huazalingo	20.98056421	-98.50774232	45.81143524
Huehuetla	20.46035677	-98.07859535	1136.262735
Huichapan	20.37553905	-99.65102930	1357.398389
Jacala de Ledezma	21.00849043	-99.18853709	1217.834983
Juárez Hidalgo	20.78338131	-98.82918225	317.6627911
La Misión	21.10140801	-99.12301200	1735.792089
Lolotla	20.84163503	-98.71711415	593.5610713
Metepec	20.23881199	-98.32208348	1375.535971
Molango de Escamilla	20.78658031	-98.73060350	769.5643214
Nicolás Flores	20.76804674	-99.15056418	743.7061007
Nopala de Villagrán	20.25195194	-99.64433781	2378.867366
Omitlán de Juárez	20.16956885	-98.64859790	11.74892797
Pacula	21.05029930	-99.29636159	615.0936316
Pisaflores	21.19493683	-99.00590219	1776.225709
San Agustín Metzquititlán	20.53306527	-98.63880134	25.68872557
San Bartolo Tutotepec	20.39915860	-98.20205941	1635.505363
San Felipe Orizatlán	21.17106016	-98.60758905	3193.503324
Santiago de Anaya	20.38273787	-98.96390502	428.2507170
Tasquillo	20.55204235	-99.31248643	506.6549098
Tecozautla	20.53415978	-99.63494250	1068.321568
Tenango de Doria	20.33845235	-98.22670080	393.5998613
Tepetitlán	20.18700312	-99.38024786	693.9012230
Tlahuiltepa	20.92389198	-98.95017064	236.2399636
Villa de Tezontepec	19.87998600	-98.81930736	526.1847575
Xochicoatlán	20.77704277	-98.67966839	164.6085326

Procesamiento y Análisis:

1. Conversión Geoespacial: Las coordenadas geográficas (latitud y longitud) se convierten a coordenadas cartesianas (x, y, z) para facilitar el análisis espacial.
2. Preprocesamiento: Las coordenadas cartesianas se escalan usando StandardScaler para normalizar los datos.
3. Determinación del Número Óptimo de Clústeres: Se emplean los métodos de Elbow y Silhouette para determinar el número óptimo de clústeres. Se grafican los resultados para visualizar el punto de codo (Elbow) y el puntaje de Silhouette. (Peter J. Rousseeuw, 1987).(Ketchen D. J. & SHOOK C. L.,1996).
4. Clustering Jerárquico: Se aplica el clustering jerárquico (Agglomerative Clustering) y se muestra el dendrograma correspondiente. Se selecciona un número óptimo de clústeres basado en el análisis visual del dendrograma. (Nielsen, 2016)
5. Aplicación de KMeans: Se aplica el algoritmo KMeans con el número óptimo de clústeres, considerando también los pesos de cada municipio. Los clústeres y sus centroides se visualizan en una gráfica 3D. (Abirami K. & Mayilvahanan P., 2016). (Di J. & Gou X., 2018).
6. Visualización en el Mapa: Se convierten los centroides de los clústeres a coordenadas geográficas y se visualizan en un mapa interactivo utilizando Folium. Los municipios se representan como círculos, mientras que los centroides de los clústeres se muestran como marcadores rojos.

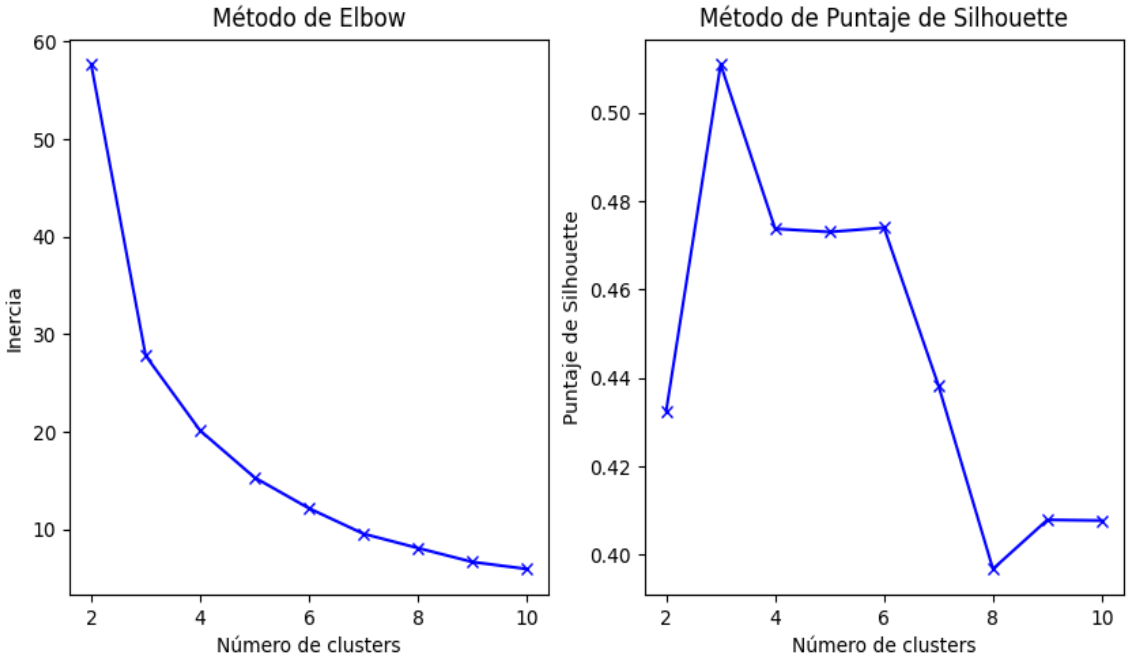
Resultados:

- Método de Elbow y Silhouette: Las gráficas de Elbow y Silhouette ayudan a determinar el número óptimo de clústeres. En este caso, se ha seleccionado 14 como el número de clústeres, ya que es el nivel de visualización que me interesa debido a la complejidad que existe en cuanto normas y estándares en Hidalgo.
- Dendrograma: El dendrograma proporciona una visión jerárquica de cómo se agrupan los municipios, y se anotan los clústeres seleccionados.

- Gráfica 3D: La visualización 3D de los clústeres y sus centroides permite una interpretación espacial de cómo se agrupan los municipios según su ubicación y peso.
- Mapa Interactivo: El mapa muestra la distribución geográfica de los municipios y las ubicaciones propuestas para los centros de distribución y acopio.

Gráfica 50

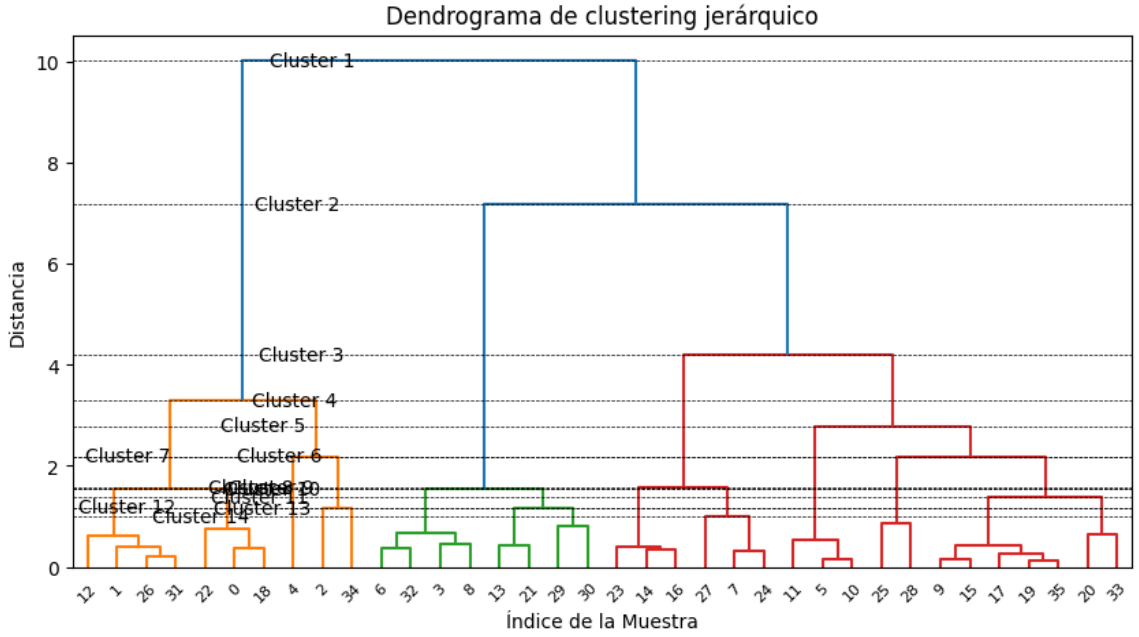
Gráficas del método de Elbow y del método de Silhouette para ubicación de los centros



Nota: En el método de elbow, la parte dónde existe un mayor descenso es dónde menciona el mejor resultado mientras que en el de silhouette la parte mayor lo muestra.

Figura 10

Dendrograma de clustering jerárquico para la ubicación de los centros



Nota: Entre más se tenga una visión mayor, en este caso representado con lineas azules muestra el resultado óptimo.

Figura 11

Mapa en 3d para los resultados del algoritmo de clustering para la ubicación de centros

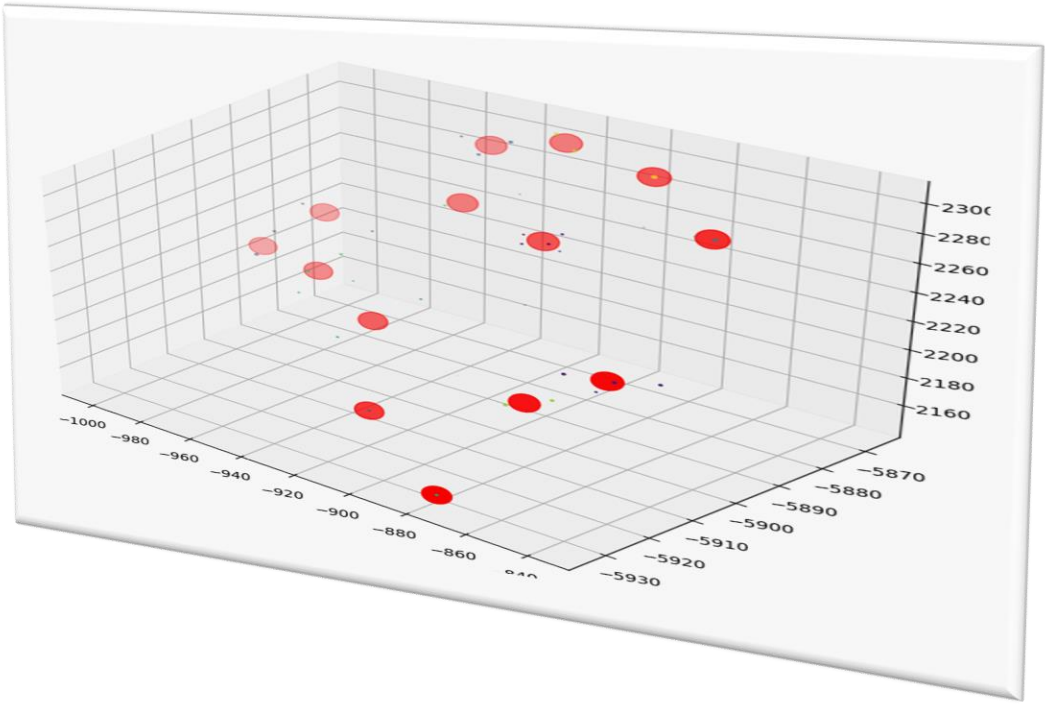
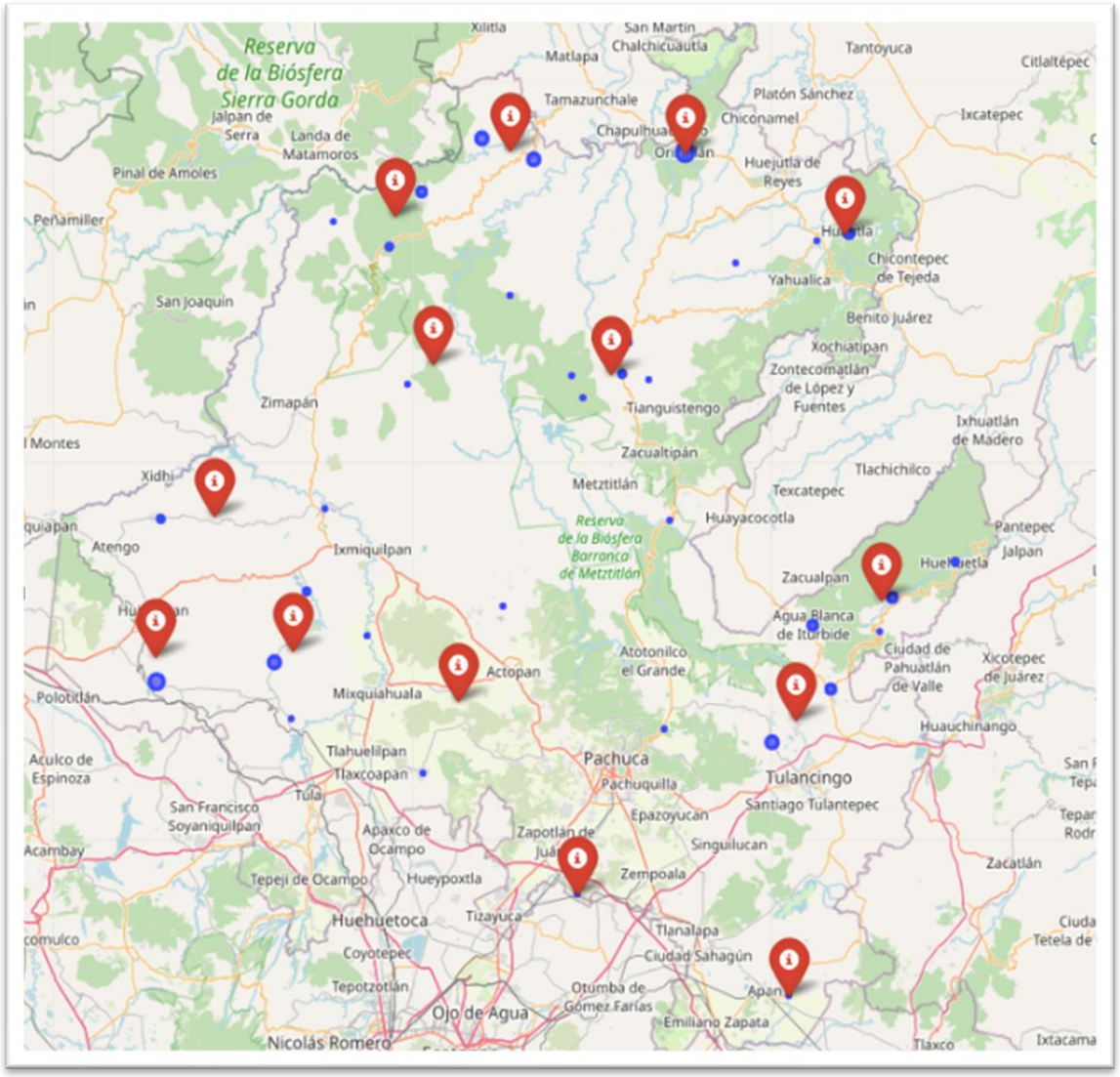


Figura 12

Mapa de resultados del algoritmo de clustering para los centros



Nota: Los círculos azules muestran mayores concentraciones de producción en diferentes municipios mientras que las figuras rojas muestran la selección de clústeres, o para este caso los diferentes centros.

Cómo se puede observar, una visión por parte de los diferentes métodos para ver qué cantidad de clústeres son los indicados, sería de 3 para este caso, sin embargo, es importante aclarar que se necesita una visualización mayor para poder ver la complejidad en cuanto distancias y problemas con los ganaderos se refiere, de tal modo que los ocupados dado el dendrograma para la visualización que nos interesa son de 14 clústeres mismos que más abajo se mencionan en la tabla.

4.28 Centro de gravedad para identificar ubicaciones para centros de distribución y acopio

Para proseguir también se analiza el centro de gravedad, una herramienta de la ingeniería que nos ayuda a determinar el centro de todos estos datos con relación a su peso y su ubicación geográfica

Contexto y Objetivo:

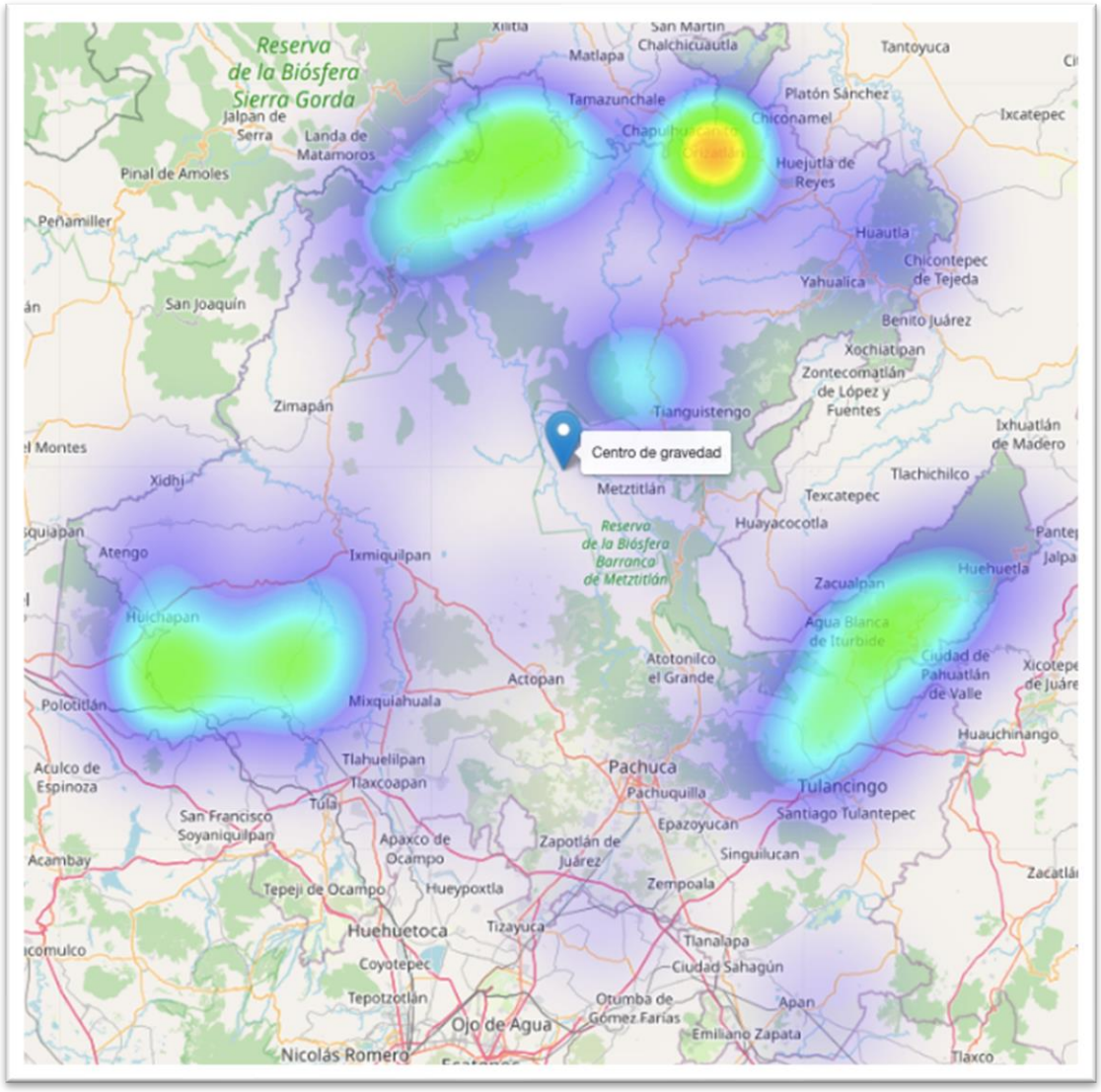
En el ámbito de la logística y la planificación, el centro de gravedad es una herramienta valiosa para determinar una ubicación óptima para, por ejemplo, un centro de distribución. En este caso, el centro de gravedad se calcula en función de la ubicación geográfica de los municipios y un peso asociado, lo cual puede representar una medida relevante como la producción de carne.

Procesamiento y Análisis:

- Definición de Datos: Los datos de los municipios, incluyendo su latitud, longitud, y peso, se definen en un DataFrame de Pandas.
- Función Objetivo: Se define una función objetivo que calcula la suma ponderada de las distancias al cuadrado entre una ubicación dada y todas las ubicaciones de los municipios. La ponderación se realiza utilizando el peso asociado a cada municipio.
- Optimización: Se utiliza el algoritmo de Nelder-Mead para minimizar la función objetivo. Esto determina las coordenadas óptimas del centro de gravedad. (Nelder & Mead, 1965)
- Visualización en Mapa: Se crea un mapa utilizando Folium, con una capa de mapa de calor que representa los pesos de los municipios y un marcador que indica la ubicación del centro de gravedad.

Figura 13

Mapa de calor de los resultados de la técnica de centro de gravedad para los centros



Nota: Las verdes azules muestran mayores concentraciones de producción en diferentes municipios mientras que las figuras azules muestran la selección de la técnica.

Resultados:

- Coordenadas del Centro de Gravedad: Las coordenadas óptimas resultantes representan el centro de gravedad de los municipios, teniendo en cuenta su ubicación y peso.

- Mapa Interactivo: El mapa interactivo muestra la distribución geográfica de los municipios y la ubicación central calculada. La capa de mapa de calor representa la importancia relativa (peso) de cada municipio, mientras que el marcador señala la ubicación del centro de gravedad.

Ahora teniendo todas las posibles ubicaciones de los posibles centros de distribución y acopio los cuales serían estos:

Tabla 7

Tabla de los resultados de las técnicas ocupadas para determinar los centros

Municipio	Latitud	Longitud
Chapantongo	20.28559483	-99.4131928
Chapulhuacán	21.15787458	-98.9043581
San Bartolo Tutotepec	20.3991586	-98.2020594
Jacala de Ledezma	21.00849043	-99.1885371
Huejutla de Reyes	21.13959058	-98.4204449
Molango de Escamilla	20.78658031	-98.7306035
Alfajayucan	20.41015828	-99.3494646
Huichapan	20.37553905	-99.6510293
Villa de Tezontepec	19.879986	-98.8193074
Francisco I. Madero	20.24540323	-99.0888141
Acatlán	20.14599753	-98.4383528
Nicolás Flores	20.76804674	-99.1505642
San Felipe Orizatlán	21.17106016	-98.6075891
Tecozautla	20.53415978	-99.6349425
Metztitlán	20.5948676	-98.7642084

Se procede a determinar los municipios aptos con relación a los clientes, teniendo en cuenta que la producción de cada uno de los municipios tiene una producción de 6602.24979 ya que sería el resultado de dividir entre los 15 municipios propuestos al total de la demanda de los posibles clientes de los municipios, que son los 30 municipios con el IDH más alto en Hidalgo en este caso que importan carne, ya que se va por el supuesto de que estos pueden pagar la calidad de la carne TIF, esta información obtenida de un pdf del gobierno del estado de Hidalgo del 2015, (última fecha de actualización), del IDH por municipio.

Aquí muestro la tabla correspondiente con los datos de los posibles clientes, con sus respectivas latitudes y longitudes y demandas:

Tabla 8

Tabla de los posibles clientes basados en el IDH

Municipio	Latitud	Longitud	Demanda
Actopan	20.2691874	-98.9428621	1339.69499
Apan	19.7100318	-98.4523672	2471.98958
Atitalaquia	20.0592979	-99.2211295	557.11692
Atotonilco de Tula	20.0003921	-99.2180777	2713.44028
Emiliano Zapata	19.6573693	-98.5473023	449.384075
Epazoyucan	20.0181109	-98.6360483	602.108129
Francisco I. Madero	20.2454032	-99.0888141	1237.10266
Huejutla de Reyes	21.1395906	-98.4204449	4979.11386
Ixmiquilpan	20.4874612	-99.2158499	2838.44661
Mineral de La Reforma	20.0724348	-98.6958849	15063.6075
Mineral del Monte	20.140557	-98.6714745	408.954391
Mixquiahuala de Juárez	20.2297154	-99.2139797	1416.81818
Pachuca de Soto	20.1269997	-98.7300549	22317.9468
Progreso de Obregón	20.2486801	-99.1894236	1158.82812
San Agustín Tlaxiaca	20.1157998	-98.886755	1756.70618
San Salvador	20.284959	-99.0154511	1244.23329
Santiago Tulantepec de Lugo Guerrero	20.0405075	-98.3573677	1837.63126
Tepeapulco	19.7858894	-98.5530995	2769.98557
Tepeji del Río de Ocampo	19.9054893	-99.3418244	2968.83737
Tezontepec de Aldama	20.1928939	-99.2727168	1921.07648
Tizayuca	19.8412622	-98.981517	7102.83277
Tlahuelilpan	20.1308266	-99.2345684	389.809182
Tlanalapa	19.8178601	-98.6038007	278.689817
Tlaxcoapan	20.0915563	-99.2213648	353.59823
Tolcayuca	19.9567383	-98.9216934	847.097281
Tula de Allende	20.0714581	-99.3448035	4899.4767
Tulancingo de Bravo	20.1075495	-98.3819646	9277.88531
Zacualtipán de Ángeles	20.6454643	-98.6535594	2064.96954
Zapotlán de Juárez	19.9741479	-98.8619187	695.70763
Zempoala	19.9156242	-98.6681116	3070.65811

Nota: La demanda está dada en cabezas de ganado bovino.

4.29 Algoritmo Evolutivo NSGA 2 para localizar las mejores localizaciones

Para la primera instancia se procederá a ocupar un algoritmo evolutivo para la selección de los mejores municipios con respecto a la distancia y el costo, esto debido a que las distancias haversine y los cálculos hechos son difíciles de trabajar con programación lineal, y para la disminución de dos objetivos, como lo es el costo de la distancia y el costo de la demanda, por lo que un algoritmo evolutivo NSGA 2 es uno que ayuda mucho en esta tarea. (Deb K. et al., 2000).

Código: Ver ANEXO B: CÓDIGO 2.

Explicación del código:

1. Importación de Módulos y Datos Iniciales

El código comienza importando las librerías necesarias para ejecutar el algoritmo genético y trabajar con datos geoespaciales. Luego, se definen dos listas, instalaciones y clientes, que contienen información sobre las ubicaciones y capacidades de las instalaciones y las ubicaciones y demandas de los clientes, respectivamente.

2. Constantes

Se define COST_MULTIPLIER, una constante utilizada para calcular el costo de exceder la capacidad de una instalación. Calcula el costo total basado en cuánto se excede la capacidad de una instalación. Si la demanda total asignada a una instalación excede su capacidad, entonces hay un costo asociado. El costo es proporcional al exceso de demanda y se multiplica por COST_MULTIPLIER.

3. Configuración del Algoritmo Genético

Se configura el algoritmo genético utilizando la biblioteca deap. Se define un tipo de "aptitud" multiobjetivo (minimizar la distancia y el costo) y un "individuo" que es una lista de asignaciones de clientes a instalaciones.

4. Funciones Auxiliares

Se definen varias funciones:

haversine(): Calcula la distancia entre dos puntos geográficos.
calcular_aptitud(): Calcula la aptitud de un individuo basado en la distancia total y el costo total.
calcular_distancia_demanda_costo(): Calcula la distancia total, la demanda total y el costo total basado en una solución dada.
proximidad_geo(): Usa el algoritmo KMeans para agrupar geográficamente los clientes a las instalaciones.
crear_individuo(): Crea un individuo basado en la agrupación geográfica.
custom_mutate(): Define cómo se mutará un individuo.
custom_mate(): Define cómo se cruzarán dos individuos.

5. Configuración del Toolbox del Algoritmo Genético

Aquí se registran las funciones definidas anteriormente en un "toolbox" que luego se utilizará para ejecutar el algoritmo genético.

6. Evolución Genética

Se define la función perform_evolution(), que realiza la evolución genética para encontrar la solución óptima. Utiliza selección basada en NSGA-II, cruce y mutación para evolucionar la población a lo largo de varias generaciones.

7. Función Principal

La función main() inicializa una población, la evalúa y luego realiza la evolución genética. Luego, muestra resultados y gráficos de la evolución de la aptitud a lo largo de las generaciones.

8. Visualización

La función visualizar_resultados() crea un mapa utilizando folium para visualizar la solución óptima. Muestra las instalaciones, los clientes y las rutas entre ellos.
Algoritmo Genético:

Un algoritmo genético es un método de optimización y búsqueda inspirado en el proceso de selección natural. Los individuos de la población son soluciones potenciales al problema, y se seleccionan para reproducirse y generar descendencia basándose en su aptitud. Los operadores genéticos incluyen selección, cruce (recombinación) y mutación. (Lange et al., 2023)

En el código, cada individuo representa una asignación de clientes a instalaciones. La aptitud se basa en la distancia total entre clientes e instalaciones y en un costo asociado con exceder la capacidad de producción de una instalación.

KMeans:

Antes de iniciar el algoritmo genético, se utiliza el algoritmo de agrupación KMeans para proporcionar una solución inicial basada en la proximidad geográfica. KMeans intenta particionar un conjunto de puntos en K grupos (clústeres), de modo que la suma de las distancias al cuadrado de los puntos al centroide del clúster sea mínima. (Abirami K. & Mayilvahanan P., 2016). (Di J. & Gou X., 2018).

NSGA-II (Algoritmo Genético de Clasificación No Dominada II)

El NSGA-II es un algoritmo para optimización multiobjetivo, que busca soluciones que optimicen varios objetivos al mismo tiempo. Dada la naturaleza conflictiva de algunos objetivos (por ejemplo, minimizar el costo al tiempo que maximiza la eficiencia), no siempre hay una única solución óptima. En lugar de eso, se tiene un conjunto de soluciones óptimas, conocido como el frente de Pareto. (Deb K et al., 2000).

Características principales de NSGA-II:

Selección basada en dominancia:

Una solución A domina a una solución B si A es al menos igual de buena que B en todos los objetivos y estrictamente mejor en al menos uno.

NSGA-II clasifica las soluciones en diferentes frentes. El frente 1 (F1) es el conjunto de soluciones no dominadas. Una vez que se identifica F1, se elimina del conjunto y se busca el siguiente conjunto de soluciones no dominadas (F2), y así sucesivamente.
Diversidad en el frente de Pareto:

Además de la dominancia, NSGA-II introduce una métrica llamada "distancia de aglomeración" para mantener la diversidad en el frente de Pareto. Esta métrica mide cuán cerca están las soluciones entre sí en el frente.

Las soluciones que están rodeadas por otras soluciones cercanas se consideran menos importantes que las que están solas, ya que representan regiones menos exploradas del espacio de soluciones.

Operadores genéticos:

NSGA-II utiliza operadores de selección, cruce y mutación similares a otros algoritmos genéticos. Sin embargo, la selección se basa en la clasificación de dominancia y la distancia de aglomeración.
Elitismo:

NSGA-II es un algoritmo elitista, lo que significa que garantiza que las soluciones de un frente de Pareto no se pierdan en las generaciones subsiguientes.

En el código proporcionado, NSGA-II es utilizado con la función tools.selNSGA2. Los objetivos que se están optimizando son la distancia total entre clientes e instalaciones y el costo total asociado con exceder la capacidad de producción de una instalación.

Aplicación en el Código:

En el contexto del problema, el uso de NSGA-II permite encontrar soluciones que equilibren la distancia y el costo. Por ejemplo, una solución podría tener una distancia total mínima pero un costo elevado, mientras que otra podría tener un costo más bajo pero una distancia mayor. Al utilizar NSGA-II, se pueden identificar múltiples soluciones óptimas que representan diferentes equilibrios entre estos dos objetivos.

Lógica Matemática del Código de Algoritmos Genéticos para Asignación de Instalaciones a Clientes

1. Datos:

- (I) representa el conjunto de instalaciones y (C) el conjunto de clientes.
- Cada instalación (i) en (I) y cada cliente (j) en (C) tienen coordenadas geográficas dadas por (lat_i, lon_i) y (lat_j, lon_j), respectivamente.
- Cada instalación (i) tiene una producción máxima (P_i).
- Cada cliente (j) tiene una demanda (D_j).

2. Variables:

- Un individuo se representa como una lista, donde el índice corresponde a un cliente y el valor en ese índice representa la instalación asignada a ese cliente.

3. Funciones:

- Haversine:

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\Delta\text{lon}}{2}\right)$$

$$c = 2 \cdot \arctan 2\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R \cdot c$$

Donde:

- $(\Delta\text{lat} = \text{lat}_2 - \text{lat}_1)$
- $(\Delta\text{lon} = \text{lon}_2 - \text{lon}_1)$
- (R) es el radio de la Tierra, que es 6371 km.

Por lo tanto:

- La distancia entre la instalación (i) y el cliente (j) es:

Dados (lat_1, lon_1) y (lat_2, lon_2):

$$\mathrm{d}_{ij} = R{\cdot}2{\cdot}\arctan 2\left(\sqrt{\sin^2\left(\frac{lat_j - lat_i}{2}\right) + \cos(lat_i) \cdot \cos(lat_j) \cdot \sin^2\left(\frac{lon_j - lon_i}{2}\right)}, \sqrt{1 - \sin^2\left(\frac{lat_j - lat_i}{2}\right) - \cos(lat_i) \cdot \cos(lat_j) \cdot \sin^2\left(\frac{lon_j - lon_i}{2}\right)}\right)$$

- Proximidad Geográfica (KMeans): Proporciona una solución inicial basada en la agrupación geográfica.

4. Objetivos:

1. Minimizar la distancia total de asignación:

$$\min \sum_{j \in C} d_{\text{instalación asignada a } j, j}$$

2. Minimizar el costo total por exceso de demanda:

$$\min \sum_{i \in I} \max\left(0, \sum_{j \in C} \text{si cliente } j \text{ es atendido por } i \times D_j - P_i\right) \times \text{COST\_MULTIPLIER}$$

5. Restricciones:

- No hay necesidad de una restricción explícita que garantice que cada cliente esté asignado a una instalación porque esta asignación está implícita en la representación de las soluciones.

6. Operadores Genéticos:

- Mutación (`custom_mutate`): Cambia aleatoriamente la asignación de un cliente a una instalación con una probabilidad (indpb) o diciendo que el operador de mutación introduce variabilidad de una manera bastante aleatoria, reasignando clientes a instalaciones al azar basándose en una probabilidad (indpb). Esto puede mejorarse considerando un enfoque basado en alguna heurística o en la proximidad geográfica, sin embargo dada las condiciones de posibilidad de clientes e instalaciones, es más factible dejar una aleatoriedad, si se quisiera mejorar dicho enfoque podría verse beneficiado hacer el estudio real de clientes potenciales y construcción real de centros.

- Cruce (`custom_mate`): Intercambia asignaciones entre dos individuos si las instalaciones asignadas están geográficamente próximas y ambas instalaciones pueden satisfacer la demanda del cliente, asegurando que los intercambios sean viables.

7. Parámetros del Algoritmo:

- Tamaño de la población: `tamanio_poblacion`.
- Probabilidad de cruce: `probabilidad_cruce`.
- Probabilidad de mutación: `probabilidad_mutacion`.
- Número de generaciones: `generaciones`.

8. Proceso del Algoritmo:

- Inicializa una población basada en la proximidad geográfica.
- Evalúa cada individuo en la población.
- Realiza selección, cruce y mutación para generar la siguiente generación.
- Repite el proceso por un número específico de generaciones.

9. Visualización:

- Una vez que el algoritmo ha terminado, se puede visualizar el mejor individuo en un mapa, mostrando las asignaciones de instalaciones a clientes.

Resultado:

Mejor Individuo (Asignaciones de Clientes a Instalaciones)

[9, 8, 9, 8, 8, 10, 9, 12, 13, 2, 9, 6, 9, 0, 9, 9, 8, 8, 8, 0, 10, 7, 8, 0, 8, 7, 8, 14, 8, 8]

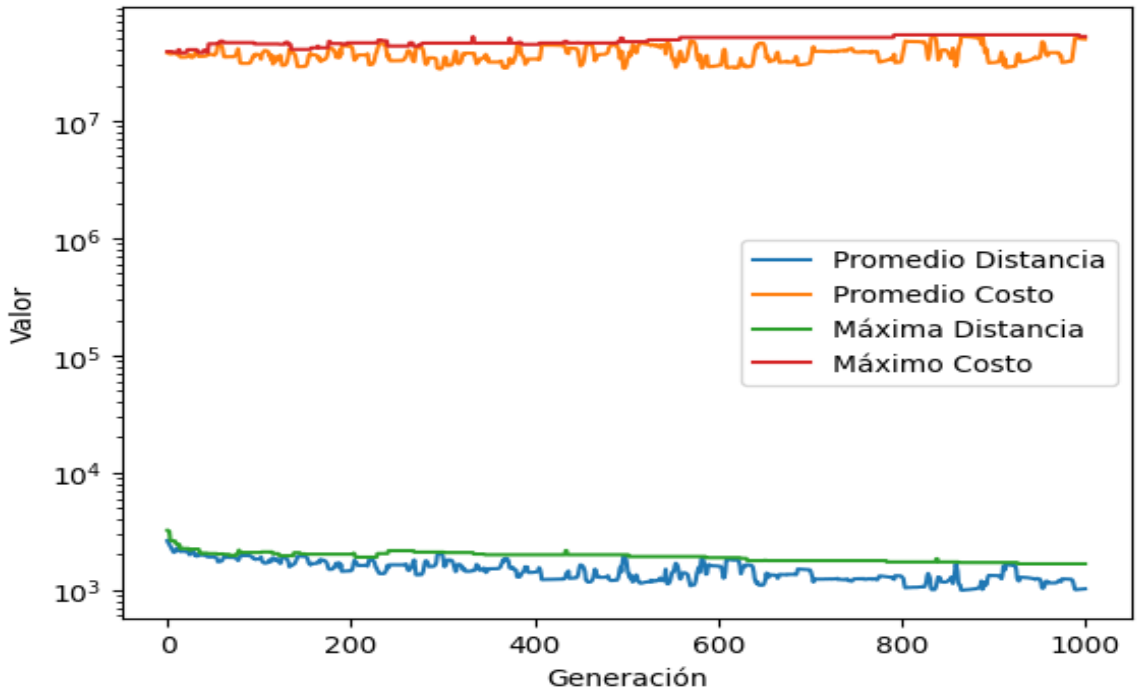
Distancia Total	**Costo Total**
952.3291478267972 km	56276505.4

Mejor Solución (Asignaciones)

	Cliente	**Instalación**	**Atendido por**
0	Actopan	9	Francisco I. Madero
1	Apan	8	Villa de Tezontepec
2	Atitalaquia	9	Francisco I. Madero
3	Atotonilco de Tula	8	Villa de Tezontepec
4	Emiliano Zapata	8	Villa de Tezontepec
5	Epazoyucan	10	Acatlán
6	Francisco I. Madero	9	Francisco I. Madero
7	Huejutla de Reyes	12	San Felipe Orizatlán
8	Ixmiquilpan	13	Tecozautla
9	Mineral de La Reforma	2	San Bartolo Tutotepec
10	Mineral del Monte	9	Francisco I. Madero
11	Mixquiahuala de Juárez	6	Alfajayucan
12	Pachuca de Soto	9	Francisco I. Madero
13	Progreso de Obregón	0	Chapantongo
14	San Agustín Tlaxiaca	9	Francisco I. Madero
15	San Salvador	9	Francisco I. Madero
16	Santiago Tulantepec de Lugo Guerrero	8	Villa de Tezontepec
17	Tepeapulco	8	Villa de Tezontepec
18	Tepeji del Río de Ocampo	8	Villa de Tezontepec
19	Tezontepec de Aldama	0	Chapantongo
20	Tizayuca	10	Acatlán
21	Tlahuelilpan	7	Huichapan
22	Tlanalapa	8	Villa de Tezontepec
23	Tlaxcoapan	0	Chapantongo
24	Tolcayuca	8	Villa de Tezontepec
25	Tula de Allende	7	Huichapan
26	Tulancingo de Bravo	8	Villa de Tezontepec
27	Zacualtipán de Ángeles	14	Metztitlán
28	Zapotlán de Juárez	8	Villa de Tezontepec
29	Zempoala	8	Villa de Tezontepec

Gráfica 51

Grafica de resultados del algoritmo genético para determinar los centros



Dónde la interpretación de los valores de la aptitud o el “fitness” y cómo evolucionan a lo largo del tiempo dependen del total ponderado entre la distancia y la asignación de la producción. En un problema de optimización multiobjetivo como este, se intenta minimizar (o maximizar) varias medidas al mismo tiempo, ya que se ocupa un algoritmo NSGA-2, lo que puede llevar a trade-offs (una situación en la que, para obtener una ganancia en un aspecto, se debe aceptar una pérdida en entre las diferentes medidas. (Deb K et al., 2000).

En este caso, la distancia está disminuyendo y el costo está aumentando, eso es el resultado de un compromiso entre minimizar la distancia y minimizar el costo debido al uso del algoritmo NSGA. Por ejemplo, las soluciones que tienen menores distancias implican asignar más clientes a instalaciones más cercanas, lo que a su vez lleva a exceder la capacidad de esas instalaciones y, por tanto, aumentar el costo.

En este caso minimizar la distancia es más importante que minimizar el costo, entonces podría ser aceptable que el costo aumente si eso permite reducir la distancia, debido a que las reses en promedio tienen una calidad mayor al ser sometidas a un estrés menor o un menor viaje. (Koza J. R., 1994).

- La distancia total representa la suma de las distancias desde cada centro de distribución o acopio hasta sus respectivos clientes asignados, calculada utilizando la fórmula de Haversine. La minimización de esta distancia es fundamental para reducir los costos de transporte y garantizar una entrega rápida.
- El costo total se calcula como la suma de las penalizaciones por exceder la capacidad de producción de los centros de distribución o acopio. Cada vez que la demanda asignada a un centro excede su capacidad de producción, se incurre en un costo adicional.

La solución óptima encontrada en este caso asigna a cada cliente a un centro de distribución o acopio específico, de manera que se minimice tanto la distancia total como el costo total, sujeto a las restricciones de capacidad.
La representación gráfica muestra cómo la aptitud evoluciona a lo largo de las generaciones en la optimización. En el gráfico, podemos observar que:

- Promedio Distancia: Representa la distancia promedio en la población en cada generación. A medida que las generaciones avanzan, este valor tiende a disminuir, lo que indica una mejor asignación de clientes a centros de distribución y acopio.
- Promedio Costo: Representa el costo promedio en la población en cada generación. Su evolución puede variar dependiendo de los trade-offs entre distancia y costo.
- Máxima Distancia: Representa la distancia máxima en la población en cada generación.
- Máximo Costo: Representa el costo máximo en la población en cada generación.

El resultado final ofrece una asignación detallada de cada cliente a una instalación específica, con una distancia total de 952.3291478267972 y un costo total de 56276505.3995 (cabezas de ganado).

Esta solución representa un equilibrio eficiente en términos de Pareto entre la minimización de la distancia y el costo. (Moc W. B. T., 2011).La elección de esta solución óptima depende de las prioridades y restricciones del problema real, y puede requerir una consideración cuidadosa de los aspectos económicos, logísticos y operativos.

Es importante recordar que el algoritmo NSGA-II, que se utiliza en este código, no busca una única solución óptima, sino un conjunto de soluciones que representan diferentes trade-offs entre los objetivos. Estas soluciones se conocen como el frente de Pareto. Cada solución en el frente de Pareto es óptima en el sentido de que no se puede mejorar un objetivo sin empeorar al menos uno de los otros. Por lo tanto, la elección de la solución final a utilizar dependerá de cuánto valor se asigne a cada uno de los objetivos en el problema específico que se está resolviendo. (Mock W. B. T., 2011)

En este caso, por lo tanto, el valor de "952.3291478267972" es la mejor (menor) distancia total encontrada por el algoritmo después de todas las generaciones de evolución.

El costo se calcula con la función calcular_costo_total. Esta función calcula la cantidad en que la demanda total de todas las asignaciones excede la capacidad de la instalación y multiplica ese exceso por 1000. Si la demanda no excede la capacidad en ninguna de las instalaciones, el costo es 0.
Dicho de otra manera, el costo es una penalización que se aplica cuando la demanda total de los clientes asignados a una instalación excede la capacidad de producción de esa instalación.(Cicirello & Smith, 2000).

El costo en este contexto no se mide en unidades monetarias, sino en términos de la cantidad en que la demanda total excede la capacidad de la instalación, y se utiliza para dirigir el algoritmo genético hacia soluciones en las que todas las asignaciones cumplen con las restricciones de capacidad.

Por lo tanto, el costo total de “56276505.3995” significa que la solución propuesta por el algoritmo genético asignó demanda a las instalaciones de tal manera que la demanda total asignada a algunas instalaciones excedió su capacidad de producción.

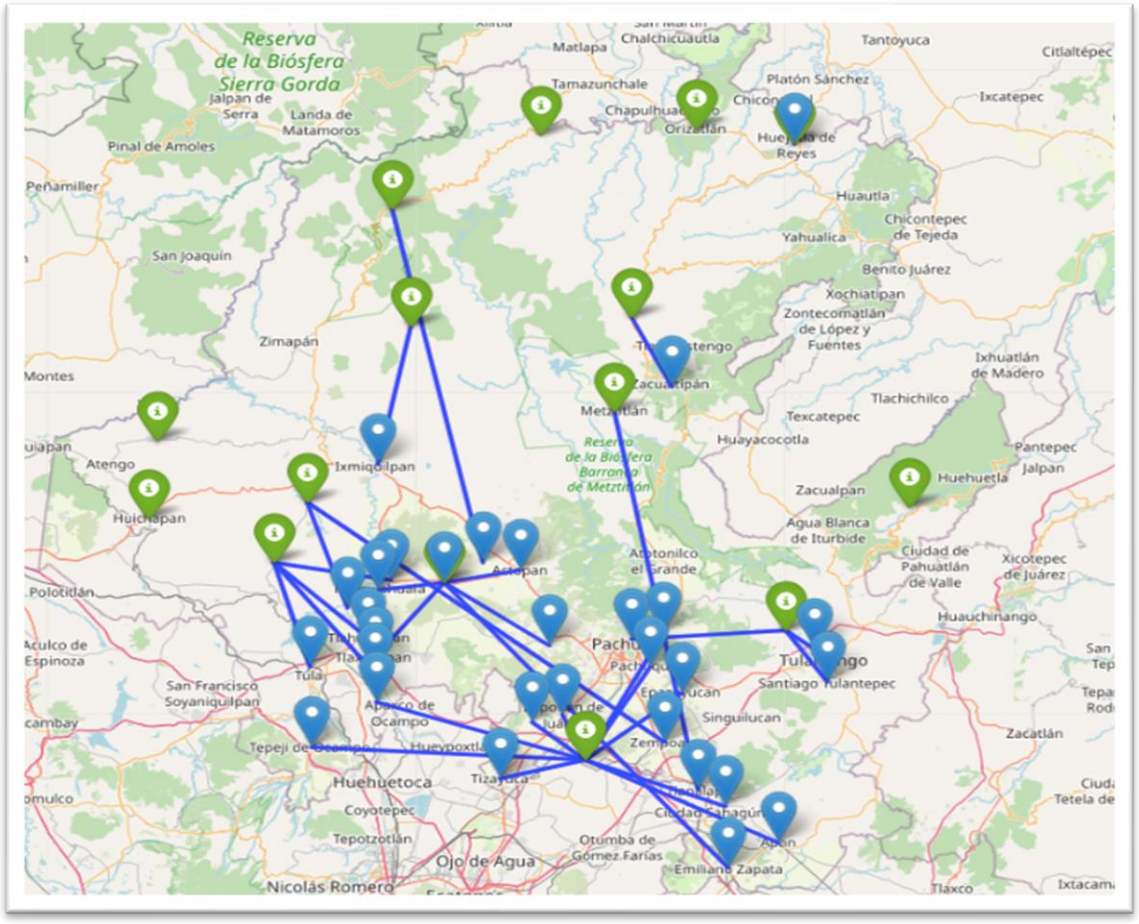
El valor exacto del costo total se calcula sumando el exceso de demanda (la cantidad por la cual la demanda asignada a una instalación excede su capacidad de producción) para todas las instalaciones, y luego multiplicando esa suma por 1000 (según la implementación de la función calcular_costo_total () en el código), esto debido a la forma del algoritmo dónde se multiplica por esa cantidad para que las siguientes generaciones tengan en cuenta ese multiplicador si es que no son aptas. (Cicirello & Smith, 2000).

Por lo tanto, un costo total de 56276505.3995 sugiere que la demanda total asignada a las instalaciones excedió la capacidad total de producción de las instalaciones por 56276.505 toneladas (ya que el costo se calcula multiplicando el exceso de demanda por 1000).

Por lo tanto, aunque el algoritmo genético está tratando de minimizar tanto la distancia total como el costo total, en este caso particular, ha encontrado una solución que minimiza la distancia total, pero resulta en un exceso de demanda asignada a las instalaciones. Este es un resultado común en los problemas de optimización multiobjetivo: a menudo hay un compromiso entre los diferentes objetivos que se están tratando de minimizar o maximizar.

Figura 14

Mapa de resultados del algoritmo genético para determinar los centros



Nota: Los puntos verdes hacen mención a los posibles centros, mientras que los azules a los posibles clientes y las líneas azules sus posibles trayectos.

En el mapa se llega a ver lógicamente cómo todo lo que se está diciendo cae en que las instalaciones con menor cercanía son mejores para proveer dicha demanda, aunque hay que tomar en cuenta que en este problema no se toman en cuenta clientes reales sino posibles clientes que probablemente comprarían dicha carne con base en el IDH, y esto puede no ser una medida que realmente sea verdad en la vida real, sin embargo nos da un gran aproximado, ya que sabemos que dichos clientes tienen que importar esas cantidades de carne, ya sean de un rastro TIF o no. Además, se tiene que tomar en cuenta qué lo s clientes asignado solamente son del Estado de Hidalgo, y pueda que esto no sea cierto en la vida real, ya que un rastro TIF también considera exportar la carne a otros países.

4.30 Problema TSP de ida y regreso por un municipio para determinar la localización de los centros

Ahora se procede a resolver un problema de optimización de logística comúnmente enfrentado por empresas de distribución y transporte. Se busca determinar cuál de varias instalaciones es la más adecuada para atender a un conjunto de clientes, minimizando la distancia total de viaje, simulando como si un transporte hiciera un viaje de ida a un municipio, regresara al centro y pasara al siguiente municipio. Esta simulación es especialmente útil debido a la cantidad de reses que pueda llevar un camión.

Para resolver este problema, se aplica el algoritmo de optimización de rutas del Vehículo de Viaje (Vehicle Routing Problem, VRP) utilizando la biblioteca OR-Tools de Google. Específicamente, se utiliza la estrategia PATH_CHEAPEST_ARC, que selecciona la arista más barata para expandir la solución en cada iteración. («Problema con el vendedor en viajes», s. f.). («Opciones de enrutamiento», s. f.).

Funcionamiento del Código

1. Datos de Entrada: Se definen las coordenadas geográficas de las instalaciones y clientes en las listas instalaciones y clientes.
2. Cálculo de Distancias: Se utiliza la fórmula de haversine para calcular la distancia entre dos puntos en la superficie terrestre.
3. Creación del Modelo de Datos: Se genera una matriz de distancias que representa las distancias entre las instalaciones y los clientes.
4. Resolución del Problema de Enrutamiento: Para cada instalación, se resuelve un problema de enrutamiento que calcula la distancia total de viaje para atender a todos los clientes desde esa instalación. Se utiliza el algoritmo VRP de OR-Tools con la estrategia PATH_CHEAPEST_ARC. . («Problema con el vendedor en viajes», s. f.). («Opciones de enrutamiento», s. f.).
5. Selección de la Mejor Instalación: Se selecciona la instalación que resulta en la menor distancia total de viaje.

6. Visualización en Mapa: Se crea un mapa interactivo que muestra las instalaciones, los clientes y las rutas seleccionadas utilizando la biblioteca folium.

O visto de otra manera:

1. Importaciones y datos de entrada:

   - Se importan las bibliotecas y módulos necesarios.
   - Se definen las listas instalaciones y clientes que contienen la información de las instalaciones y los clientes respectivamente.

2. Función haversine:

   - Esta función calcula la distancia entre dos puntos geográficos (dadas sus latitudes y longitudes) usando la fórmula de haversine. La distancia se devuelve en kilómetros. Función create_data_model: Esta función crea un modelo de datos a partir de una lista de ubicaciones. Específicamente, genera una matriz de distancias que indica la distancia entre cada par de ubicaciones en la lista. Se utiliza la función haversine para calcular las distancias entre ubicaciones.

3. Función main:

   - La función principal (main) itera sobre cada instalación y calcula la distancia total entre dicha instalación y todos los clientes.
   - Para cada instalación, se crea un modelo de enrutamiento usando ortools y se resuelve para encontrar la distancia más corta entre la instalación y cada cliente.
   - Luego, suma las distancias para obtener la distancia total de la ruta para esa instalación.
   - Si la distancia total para una instalación es menor que la distancia mínima encontrada hasta ahora, se actualiza la distancia mínima y se establece esa instalación como la mejor instalación.

- Al final del bucle, se imprime la mejor instalación y su distancia total.

4. Visualización en el mapa:

   - Se crea un mapa centrado en la mejor instalación usando folium.
   - Se agregan marcadores para cada instalación y cliente en el mapa.
   - Se conectan la mejor instalación y cada cliente con líneas en el mapa para mostrar las rutas.
   - Finalmente, se muestra el mapa.

5. Ejecución del código:

   - Se llama a la función main para ejecutar el código.

El código se basa en una variante del Problema del Viajante de Comercio (TSP). El TSP clásico busca encontrar la ruta más corta que visita un conjunto de ciudades y regresa al punto de partida. La variante en cuestión busca encontrar, para cada instalación, la suma total de las distancias para visitar a todos los clientes y regresar a la instalación. La fórmula para calcular la distancia total para una instalación específica sería:

$$\text{DistanciaTotal}_i = \sum_{j=1}^{n} \text{Distancia}_{ij}$$

Donde:

- (i) es una instalación específica.

- (j) es un cliente específico.

- (n) es el número total de clientes.

- (Distancia_{ij}) es la distancia de la instalación (i) al cliente (j) y de regreso a la instalación (i).

Además, para identificar cuál de las instalaciones tiene la menor distancia total, se utiliza la siguiente lógica:

$$\text{MejorInstalación} = \min_{i} \text{DistanciaTotal}_i$$

Donde:

- (MejorInstalación) es la instalación con la menor distancia total.
- (i) recorre todas las instalaciones.

La estrategia "PATH_CHEAPEST_ARC" es una heurística para resolver el TSP. La idea es simple: en cada paso, selecciona el arco (o ruta) más barato disponible y lo agrega a la solución, repitiendo este proceso hasta que todas las ciudades (o puntos) estén en la ruta.

Matemáticamente, la estrategia se puede describir de la siguiente manera:

1. Comenzando desde un nodo inicial (i), busca el nodo (j) tal que la distancia ($d(i,j)$) sea mínima y (j) aún no esté en la ruta.
2. Haz (j) el nuevo nodo actual y repite el paso 1 hasta que todos los nodos estén en la ruta.

La fórmula para encontrar el nodo (j) en cada paso sería:

$$j^* = \arg\min_{j \notin \text{Ruta}} d\,(i,j)$$

Donde:

- (j^*) es el nodo seleccionado.
- (i) es el nodo actual.
- ($d(i,j)$) es la distancia entre el nodo (i) y el nodo (j).

Es importante señalar que, aunque "PATH_CHEAPEST_ARC" es eficiente y a menudo produce soluciones de buena calidad, no garantiza la solución óptima. Es una heurística, lo que significa que es una aproximación rápida para encontrar una solución aceptable, pero no necesariamente la mejor.

Código: Ver ANEXO C: CÓDIGO 3.

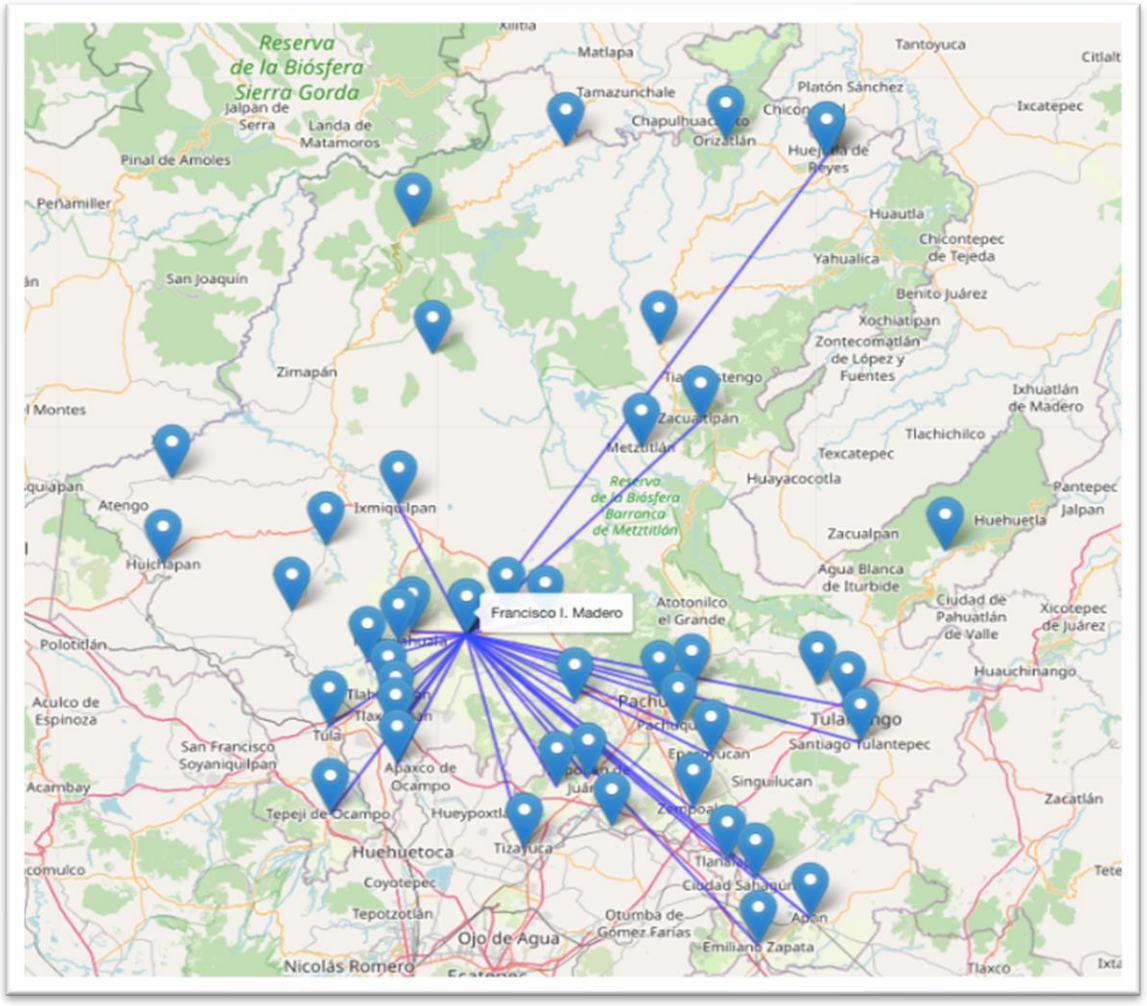
Resultado:

Instalación	Distancia total de la ruta (km)
Chapantongo	3898
Chapulhuacán	7296
San Bartolo Tutotepec	5200
Jacala de Ledezma	6644
Huejutla de Reyes	7674
Molango de Escamilla	5154
Alfajayucan	3980
Huichapan	5380
Villa de Tezontepec	2702
Francisco I. Madero	**2618**
Acatlán	3544
Nicolás Flores	5172
San Felipe Orizatlán	7544
Tecozautla	5754
Metztitlán	4094

Mejor instalación	
Francisco I. Madero	**2618**

Figura 15

Mapa de resultados del algoritmo TSP uno a uno para determinar los centros



Nota: Los resultados se muestran en líneas azules que muestra la ruta a tomar, mientras que los puntos azules los clientes y la parte en que menciona el nombre es el centro.

Interpretación de los Resultados

Los resultados muestran la distancia total de la ruta para cada instalación y la mejor instalación encontrada. En este caso específico, la mejor instalación es "Francisco I. Madero", con una distancia total de la ruta de 2618 km.

Este resultado significa que, de todas las instalaciones disponibles, "Francisco I. Madero" es la más eficiente en términos de distancia total para atender a los clientes. La selección

de esta instalación podría llevar a una reducción significativa en los costos de transporte y tiempo de entrega.

El código es una poderosa herramienta para la planificación de rutas y la toma de decisiones en logística. Utilizando el algoritmo VRP y la estrategia específica de PATH_CHEAPEST_ARC, permite a las empresas identificar la mejor ubicación para la distribución, optimizando así los recursos y mejorando la eficiencia operativa. («Problema con el enrutamiento del vehículo», s. f.). («Opciones de enrutamiento», s. f.-b).

4.31 Problema TSP tradicional para la localización de los centros

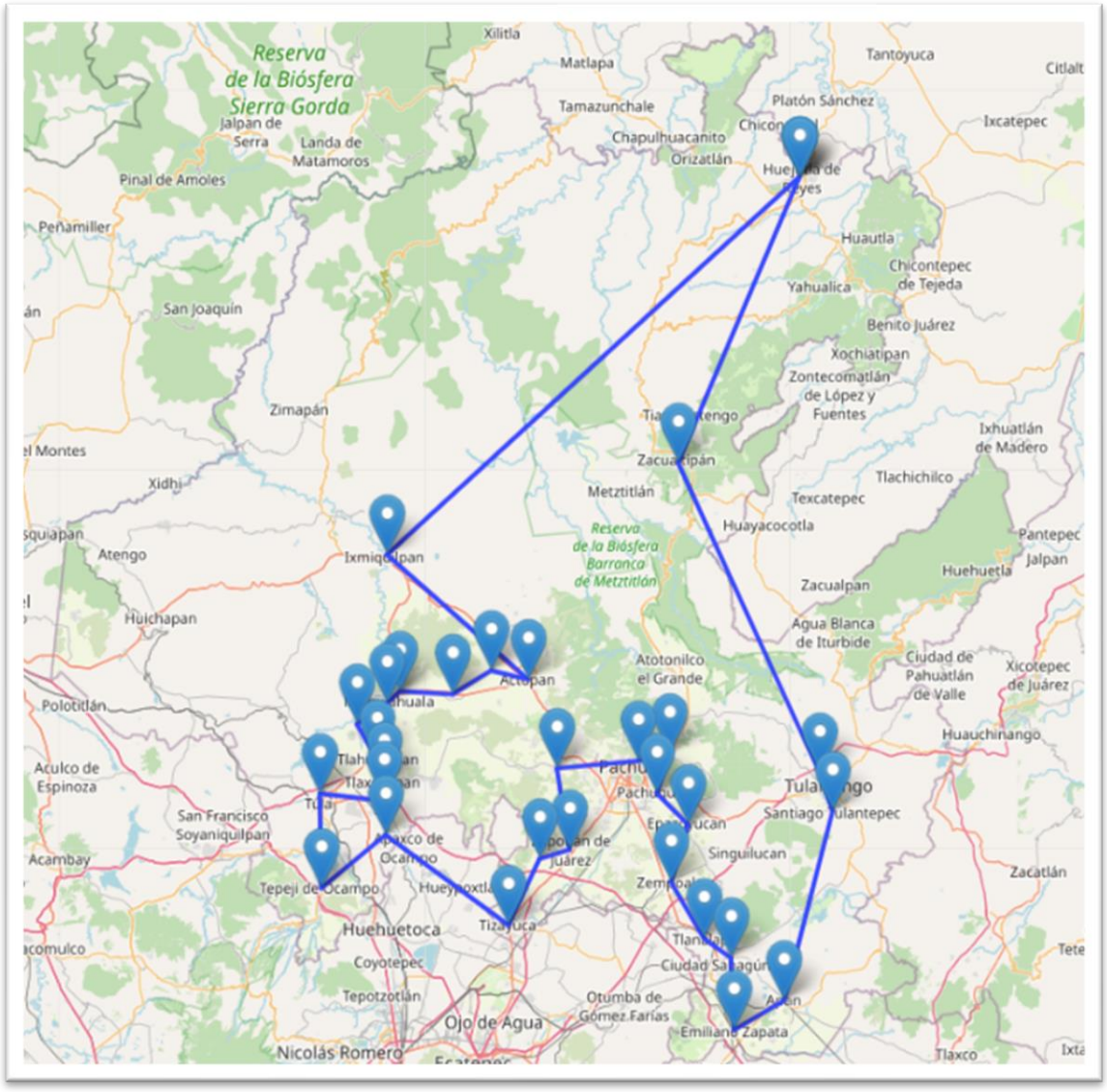
Viéndolo desde este punto con esa variación, ahora se resolverá viendo el problema de la manera clásica dónde se aborda el problema TSP, o sea que el transporte pasara por todos los municipios para poder llegar de nuevo al centro, dónde lo único que se cambia es esa parte.

Para lo que se obtiene el siguiente resultado:

Instalación	**Distancia total de la ruta (km)**
Chapantongo	611
Chapulhuacán	590
San Bartolo Tutotepec	649
Jacala de Ledezma	598
Huejutla de Reyes	**568**
Molango de Escamilla	581
Alfajayucan	580
Huichapan	651
Villa de Tezontepec	608
Francisco I. Madero	580
Acatlán	618
Nicolás Flores	576
San Felipe Orizatlán	576
Tecozautla	645
Metztitlán	581
Mejor instalación	
Huejutla de Reyes	**568**

Figura 16

Mapa de resultados del algoritmo TSP normal para determinar los centros



Nota: Los resultados se muestran en líneas azules que muestra la ruta a tomar.

Dónde ahora cómo podemos ver la mejor instalación es Huejutla de Reyes con una distancia total de 568 Km.

4.32 Problema VRP para la localización de los centros

Ahora para checar un problema común en las empresas y en este caso para los centros, sería la de un problema VRP dónde en un contexto de logística y distribución, las empresas a menudo enfrentan el desafío de determinar la mejor manera de asignar vehículos a rutas y clientes para minimizar la distancia total recorrida. El código proporcionado aborda este problema utilizando un enfoque de agrupamiento junto con el problema de enrutamiento de vehículos (VRP). La idea es agrupar clientes y asignarlos a instalaciones, y luego resolver el VRP para cada grupo, teniendo en cuenta una simulación en dónde se tienen diferentes vehículos, estos seleccionados por un algoritmo de clustering que tienen la tarea de tener los mejores municipios con relación al punto de inicio para después pasara por cada uno de ellos y regresar al punto de origen, esto teniendo en cuenta la ruta óptima del problema TSP. («Problema con el enrutamiento del vehículo», s. f.). («Opciones de enrutamiento», s. f.-b).

Funcionamiento del Código

Datos de Entrada: Se definen las coordenadas geográficas de las instalaciones y clientes.
Cálculo de Distancias: Se utiliza la fórmula de haversine para calcular la distancia entre dos puntos en la superficie terrestre.
Agrupamiento de Clientes: Se aplica el algoritmo K-Medoids para agrupar los clientes en tres grupos. K-Medoids es un método de agrupamiento que selecciona puntos reales del conjunto de datos como centros.
Resolución del VRP para Cada Grupo: Para cada instalación, se resuelve un VRP para cada grupo de clientes utilizando la estrategia PATH_CHEAPEST_ARC. («Opciones de enrutamiento», s. f.-b) (Maranzana F. E., 1963). (Park H. S. & Jun C. H., 2009).

Visualización en Mapa: Se crea un mapa interactivo utilizando folium para visualizar las instalaciones, los clientes, y las rutas seleccionadas.

Dando estos resultados:

Instalación	Grupo 1 (km)	Grupo 2 (km)	Grupo 3 (km)	Distancia Total (km)
Chapantongo	289.002	336.606	627.173	1252.781
Chapulhuacán	172.921	450.283	733.65	1356.854
San Bartolo Tutotepec	199.863	468.948	500.955	1169.766
Jacala de Ledezma	209.77	414.207	717.649	1341.626
Huejutla de Reyes	120.089	491.074	656.347	1267.51
Molango de Escamilla	128.437	392.207	663.398	1184.042
Alfajayucan	263.215	331.671	630.162	1225.048
Huichapan	321.764	388.759	680.27	1390.792
Villa de Tezontepec	292.992	359.698	477.761	1130.451
Francisco I. Madero	244.894	309.621	564.826	1119.342
Acatlán	**230.439**	**414.23**	**438.3**	**1082.968**
Nicolás Flores	199.848	361.581	665.953	1227.382
San Felipe Orizatlán	138.403	475.148	746.165	1359.716
Tecozautla	305.99	391.98	694.614	1392.584
Metztitlán	143.178	394.51	623.192	1160.879
Acatlán	230.439	414.23	438.3	1082.968
		Mejor instalación		
Acatlán	**230.439**	**414.23**	**438.3**	**1082.968**

Interpretación de los Resultados

Este estudio presenta una solución al problema de Ruteo de Vehículos (VRP) enfocada en optimizar los costos de transporte, entendiendo que minimizar la distancia total recorrida es clave para reducir estos costos. En lugar de enfocarnos en minimizar la demanda de los clientes, el objetivo principal es modelar eficientemente los gastos de transporte. Para lograr esto, se ha implementado una variante del VRP que incluye una fase de agrupación de clientes utilizando el algoritmo K-Medoids.

Se ha dividido los clientes en tres grupos distintos basados en el resultado óptimo obtenido de K-Medoids. La calidad y eficacia de estos grupos se evaluaron utilizando el método de Silhouette y la visualización a través de un Dendrograma.

Estas herramientas nos permitieron determinar la cohesión interna de los clústeres y visualizar la estructura jerárquica del agrupamiento. A continuación, cada grupo de clientes se asignó a una instalación específica, y se optimizaron las rutas de entrega para cada grupo utilizando el algoritmo VRP de OR-Tools.

Es importante destacar que en un VRP tradicional, es crucial tener información detallada sobre las limitaciones de número de vehículos y su capacidad máxima. Sin embargo, en nuestro enfoque, hemos priorizado la eficiencia en las rutas de transporte sobre estas restricciones. Con estos métodos, hemos podido determinar rutas óptimas para cada grupo de clientes, resultando en una solución eficiente para el VRP en nuestro caso de estudio. Las gráficas resultantes de los métodos de Silhouette y el Dendrograma, así como las rutas optimizadas, proporcionan una visión clara de la efectividad de nuestro enfoque:

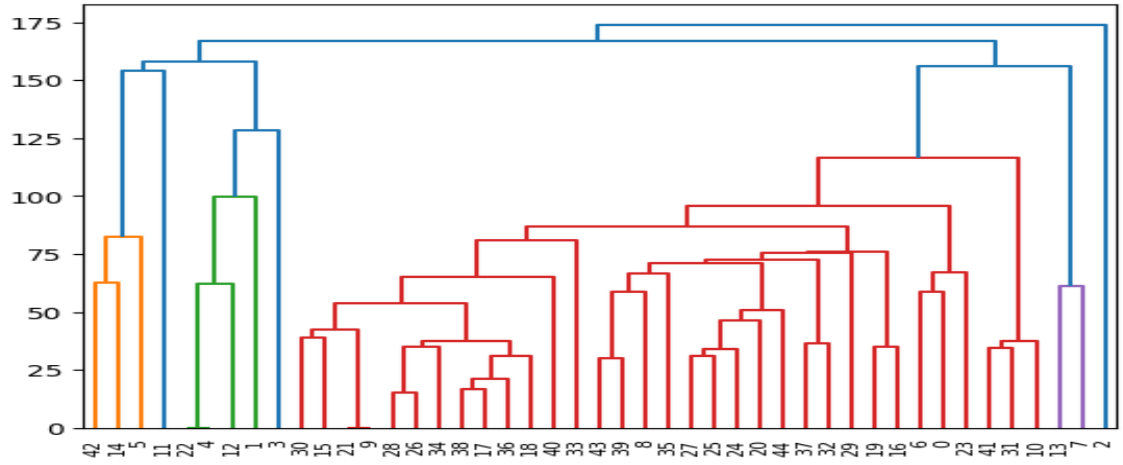
Gráfica 52

Método de Silhouette para óptimo de k para el algoritmo Kmedoids del VRP

Nota: k se refiere a la cantidad de los clústeres y el óptimo se define en el punto más alto del Silhouette score, que para este caso son 3 clústeres.

Figura 17

Dendrograma de los clústeres del algortimo Kmdoids del VRP



Nota: la cantidad de clústeres óptimos de igual manera se ve que es 3 dado el alcance de estos para visualizar a los datos, este puede ser modificado según sea necesario.

Tabla 9

Tabla de ejemplificación de los vehículos ocupados para cada ruta del VRP

Clúster	Municipios	Demanda Total	Vehículos Diarios con capacidad de 48 cabezas
Clúster 1	Huejutla de Reyes, Zacualtipán de Ángeles	7044.083403	1
Clúster 2	Actopan, Atitalaquia, Atotonilco de Tula, Francisco I. Madero, Ixmiquilpan, Mixquiahuala de Juárez, Progreso de Obregón, San Salvador, Tepeji del Río de Ocampo, Tezontepec de Aldama, Tlahuelilpan, Tlaxcoapan, Tula de Allende	23038.47899	2
Clúster 3	Apan, Emiliano Zapata, Epazoyucan, Mineral de La Reforma, Mineral del Monte, Pachuca de Soto, San Agustín Tlaxiaca, Santiago Tulantepec de Lugo Guerrero, Tepeapulco, Tizayuca, Tlanalapa, Tolcayuca, Tulancingo de Bravo, Zapotlán de Juárez, Zempoala	68951.18439	4

Nota: Esta tabla muestra la demanda total y el número estimado de vehículos diarios necesarios para cada clúster, basándose en una capacidad de vehículo simulada de 48 cabezas de ganado bovino. Los cálculos se han realizado sumando las demandas diarias de cada municipio por clúster, las cuales se obtuvieron dividiendo la demanda anual entre 365 días. Posteriormente, esta suma se dividió por la capacidad máxima del vehículo (48 cabezas) y se redondeó al entero superior más cercano para estimar el número de vehículos necesarios. Es importante destacar que esta simulación no refleja necesariamente las capacidades reales de los vehículos en un contexto práctico, ya que en

la realidad, las capacidades de transporte pueden variar significativamente. Además, el modelo asume una distribución uniforme de la demanda a lo largo del año, lo cual podría no ser el caso en situaciones reales. Por lo tanto, los resultados presentados aquí deben considerarse como una aproximación teórica basada en suposiciones simplificadas, útiles para la comprensión y análisis del problema de Ruteo de Vehículos en un contexto académico o de simulación.

La mejor instalación encontrada fue "Acatlán", con una distancia total de la ruta de 1082.97 kilómetros. Esto significa que, de todas las instalaciones disponibles, Acatlán ofrece la menor distancia total para atender a todos los grupos de clientes.

Las siguientes son algunas de las distancias para las diferentes instalaciones:

- Chapantongo: 1252.78 kilómetros
- Chapulhuacán: 1356.85 kilómetros
- Acatlán: 1082.97 kilómetros (mejor instalación)
- Metztitlán: 1160.88 kilómetros

El código combina el algoritmo K-Medoids para agrupar a los clientes y el VRP para determinar la mejor manera de atenderlos desde diferentes instalaciones. La solución optimizada puede ayudar a las empresas a reducir significativamente los costos de transporte y mejorar la eficiencia operativa, en este caso se pretende disminuir el costo para el rastro TIF pero se menciona el poder de la herramienta. («Problema con el enrutamiento del vehículo», s. f.) («Opciones de enrutamiento», s. f.-b) (Maranzana F. E., 1963). (Park H. S. & Jun C.H., 2009).

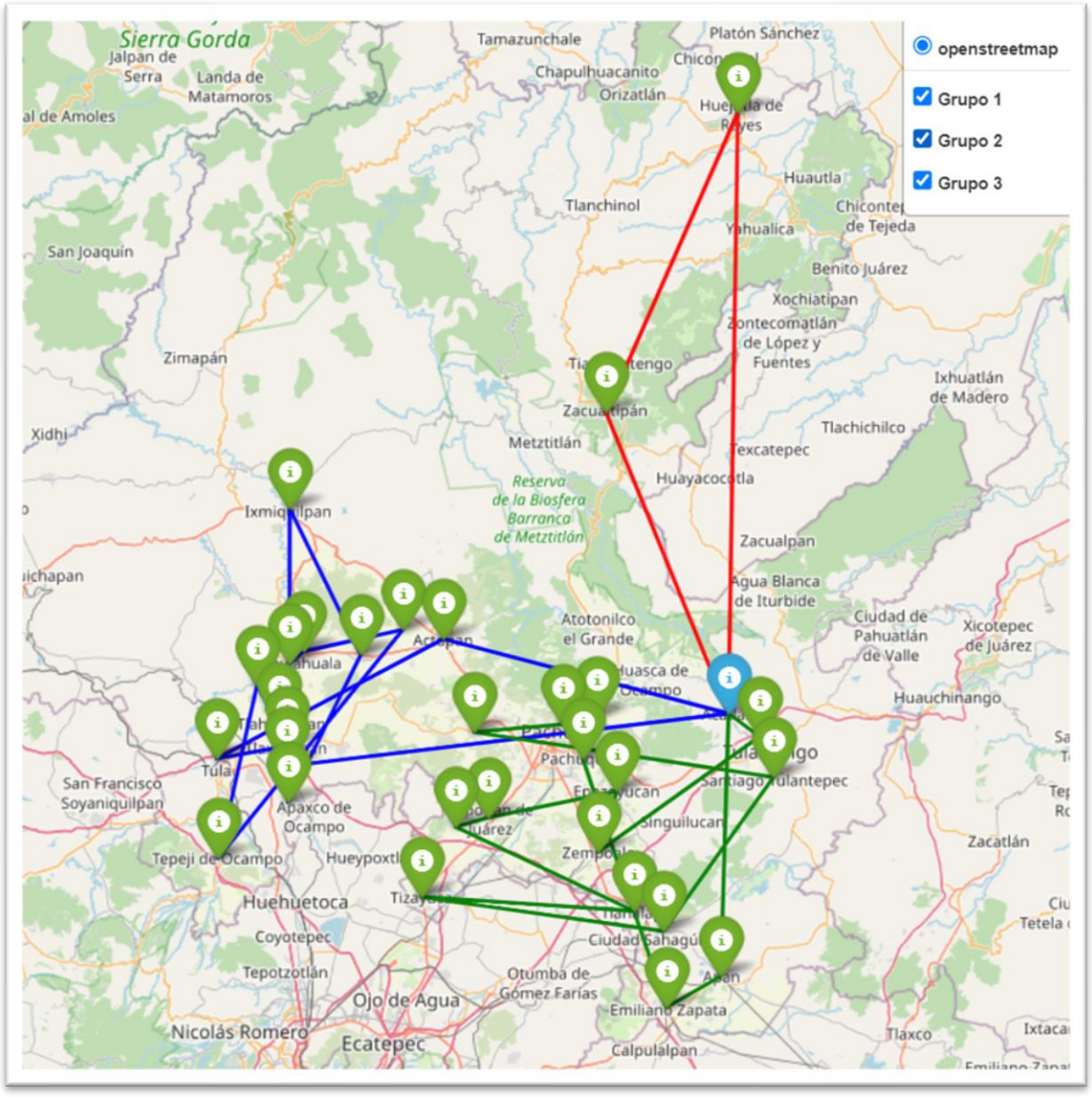
La visualización en el mapa proporciona una comprensión clara de cómo se asignan las rutas y cómo los clientes se dividen en diferentes grupos, lo cual es valioso para los planificadores y tomadores de decisiones en la industria de la logística.

La selección de la instalación "Acatlán" como la más eficiente en términos de distancia total puede guiar las decisiones estratégicas en términos de dónde enfocar los recursos de distribución y cómo planificar las operaciones de entrega.

Este enfoque combina técnicas de agrupamiento y optimización de rutas para proporcionar una solución integral a un problema común en la logística de distribución, demostrando el poder de los métodos de aprendizaje automático y optimización en la toma de decisiones empresariales.

Figura 18

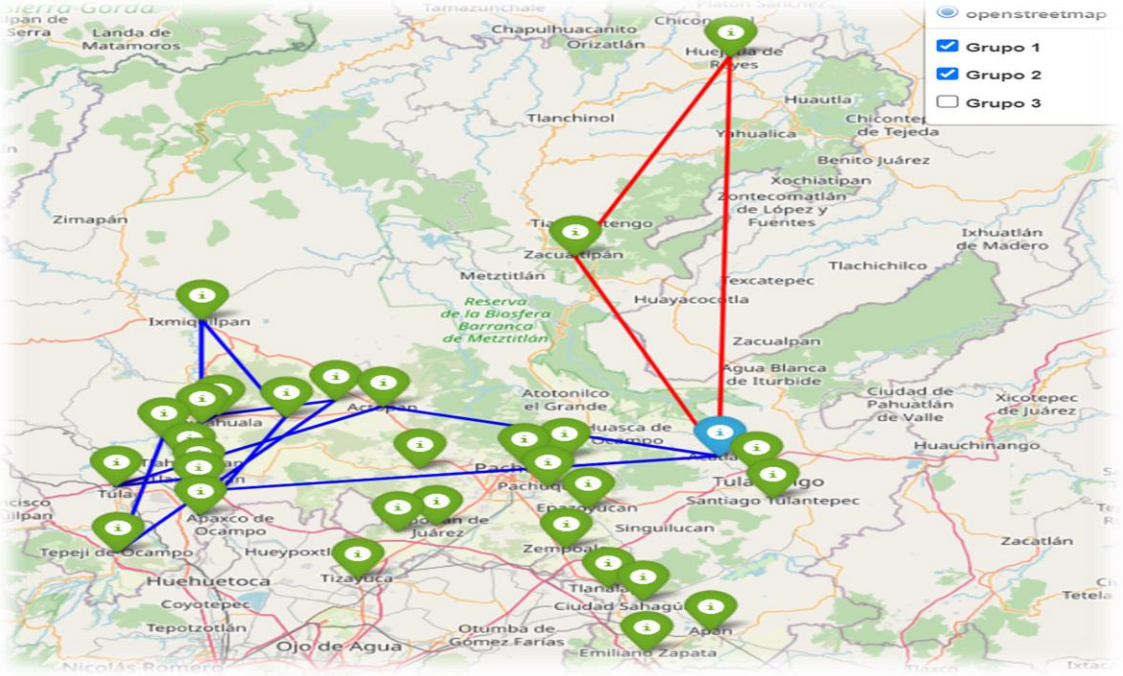
Mapa de resultados del algoritmo VRP para determinar los centros 1



Nota: El mapa muestra las tres rutas de los posibles 3 grupos de municipios en una línea rojas, azul y verde desde el punto de partida que es Acatlán que se muestra con un punto azul y los puntos verdes los posibles clientes, así como en la parte superior derecha muestra la selección de las rutas de cada grupo.

Figura 19

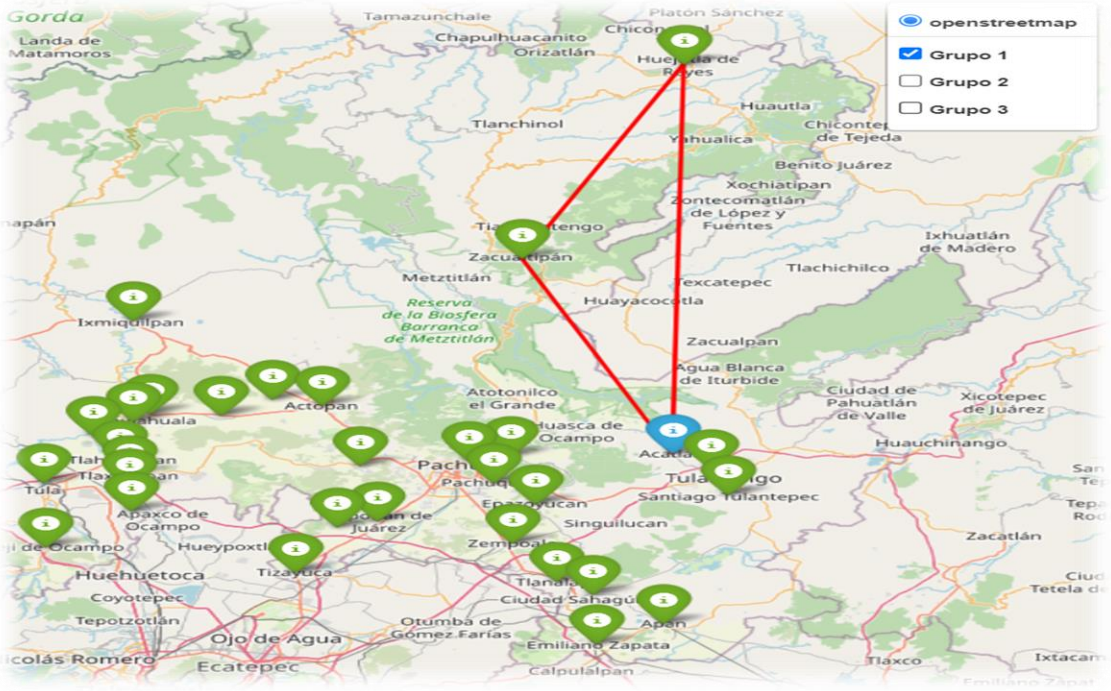
Mapa de resultados del algoritmo VRP para determinar los centros 2



Nota: El mapa muestra las dos rutas de los 3 grupos.

Figura 20

Mapa de resultados del algoritmo VRP para determinar los centros 3



Nota: El mapa muestra una ruta de uno de los tres grupos de municipios.

Ahora ya se tienen los mejores municipios con relación a sus costos, esto es especialmente importante, debido a que estos se compararán con los posibles lugares de localización del rastro, que tendrá una agrupación diferente debido a varias consideraciones.

Para empezar se ilustran los resultados de los mejores centros, estos sacados de los métodos ya antes mencionados, los cuales son:

Tabla 10
Tabla de los posibles centros basados en los métodos utilizados

Municipio	Latitud	Longitud
Chapantongo	20.28559483	-99.4131928
Chapulhuacán	21.15787458	-98.9043581
San Bartolo Tutotepec	20.3991586	-98.2020594
Jacala de Ledezma	21.00849043	-99.1885371
Huejutla de Reyes	21.13959058	-98.4204449
Molango de Escamilla	20.78658031	-98.7306035
Alfajayucan	20.41015828	-99.3494646
Huichapan	20.37553905	-99.6510293
Villa de Tezontepec	19.879986	-98.8193074
Francisco I. Madero	20.24540323	-99.0888141
Acatlán	20.14599753	-98.4383528
Nicolás Flores	20.76804674	-99.1505642
San Felipe Orizatlán	21.17106016	-98.6075891
Tecozautla	20.53415978	-99.6349425
Metztitlán	20.5948676	-98.7642084

Ahora bien para poder analizar estos centros en comparación de las diferentes localizaciones del rastro tenemos que elegir cuales serían los municipios a considerar, para ello primero, se verán cuáles son los municipios que posiblemente exportan como anteriormente lo hicimos, pero con un pequeño filtro, con base en el costo de puesta en marcha de un programa destinado a la identificación de ganado para rastro TIF; por ejemplo, que este no contenga algunas enfermedades. Para ello es imprescindible contar con información precisa, dado que esta información solamente está medio puesta a disposición y eso por una nota de gobierno federal que se llama: “LA SIERRA Y LA HUASTECA HIDALGUENSE PODRÁN OBTENER RECONOCIMIENTO INTERNACIONAL PARA EXPORTAR GANADO BOVINO A ESTADOS UNIDOS”,

dónde menciona que el gobierno federal ya invirtió una cantidad de 23 millones de pesos para la erradicación de la tuberculosis bovina, lo colocaremos con el planteamiento de que este gasto no se vaya a hacer por falta de presupuesto en algunos otros municipios y por lo tanto aumenta considerablemente la posibilidad de éxito para su construcción, por lo que estos serían los municipios en proceso de reconocimiento para exportar este tipo de producto: Atlapexco, Huautla, Huazalingo, Huejutla, Jaltocán, San Felipe Orizatlán, Xochiatipan, Yahualica, Tianguistengo, Lolotla, Calnali, Tlanchinol, Tepehuacán de Guerrero, Molango de Escamilla, Xochicoatlán, Huehuetla, San Bartolo Tutotepec, Tenango de Doria, Zacualtipán, San Agustín Metzquititlán, Meztitlán, Eloxochitlán, Juárez Hidalgo, Tlahuiltepa, y Chapulhuacán. De los cuales esta tabla indica los municipios que ya probablemente exportan ganado, con sus respectivos pesos, que para este caso serían las posibles exportaciones:

Tabla 11

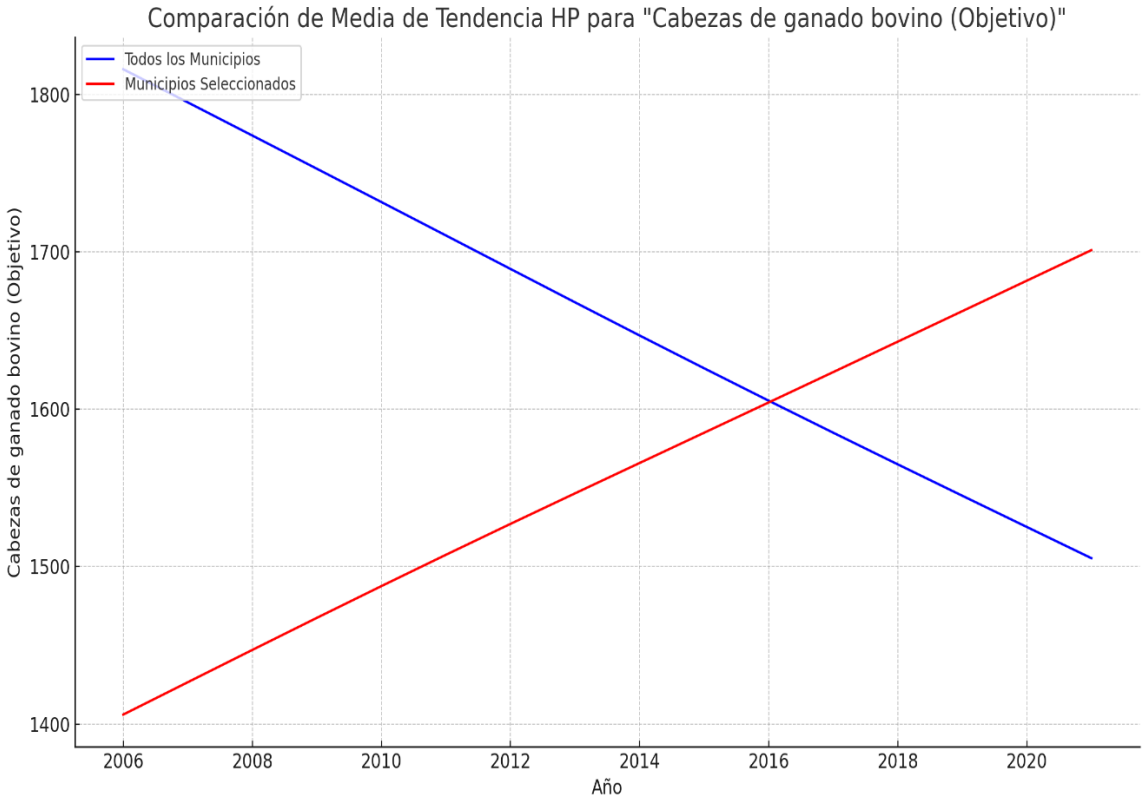
Tabla de las posibles ubicaciones de los rastros a evaluar

Municipio	Latitud	Longitud	Producción
Atlapexco	21.01745106	-98.3482964	70.18432667
Chapulhuacán	21.15787458	-98.9043581	2076.831214
Eloxochitlán	20.74687461	-98.8093052	535.8706135
Huautla	21.03188243	-98.2866852	1531.505859
Huazalingo	20.98056421	-98.5077423	45.81143524
Huehuetla	20.46035677	-98.0785954	1136.262735
Juárez Hidalgo	20.78338131	-98.8291823	317.6627911
Lolotla	20.84163503	-98.7171142	593.5610713
Molango de Escamilla	20.78658031	-98.7306035	769.5643214
San Agustín Metzquititlán	20.53306527	-98.6388013	25.68872557
San Bartolo Tutotepec	20.3991586	-98.2020594	1635.505363
San Felipe Orizatlán	21.17106016	-98.6075891	3193.503324
Tenango de Doria	20.33845235	-98.2267008	393.5998613
Tlahuiltepa	20.92389198	-98.9501706	236.2399636
Xochicoatlán	20.77704277	-98.6796684	164.6085326

Nota: La producción está dada en cabezas de ganado bovino.

Gráfica 53

Comparación de la media de la tendencia HP para "Cabezas de ganado bovino"



El gráfico ilustra las tendencias en la cantidad de cabezas de ganado bovino en el estado de Hidalgo. Mientras que la tendencia general para todos los municipios apunta hacia una disminución, un grupo seleccionado de municipios muestra un incremento evidente en esta variable objetivo. Estos municipios, que cuentan con una reserva de agua superior al promedio y han recibido inversión para fortalecer su área productiva, presentan una tendencia positiva respaldada por el filtro Hodrick-Prescott. Por lo tanto, resultan ser las ubicaciones más adecuadas para el establecimiento de un nuevo rastro.

Teniendo esto en cuenta se procede a generar el mismo proceso que con la elección de los mejores municipios para la ubicación de los centros. Para lo cual se seguirá con la misma estrategia que la elección de los centros, primero empezando por la clusterización.

4.33 Análisis de agrupamiento para identificar ubicaciones para el rastro

Primero cómo se había hecho antes, se seleccionan la cantidad de clústeres y ya que no se necesita una visualización profunda se le hará caso a lo establecido en dendrograma y en las gráficas de los métodos de Elbow y Silhouette, que para este caso un ideal serían 4, cómo lo imprime el mapa.

Gráfica 54

Gráficas del método de Elbow y del método de Silhouette para ubicación del rastro

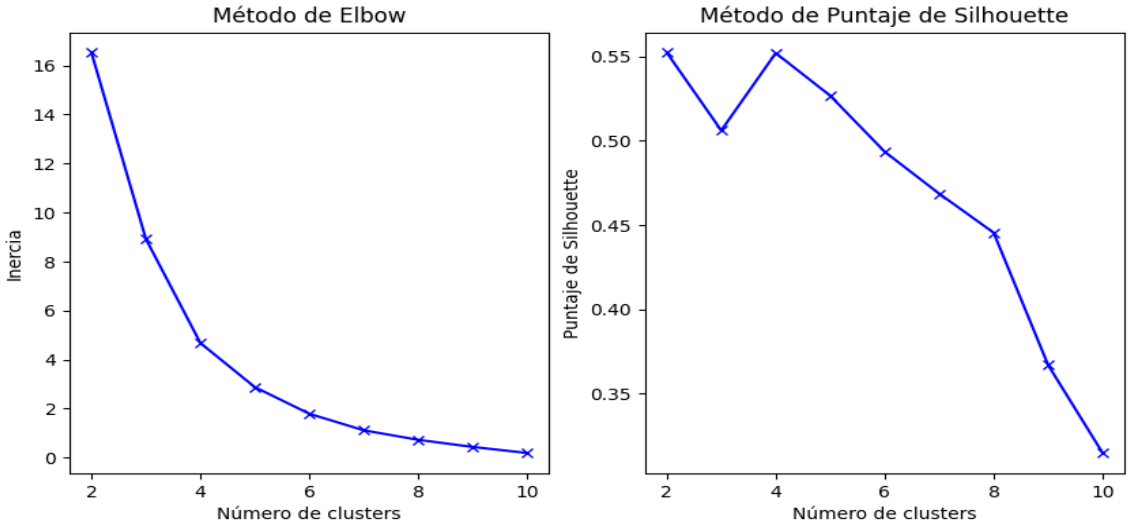


Figura 21

Dendrograma de clustering jerárquico para la ubicación del rastro

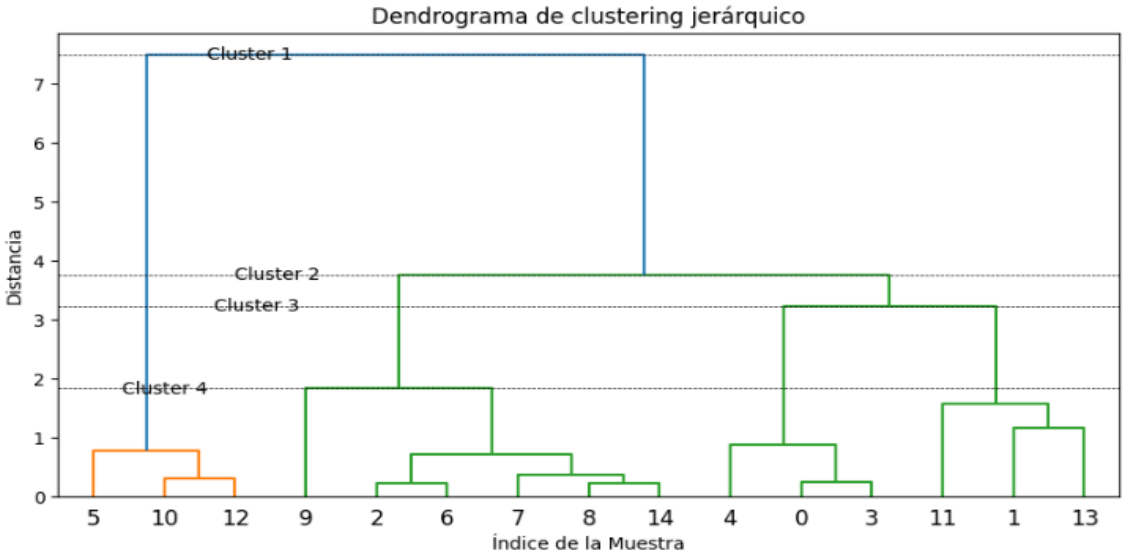


Figura 22

Mapa en 3d para los resultados del algoritmo de clustering para la ubicación del rastro

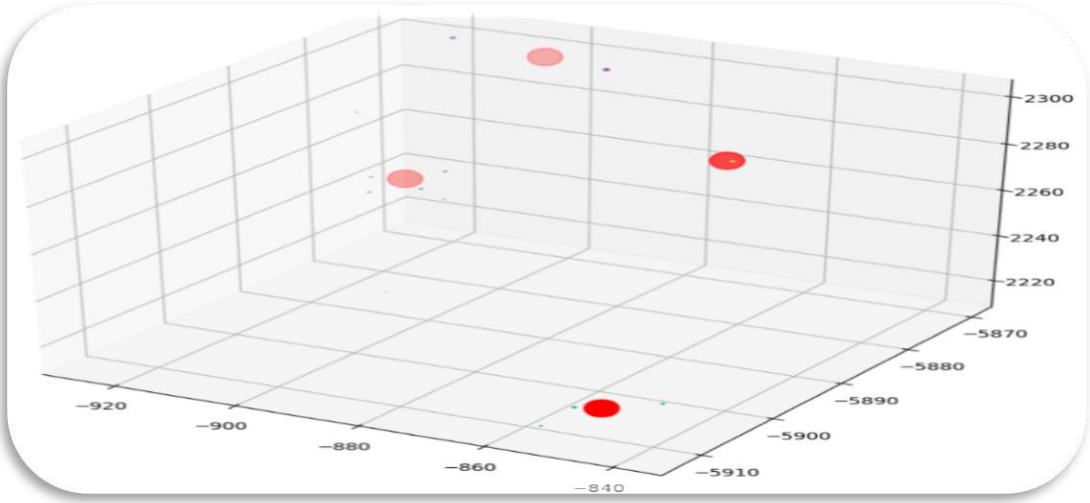
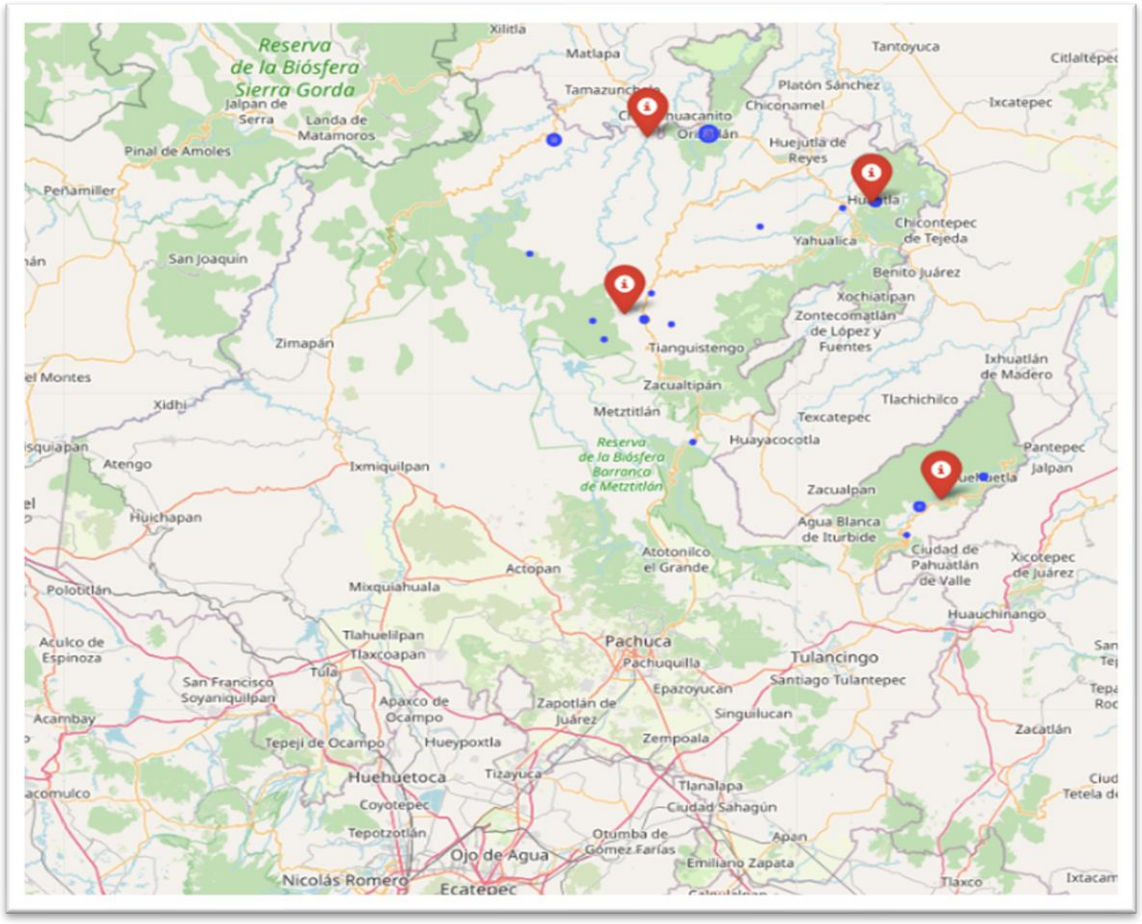


Figura 23

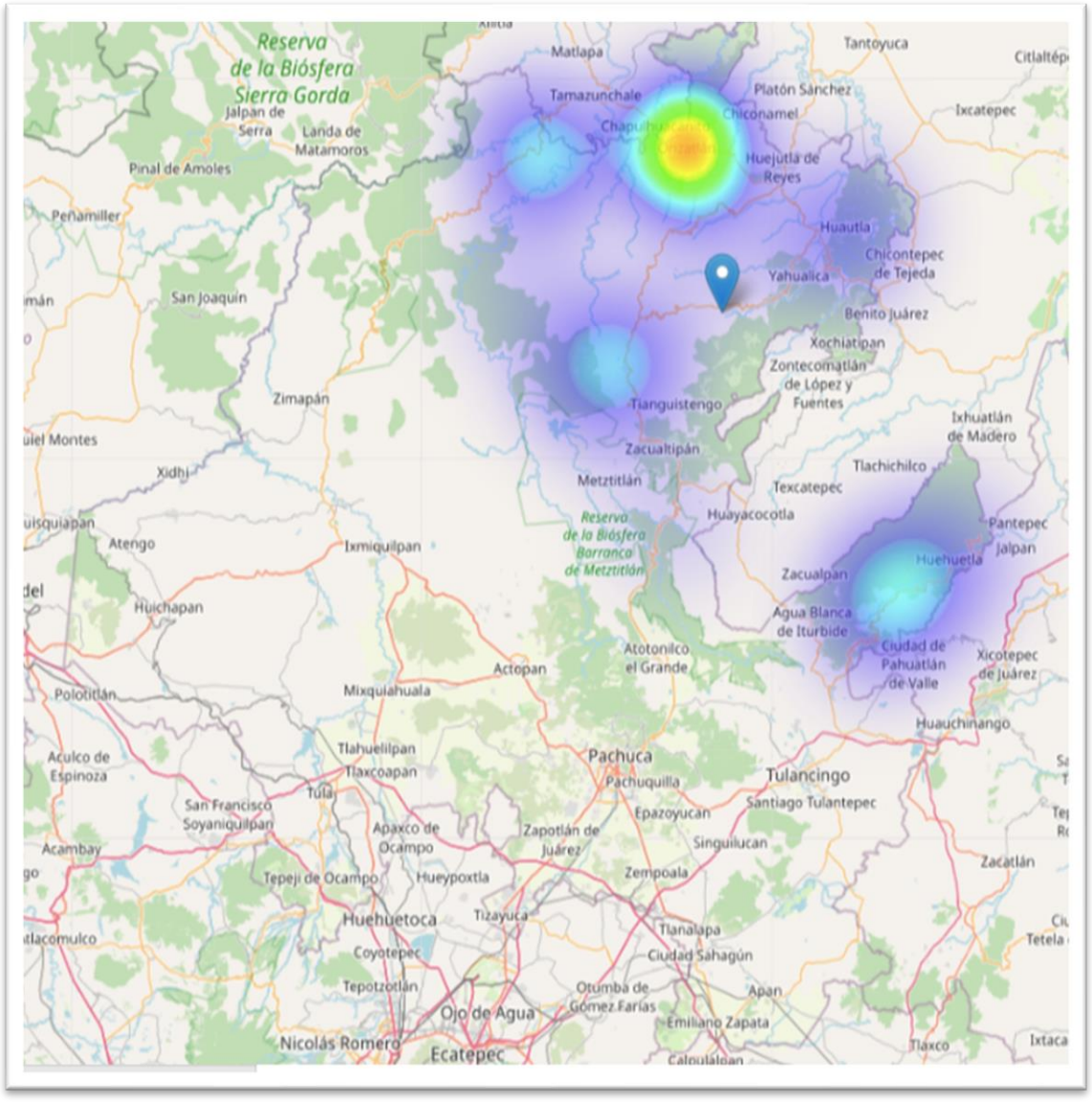
Mapa de resultados del algoritmo de clustering para el rastro



Ahora bien teniendo estos resultados se procederá cómo lo hicimos anteriormente a sacar el centro de gravedad que para este caso nos es importante por lo ya antes mencionado.

4.34 Centro de gravedad para identificar ubicaciones para rastros

Figura 24
Mapa de calor de los resultados de la técnica de centro de gravedad para el rastro



Ya obtenido el centro de gravedad y las posibles localizaciones para el rastro, se obtienen los siguientes datos:

Tabla 12

Tabla de las posibles ubicaciones de los rastros a evaluar a partir de los métodos

Municipio	Latitud	Longitud
San Bartolo Tutotepec	20.3991586	-98.2020594
Huejutla de Reyes	21.13959058	-98.4204449
Molango de Escamilla	20.78658031	-98.7306035
Calnali	20.89743021	-98.5835982
Tlanchinol	20.98951512	-98.6602832

Ahora para proseguir es importante tener estos datos junto con los datos anteriormente proporcionados de los centros, que ahora para este caso se convertirán en los clientes, para proceder como anteriormente se propuso con el algoritmo evolutivo.

4.35 Algoritmo Evolutivo NSGA 2 para localizar rastro.

Cómo se hizo anteriormente para los centros, se eligieron la misma cantidad de producción sólo que ahora como demanda para el rastro y para equilibrar dicha demanda se puso en predicción la cantidad total de la demanda de los clientes divida entre los 5 municipios dando una producción de 19,806.74937 por rastro.

Dándonos un resultado de:

Mejor Individuo (Asignaciones de Clientes a Instalaciones)

[2, 4, 0, 4, 1, 2, 2, 2, 0, 2, 0, 2, 1, 2, 2]

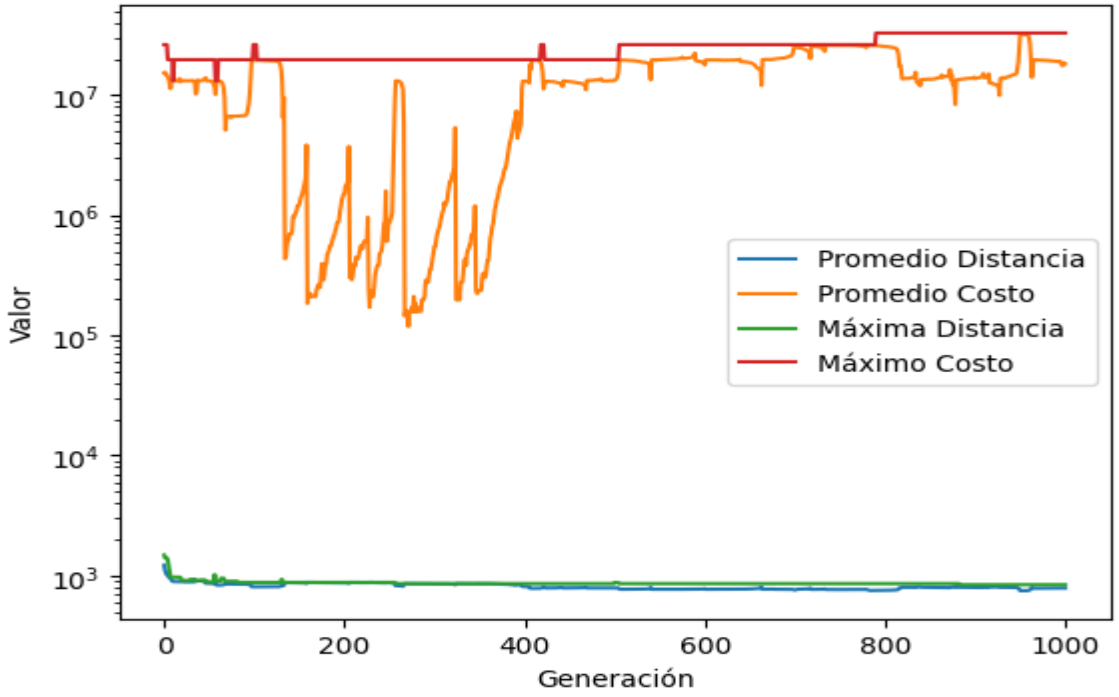
Distancia Total	Costo Total
737.6052504804638 km	33011248.95

Mejor Solución (Asignaciones)

	Cliente	Instalación	Atendido por
0	Chapantongo	2	Molango de Escamilla
1	Chapulhuacán	4	Tlanchinol
2	San Bartolo Tutotepec	0	San Bartolo Tutotepec
3	Jacala de Ledezma	4	Tlanchinol
4	Huejutla de Reyes	1	Huejutla de Reyes
5	Molango de Escamilla	2	Molango de Escamilla
6	Alfajayucan	2	Molango de Escamilla
7	Huichapan	2	Molango de Escamilla
8	Villa de Tezontepec	0	San Bartolo Tutotepec
9	Francisco I. Madero	2	Molango de Escamilla
10	Acatlán	0	San Bartolo Tutotepec
11	Nicolás Flores	2	Molango de Escamilla
12	San Felipe Orizatlán	1	Huejutla de Reyes
13	Tecozautla	2	Molango de Escamilla
14	Metztitlán	2	Molango de Escamilla

Gráfica 55

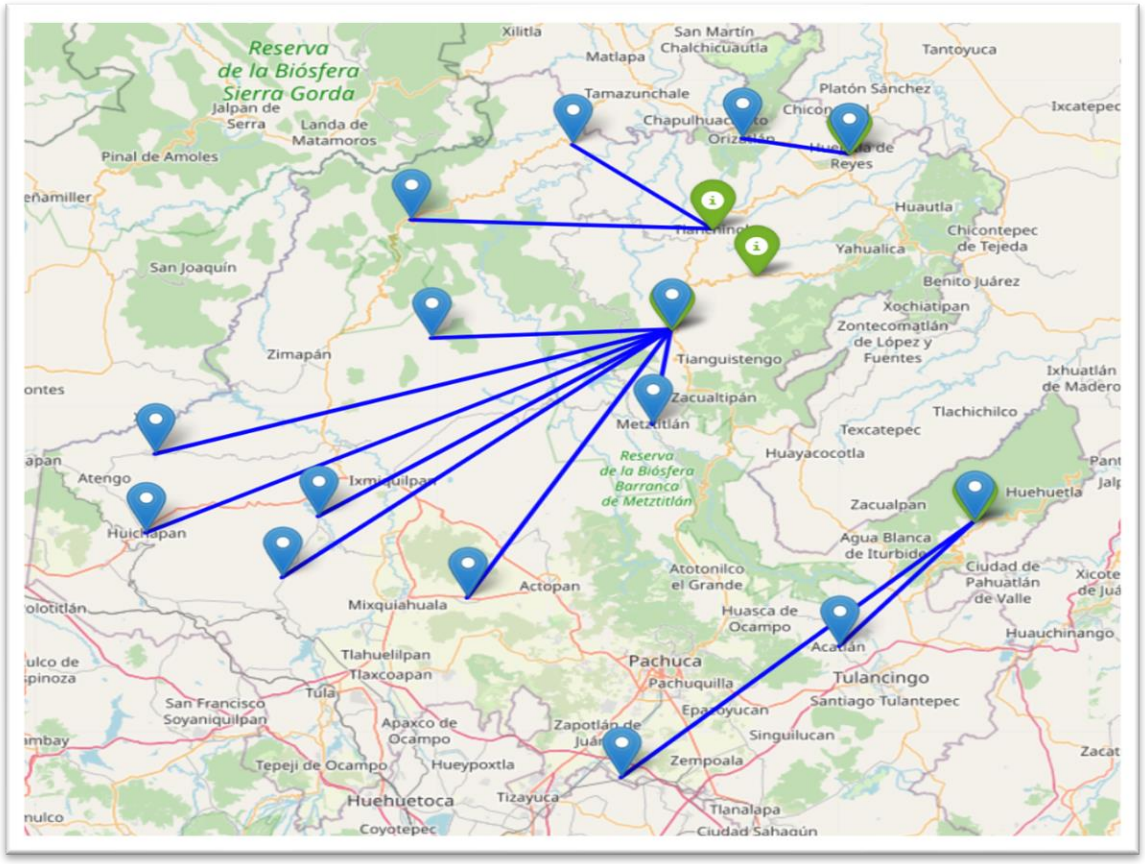
Grafica de resultados del algoritmo genético para determinar los centros



Para lo que se observa que se tiene una distancia total de **737.6052504804638 km** para todos los municipios y que cómo lo muestra el mapa y los resultados el municipio que en esta caso no se ocupa sería el del centro de gravedad y que el que tiende a distribuir mejor los productos es el municipio o en este caso el rastro de Molango de Escamilla incluso teniendo una mucha mayor distribución que los rastros de los otros municipios, teniendo en cuenta que todos los centros tienen la misma cantidad de producción, con un costo total para todos de **33011248.950000007 cabezas de ganado** por lo cual el rastro elegido por el algoritmo sería Molango de Escamilla.

Figura 25

Mapa de resultados del algoritmo genético para determinar el rastro



4.36 Problema TSP de ida y regreso por un municipio para determinar la localización del rastro.

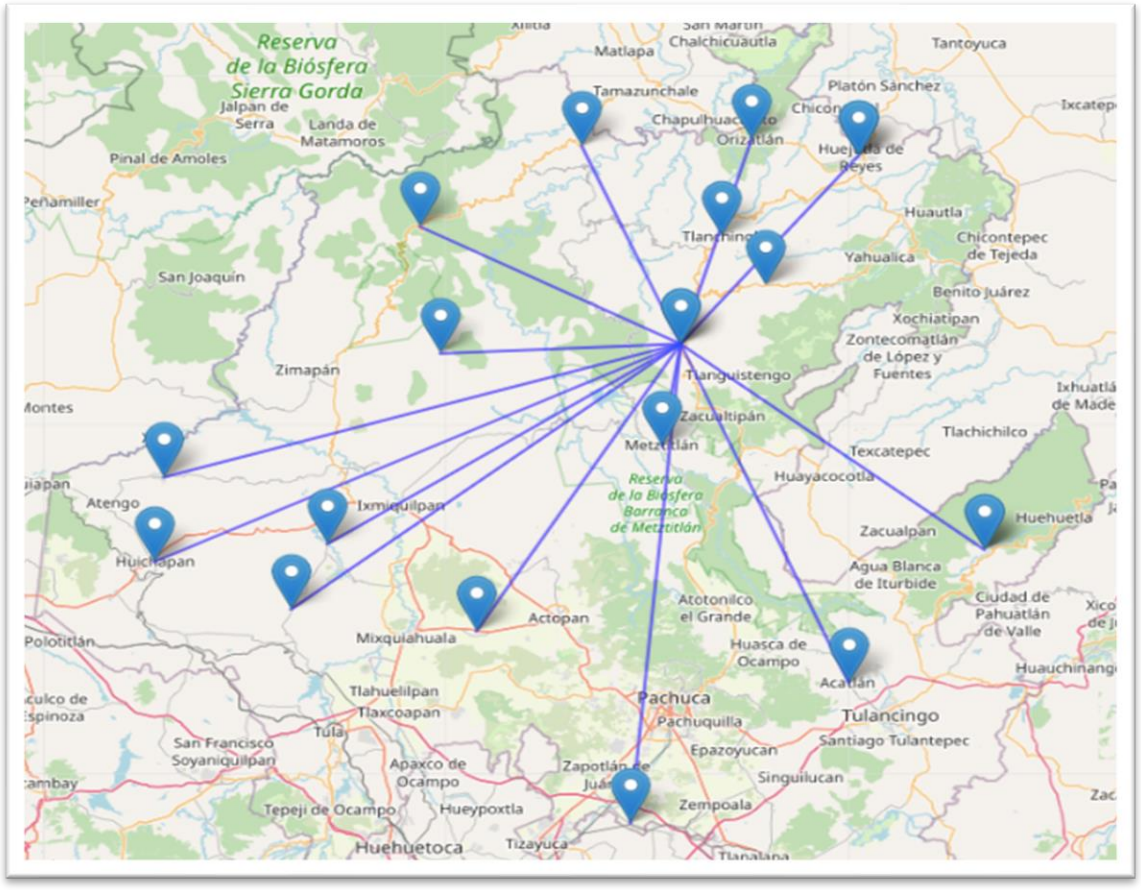
Ahora se repite el proceso para determinar la localización de los centros, se calcula la distancia total simulando que un transporte fuera de un municipio al rastro y después fuera otro, así sucesivamente hasta terminar con todos los municipios dando como resultado las distancias totales de estos municipios::

Instalación	Distancia total de la ruta (km)
San Bartolo Tutotepec	2826
Huejutla de Reyes	2760
Calnali	2176
Tlanchinol	2188
Molango de Escamilla	**1886**
Mejor instalación	
Molango de Escamilla	**1886**

Resultado que concuerda con el del algoritmo evolutivo, dando como resultado este mapa:

Figura 26

Mapa de resultados del algoritmo TSP uno a uno para determinar el rastro



4.37 Problema TSP tradicional para la localización del rastro

Ahora cómo se prosiguió en el problema anterior se da la variación original del TSP para la localización, dónde se obtiene que muchos municipios tienen la misma distancia cómo se puede mostrar:

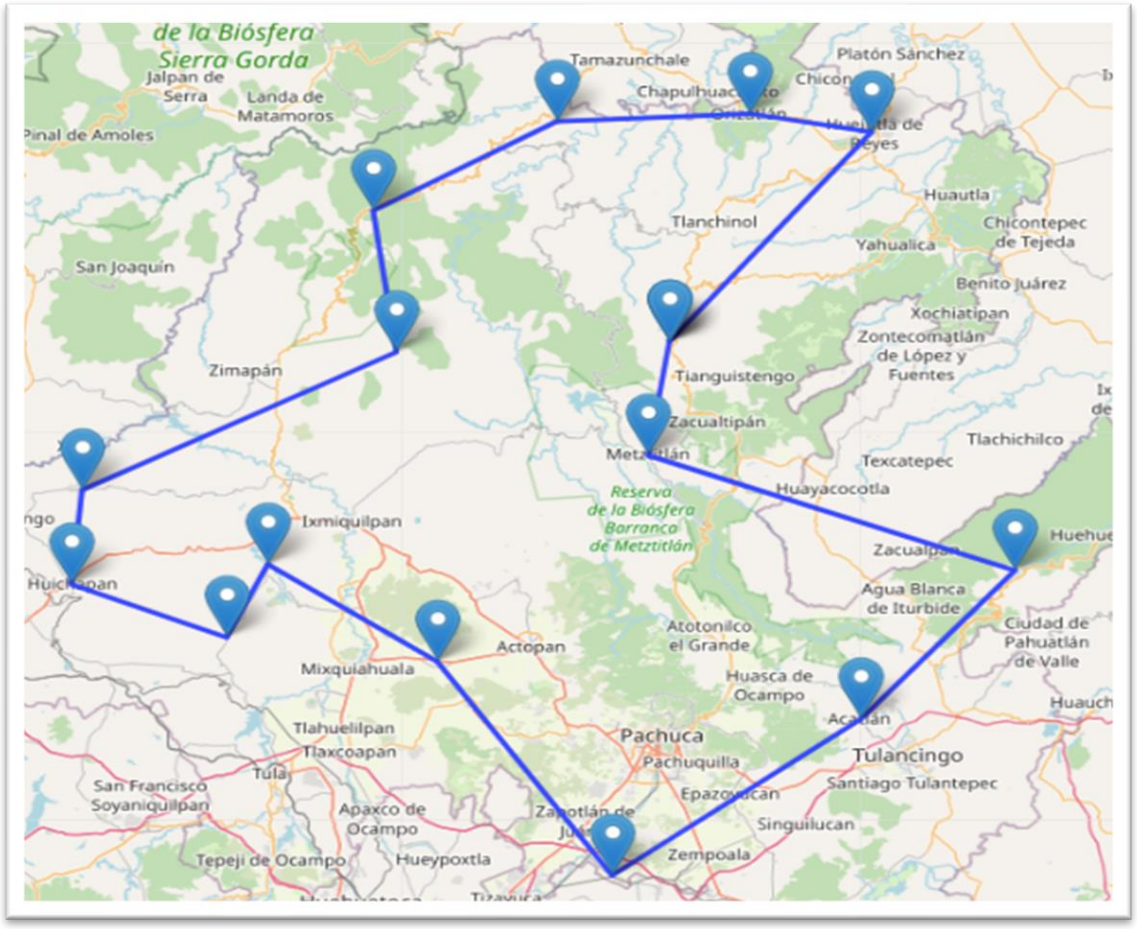
Instalación	Distancia total de la ruta (km)
San Bartolo Tutotepec	**523**
Huejutla de Reyes	**523**
Molango de Escamilla	**523**
Calnali	**523**
Tlanchinol	525

Dónde las mejores ubicaciones serían:

Mejor instalación: San Bartolo Tutotepec, Huejutla de Reyes, Molango de Escamilla, Calnali Distancia total de la ruta: 523 km

Esto debido a que al tomar en cuenta la distancia mínima tomando ciertos municipios que se comparten uno al otro, generarían tal resultado, por ello es importante abordar el problema de varios lados, ya que la distancia óptima para varios de esos municipios quedaría de la siguiente manera:

Figura 27
Mapa de resultados del algoritmo TSP normal para determinar el rastro



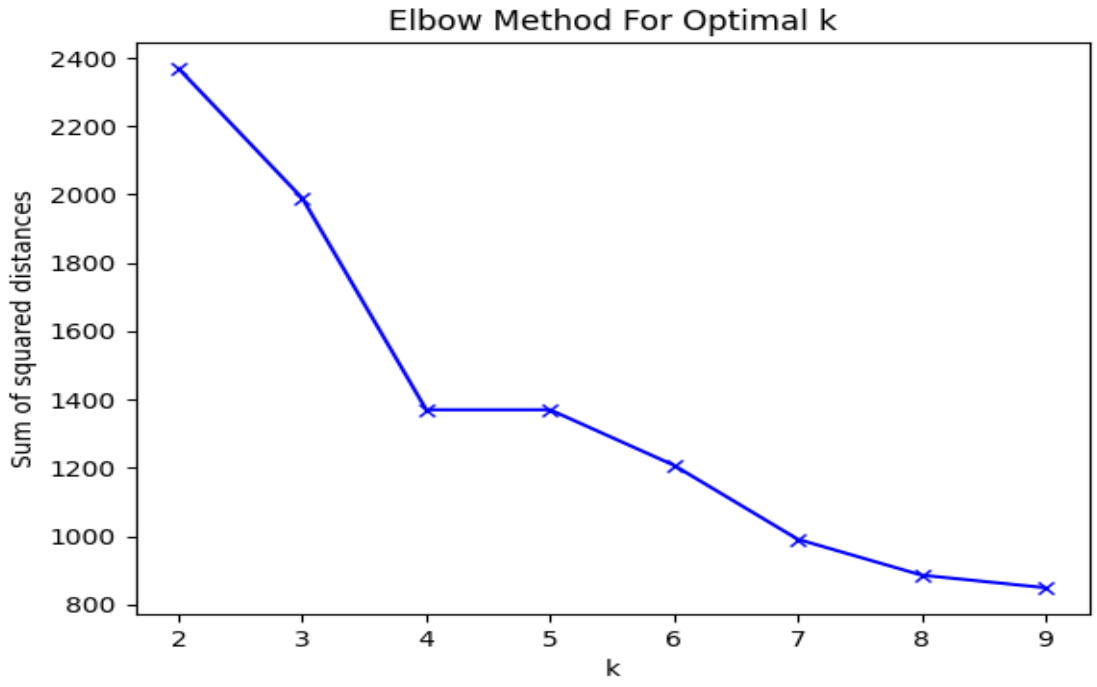
4.38 Problema VRP para la localización del rastro

Cómo última instancia se determinará la ubicación óptima del rastro ocupando un problema VRP modificado a clustering, dónde en este caso se tomará 4 clústeres, esto debido a las distancias de los centros, de tal modo que se simulará estas 4 rutas óptimas para pasar por los centros seleccionados por el algoritmo, dónde se observan estos resultados:

Instalación	Grupo 1 (km)	Grupo 2 (km)	Grupo 3 (km)	Distancia Total (km)
San Bartolo Tutotepec	74.831	247.506	327.288	451.621
Huejutla de Reyes	233.319	100.754	224.987	524.407
Molango de Escamilla	184.754	159.62	146.784	423.909
Calnali	190.489	144.599	176.845	462.305
Tlanchinol	215.138	122.353	171.792	467.544
Molango de Escamilla	**184.754**	**159.62**	**146.784**	**423.909**
		Mejor instalación		
Molango de Escamilla	**184.754**	**159.62**	**146.784**	**423.909**

Gráfica 56

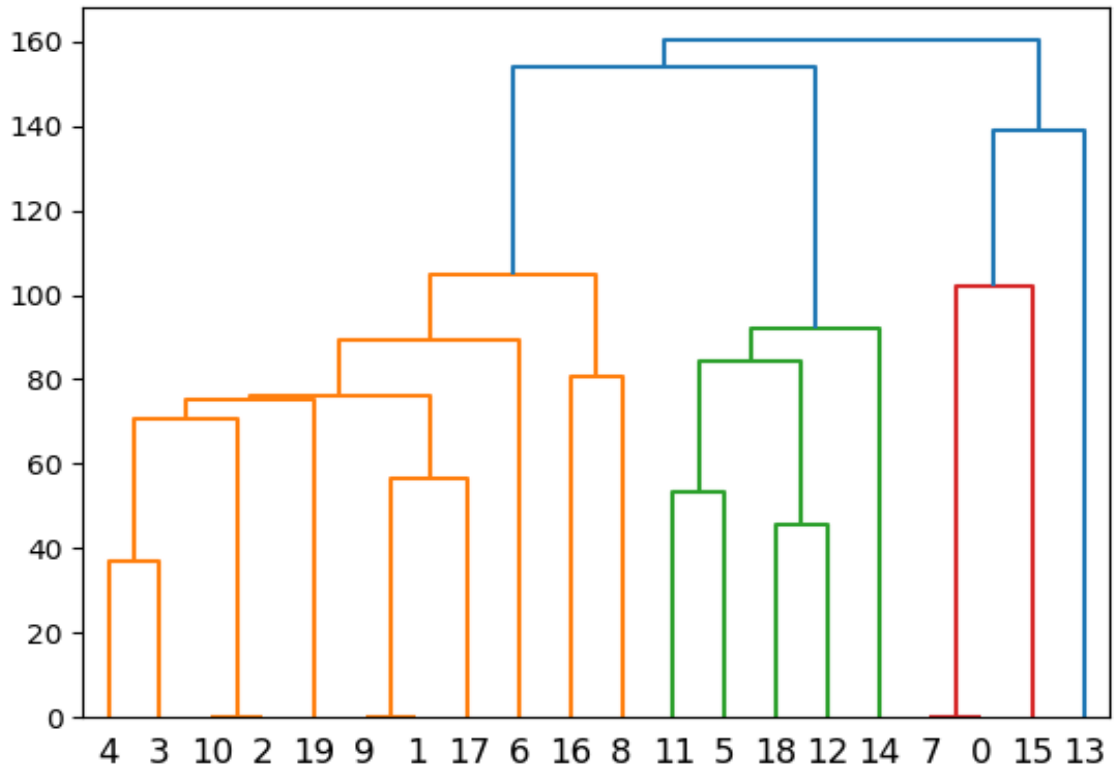
Método de Elbow para determinar el óptimo de clústeres del VRP para rastro



Nota: La k representa el número de clústeres y la parte dónde se logra visualizar la caída más pronunciada, formando el codo dictamina el número óptimo de clústeres.

Figura 28

Dendrograma para óptimo de clústeres del método VRP para el rastro



Nota: Aquí se logra ver la cantidad de visualización que se tiene de acuerdo a los municipios.

Tabla 13

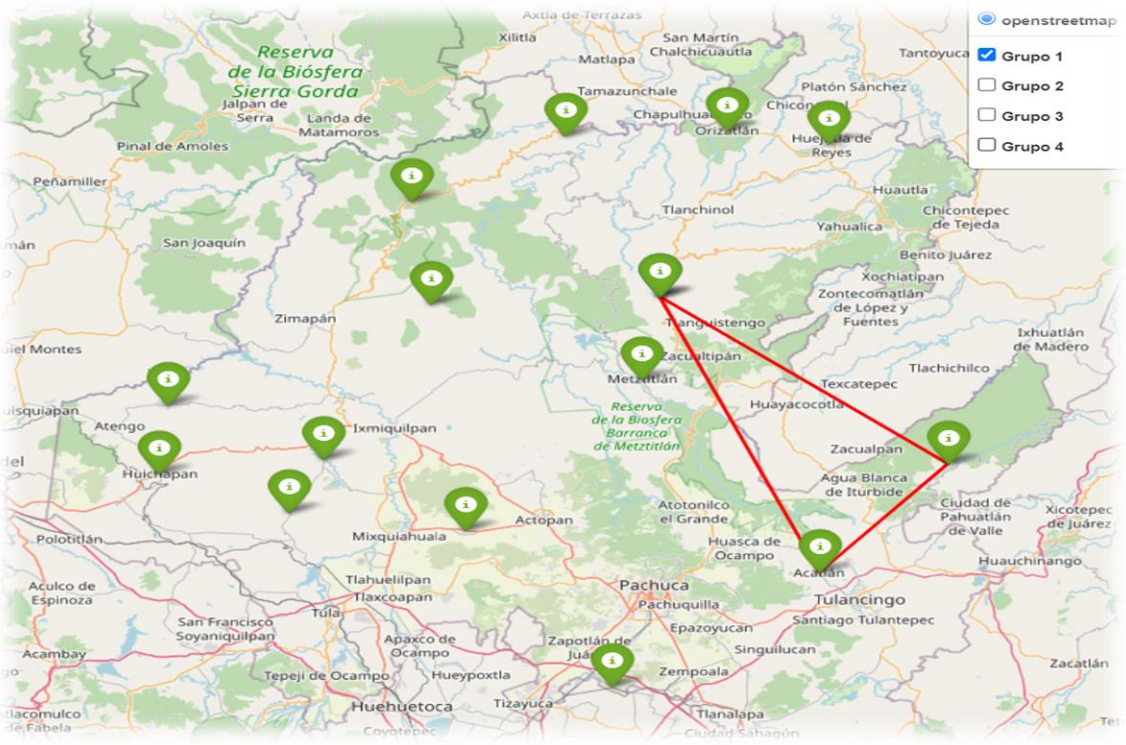
Tabla de ejemplificación de los vehículos ocupados para cada ruta del VRP para el rastro

Clúster	Municipios	Demanda Total	Vehículos Diarios con capacidad de 48 cabezas
Clúster 1	Chapantongo, Alfajayucan, Huichapan, Villa de Tezontepec, Francisco I. Madero, Tecozautla	39613.49874	3
Clúster 2	Chapulhuacán, Huejutla de Reyes, San Felipe Orizatlán	19806.74937	2
Clúster 3	San Bartolo Tutotepec, Jacala de Ledezma, Molango de Escamilla, Acatlán, Nicolás Flores, Metztitlán	39613.49874	3

Nota: esto es simplemente una ejemplificación del posible número de vehículos ocupados.

Figura 29

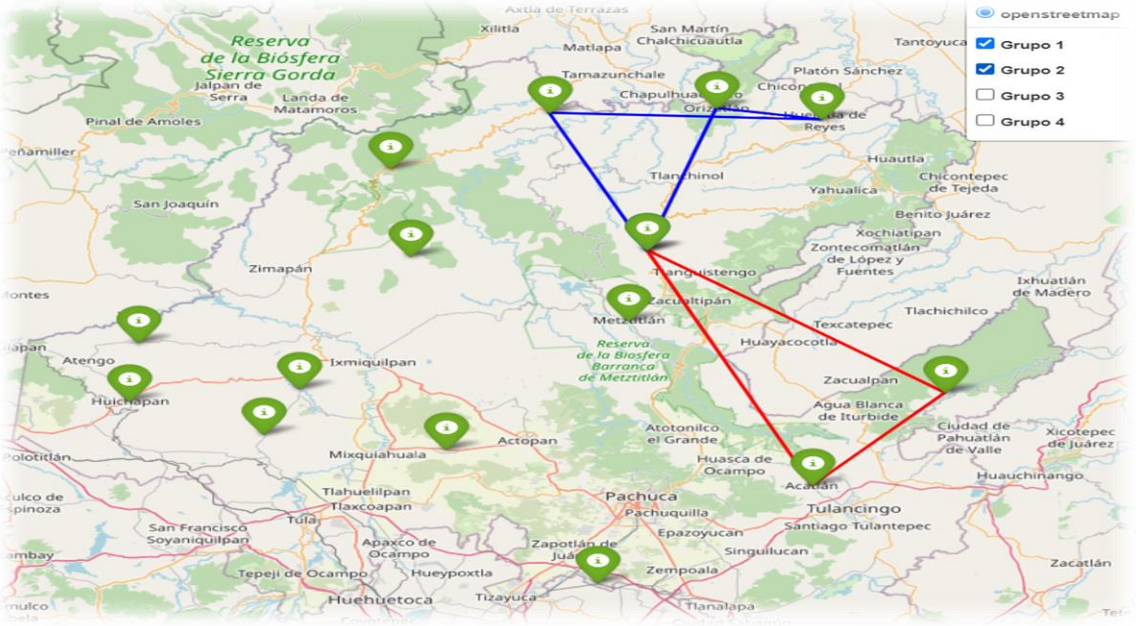
Mapa de resultados del algoritmo VRP para determinar el rastro 1



Nota: El mapa muestra una ruta de un grupo.

Figura 30

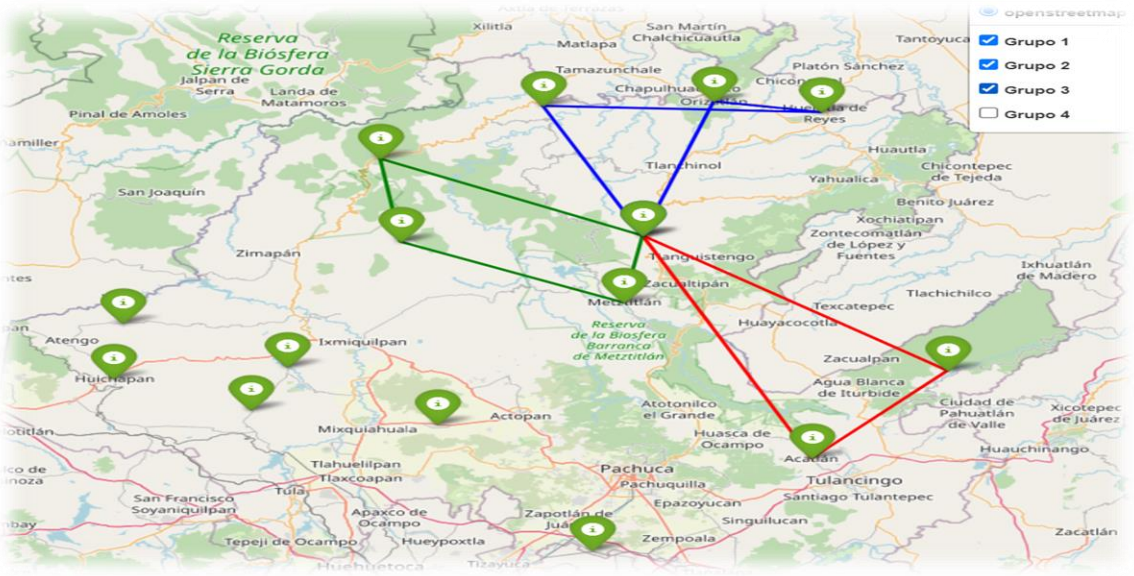
Mapa de resultados del algoritmo VRP para determinar el rastro 2



Nota: El mapa muestra dos rutas de 2 de los grupos.

Figura 31

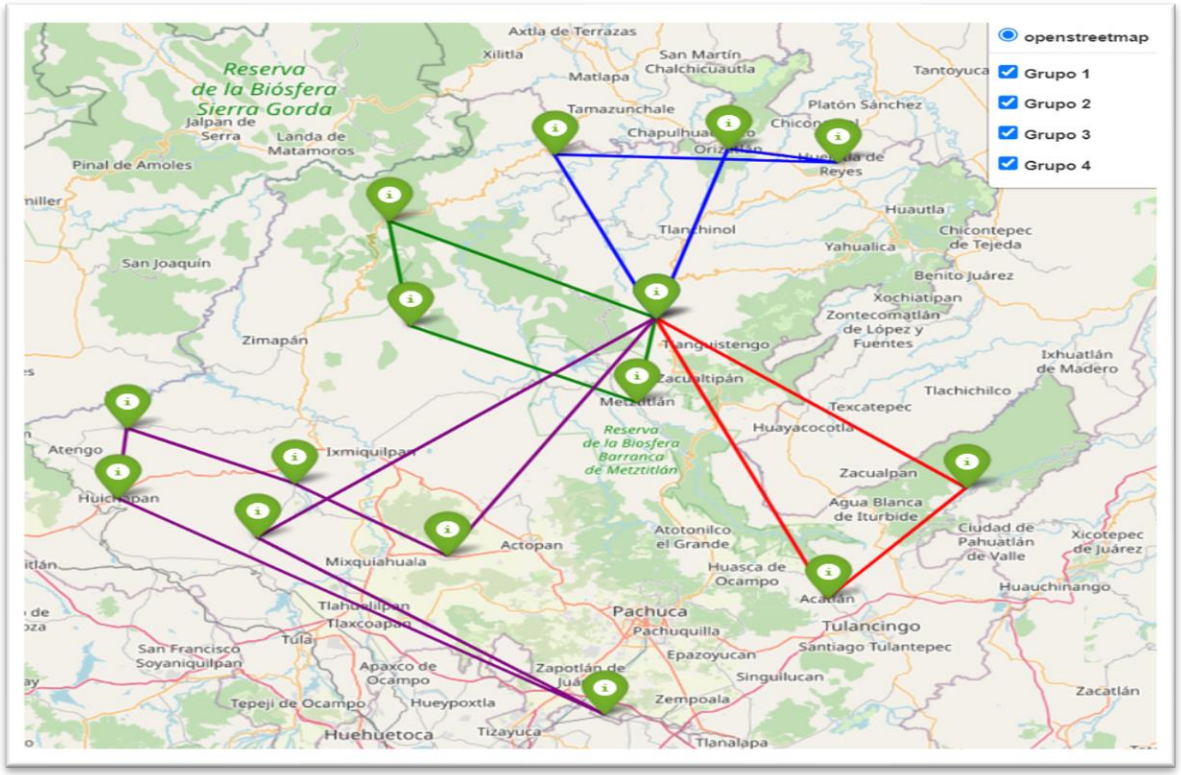
Mapa de resultados del algoritmo VRP para determinar el rastro 3



Nota: El mapa muestra tres rutas de 3 grupos.

Figura 32

Mapa de resultados del algoritmo VRP para determinar el rastro 4



Nota: El mapa muestra las cuatro rutas de los 4 grupos.

Dónde se observa de igual manera que la mejor ubicación sería **Molango de Escamilla**.

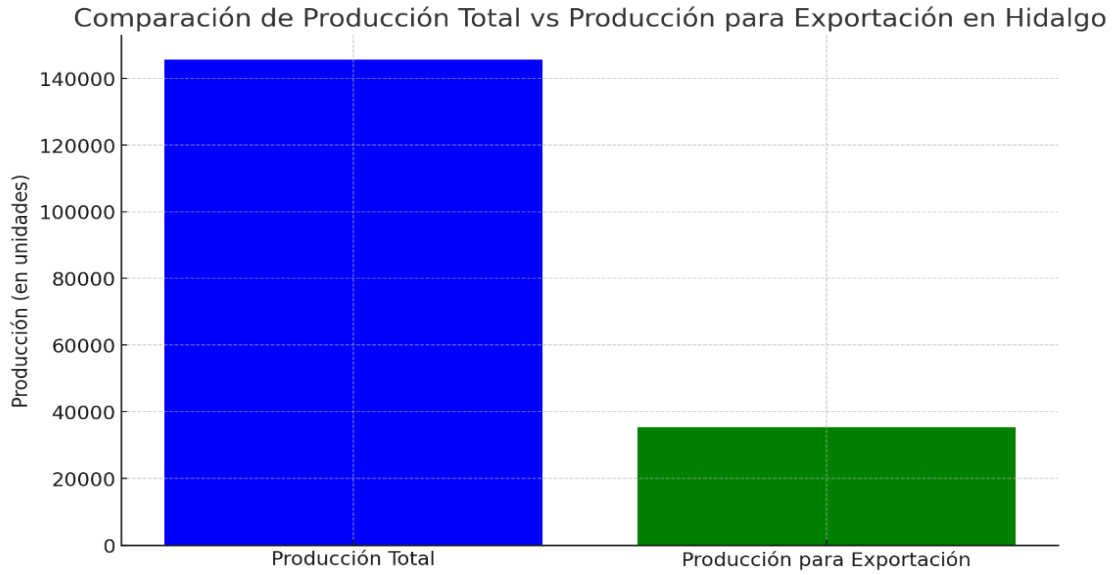
Lugar óptimo para la localización del rastro.

Teniendo en cuenta que la proyección realizada dado el pronóstico de las series de tiempo considerando el promedio de la producción de cabezas de ganado bovino de 2023 a 2030, y dado el análisis de las posibles exportaciones, importaciones y producción entendemos que la Estimación de la Demanda Basada en Producción Proyectada es de:

- Producción total para todos los municipios: 145634.39 cabezas de ganado.
- Producción para municipios exportadores: 35367.33 cabezas de ganado.
- Porcentaje estimado destinado a exportación: aproximadamente 24.29%.

Gráfica 57

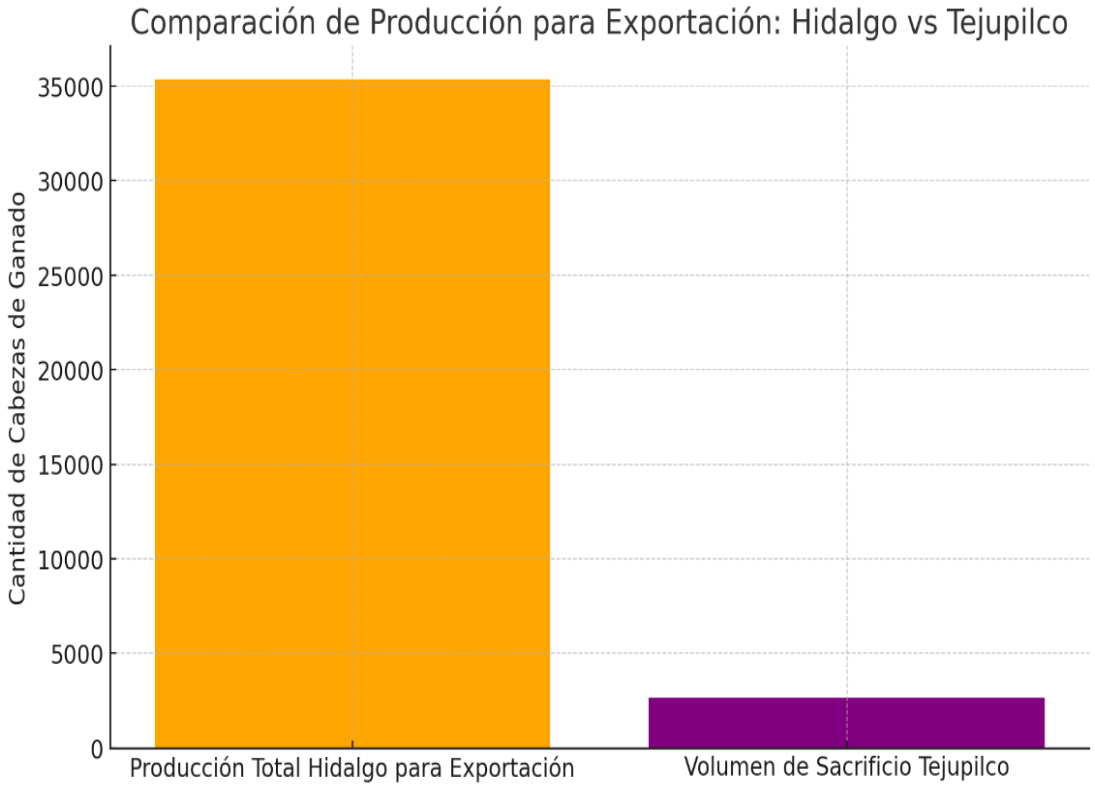
Comparación de Producción Total vs Producción para Exportación en Hidalgo



Con estas cifras, la evaluación de la vialidad de un rastro de tipo inspección federal TIF en Hidalgo es bastante buena, comparándola por ejemplo a una operación que comprende el funcionamiento de un rastro de tipo inspección federal en el municipio TEJUPILCO, ESTADO DE MÉXICO, que está considerado en el estudio mostrado por Daniel Jaramillo, Samuel Rebollar y Jesús González, en el análisis de "Modelo base para la estimación de ingresos costos e indicadores de rentabilidad del rastro privado Lodo Prieto Tejupilco Estado de México octubre de 2019" del artículo "ANÁLISIS POST INVERSIÓN DE UN RASTRO PRIVADO DE BOVINOS Y PORCINOS EN TEJUPILCO, ESTADO DE MÉXICO".

Gráfica 58

Comparación de Producción para Exportación: Hidalgo vs Tejupilco



La gráfica compara la producción de Hidalgo destinada a exportación con el volumen de sacrificio en Tejupilco. Muestra que la producción de Hidalgo para exportación (35,367 cabezas de ganado) es sustancialmente mayor que el volumen de sacrificio anual en Tejupilco (3,957 cabezas de ganado) (Jaramillo et al., 2020). Este contraste refuerza la idea de que un rastro TIF en Hidalgo, enfocado en la exportación, podría tener una operación mucho más grande y potencialmente más rentable que el caso de Tejupilco, dada la alta proporción de producción destinada a mercados internacionales. Por lo tanto la creación de un rastro TIF en Hidalgo es bastante viable.

CAPÍTULO 5 CONCLUSIONES

5.1 Conclusiones relativas a objetivos

- Existe evidencia suficiente para afirmar que el análisis de la producción histórica de ganado bovino en Hidalgo de 2006 a 2021 permite identificar tendencias y patrones de comportamiento (Objetivo 1).
- Las técnicas de series de tiempo aplicadas generan proyecciones confiables de la producción de ganado en la región hasta 2030, constituyendo una base para estimar capacidades futuras (Objetivo 2).
- La estimación de capacidad exportadora actual y futura en cada municipio, con base en los datos de producción, es una métrica adecuada para determinar el potencial productivo (Objetivo 3).
- El modelado estadístico demuestra correlaciones significativas entre variables como producción, precio promedio y población (Objetivo 4).
- Los algoritmos de optimización logran minimizar efectivamente los costos totales de transporte y distribución, cumpliendo con las restricciones planteadas (Objetivo 6).
- El análisis comparativo de escenarios permite identificar como ubicación óptima para el rastro TIF el municipio de Molango de Escamilla, maximizando las ganancias netas por exportación (Objetivo 7).
- Se proponen localizaciones estratégicas para centros de acopio en diferentes municipios, con base en la demanda asignada en el modelo (Objetivo 8).
- La ubicación del rastro TIF en Molango de Escamilla resulta la más adecuada considerando su proximidad a los centros de acopio propuestos (Objetivo 9).
- El modelo desarrollado logra determinar la localización óptima para el rastro TIF en el municipio de Molango de Escamilla con base a todos los municipios del estado Hidalgo, maximizando el potencial exportador y minimizando costos de transporte (Objetivo general).

5.2 Aportaciones originales

- Integración de técnicas innovadoras como series de tiempo, modelado estadístico y optimización multiobjetivo para resolución de problemas de ubicación de instalaciones en el sector pecuario.
- Determinación informada de localización óptima considerando capacidades futuras y restricciones logísticas.
- Metodología reproducible para maximizar beneficios económicos y productivos en la región.

5.3 Límites del modelo

- Datos históricos limitados y supuestos necesarios en proyecciones futuras.
- Costos estimados con distancias euclidianas en lugar de rutas vehiculares.
- Escaso conocimiento público de planes reales de infraestructura pecuaria.

5.4 Recomendaciones para futuros estudios

- Incorporar mayor cantidad de datos históricos y utilizar APIs de mapas para rutas reales.
- Considerar factores ambientales, sociales y políticos.
- Evaluar opciones de financiamiento y realizar análisis de factibilidad detallado.
- Extender el modelo a otras regiones y casos de estudio en el sector agropecuario.

REFERENCIAS

Ferri, P., Ferri, P., & Ferri, P. (2023, 5 junio). Morena consolida su poder territorial de cara a las elecciones de 2024. El País México. https://elpais.com/mexico/elecciones-mexicanas/2023-06-05/morena-consolida-su-poder-territorial-de-cara-a-las-elecciones-de-2024.html

De Información Agroalimentaria Y Pesquera, S. (s. f.). Sistema de Información Agroalimentaria de Consulta (SIACON). gob.mx. https://www.gob.mx/siap/prensa/sistema-de-informacion-agroalimentaria-de-consulta-siacon

De Estadística Y, I. N. (s. f.). Serie histórica censal e intercensal (1990-2010). https://www.inegi.org.mx/programas/ccpv/cpvsh/#documentacion

Wayback machine. (s. f.). https://web.archive.org/web/20200727040938/http://poblacion.hidalgo.gob.mx/pdf/idh%202015_mun_hgo_web.pdf

Hidalgo, R. A. (s. f.). LA SIERRA y LA HUASTECA HIDALGUENSE PODRÁN OBTENER RECONOCIMIENTO . . . gob.mx. https://www.gob.mx/agricultura%7Chidalgo/es/articulos/la-sierra-y-la-huasteca-hidalguense-podran-obtener-reconocimiento-internacional-para-exportar-ganado-bovino-a-estados-unidos

Greene, W. H. (2017). Econometric analysis.

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied linear statistical models with student CD. McGraw-Hill/Irwin.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). Introduction to linear regression analysis. John Wiley & Sons.

Davey, A., & Savla, J. (2009). Statistical power analysis with missing data: A structural equation modeling approach. Routledge.

Cruz, V.M., Pérez, A.J., Domínguez, H.D., & Navarro, M.A. (2018). Síntesis VLSI de un multiplicador de punto flotante de precisión simple.

Hair, J. F. (2010). Multivariate data analysis: A global perspective. Prentice Hall.

Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). Introduction to time series analysis and forecasting. John Wiley & Sons.

Wickham, H. (2016). GGPlot2: Elegant graphics for data analysis. Springer.

Vanderplas, J. T., & VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media.

Wilke, C. O. (2019). Fundamentals of data visualization: A primer on making informative and compelling figures.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. Journal of the American Statistical Association, 74(366a), 427-431. https://doi.org/10.1080/01621459.1979.10482531

Brockwell, P. J., & Davis, R. A. (2013). Introduction to time series and forecasting. Springer Science & Business Media.

Slocum, T. A. (2005). Thematic cartography and geographic visualization. Prentice Hall.

Galety, M. G., Natarajan, A. K., & Sriharsha, A. V. (2023). Advanced applications of Python data structures and algorithms. IGI Global.

Box, G.E.P., & Cox, D.R. (1964). An analysis of transformations. Journal of the Royal Statistical Society B, 26, 211-252.

Sklearn.preprocessing.StandardScaler. (n.d.). scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Kuhn, M., & Johnson, K. (2021). Feature engineering and selection: A practical approach for predictive models. CRC Press.

Cai, Y., Li, Z., Jiang, H., Zhang, H., Wang, C., Chen, E., ... & Liu, J. (2023). An adaptive stacking regressor with a self-iterative optimization module for improving fractional woody cover mapping. IEEE Geoscience and Remote Sensing Letters, 20, 1-5. https://doi.org/10.1109/LGRS.2023.3281646

Comp.ai.neural-nets FAQ, Part 3 of 7: GeneralizationSection - What are cross-validation and bootstrapping? (n.d.). http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-12.html

Hastie, T., Tibshirani, R., & Friedman, J. (2013). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media.

Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression: The x-random case. Machine learning, 19(1), 5-23.

Kohavi, R. (1995). A study on cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI (Vol. 14, No. 2, pp. 1137–1145).

Rao, R., Fung, G., & Rosales, R. (2008). On the dangers of cross-validation. An experimental evaluation.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013b). An introduction to statistical learning: With applications in R. Springer Science & Business Media.

Google. (2023, June 6). Bayesian inference - Optimize Help - Google Help. https://support.google.com/optimize/answer/9988285?hl=es

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Minitab Blog Editor. (n.d.). Regression analysis: How can I interpret R-squared and assess the goodness-of-fit? https://blog.minitab.com/en/adventures-in-statistics/regression-analysis-how-to-interpret-r-squared-and-assess-the-goodness-of-fit

Breusch, T. S., & Pagan, A. R. (1979). A simple test for heteroskedasticity and random coefficient variation. Econometrica, 47(5), 1287-1294. https://doi.org/10.2307/1910122

Koenker, R. (1981). A note on studentizing a test for heteroskedasticity. Journal of Econometrics, 17(1), 107-112. https://doi.org/10.1016/0304-4076(81)90062-2

Greene, W. H. (2002). Econometric analysis (5th ed.). Prentice Hall.

Breiman, L. (1984). Classification and regression trees. Wadsworth.

Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140. https://doi.org/10.1007/BF00058655

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. PeerJ Computer Science, 3, e127. https://doi.org/10.7717/peerj-cs.127

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in Neurorobotics, 7, 21. https://doi.org/10.3389/fnbot.2013.00021

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301-320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Glantz, S., & Slinker, B. (2000). Primer of applied regression & analysis of variance (2nd ed.). McGraw-Hill.

Montesinos López, O.A., Montesinos López, A., & Crossa, J. (2022). Overfitting, model tuning, and evaluation of prediction performance. In Multivariate statistical machine

learning methods for genomic prediction (pp. 75-92). Springer. https://doi.org/10.1007/978-3-030-89010-0_4

Abirami, K., & Mayilvahanan, P. (2016). Performance analysis of K-Means and bisecting K-Means algorithms on weblog data. Journal of Computer Applications, 119(8), 1-6.

Di, J., & Gou, X. (2018). K-means clustering algorithm based on clustering center optimization and self-determination of K values. IEEE Access, 6, 62764-62771. https://doi.org/10.1109/ACCESS.2018.2875191

Nielsen, M. A. (2016). Introduction to HPC with MPI for data science. Springer.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

Ketchen Jr, D. J., & Shook, C. L. (1996). The application of cluster analysis in strategic management research: An analysis and critique. Strategic management journal, 17(6), 441-458. https://doi.org/10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. The Computer Journal, 7(4), 308-313. https://doi.org/10.1093/comjnl/7.4.308

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197. https://doi.org/10.1109/4235.996017

Mock, W.B.T. (2011). Pareto optimality. In D.K. Chatterjee (Ed.), Encyclopedia of global justice (pp. 903-904). Springer. https://doi.org/10.1007/978-1-4020-9160-5_341

Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. Statistics and Computing, 4(2), 87-112. https://doi.org/10.1007/BF00175355

Cicirello, V. A., & Smith, S. F. (2000, July). Modeling GA performance for control parameter optimization. In Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512) (Vol. 2, pp. 1135-1142). IEEE. https://doi.org/10.1109/CEC.2000.870288

Opciones de enrutamiento. (n.d.). Google for Developers. https://developers.google.com/optimization/routing/routing_options?hl=es-419

Problema con el vendedor en viajes. (n.d.). Google for Developers. https://developers.google.com/optimization/routing/tsp?hl=es-419

Problema con el enrutamiento del vehículo. (n.d.). Google for Developers. https://developers.google.com/optimization/routing/vrp?hl=es-419

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. Expert systems with applications, 36(2), 3336-3341. https://doi.org/10.1016/j.eswa.2008.01.039

Maranzana, F. E. (1963). On the location of supply points to minimize transportation costs. IBM Systems Journal, 2(2), 129-135. https://doi.org/10.1147/sj.22.0129

De Agricultura y Desarrollo Rural, S. (n.d.). Establecimientos tipo inspección federal garantía de cárnicos de excelente calidad garantizan calidad e inocuidad de la carne. Gobierno de México. https://www.gob.mx/agricultura/articulos/establecimientos-tipo-inspeccion-federal-garantia-de-carnicos-de-excelente-calidad

Hidalgo, R. A. (n.d.-a). Hidalgo; polo de desarrollo pecuario y de producción de carne ovina. Gobierno de México. https://www.gob.mx/agricultura%7Chidalgo/articulos/hidalgo-polo-de-desarrollo-pecuario-y-de-produccion-de-carne-ovina-carmen-dorantes

Chopra, S., & Meindl, P. (2016). Supply chain management: Strategy, planning, and operation (6th ed.). Pearson.

Daskin, M. S. (2013). Network and discrete location: Models, algorithms, and applications (2nd ed.). Wiley.

Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. Wiley.

Instituto Nacional de Investigaciones Forestales, Agrícolas y Pecuarias (INIFAP). (2017). Calidad de la carne de res. https://www.gob.mx/cms/uploads/attachment/file/226086/Calidad_de_la_carne_de_res.pdf

Hernández, J. A., & García, J. A. (2017). Prácticas de manejo para mejorar la productividad de la producción de carne de res. Revista Mexicana de Ciencias Pecuarias, 8(1), 1-14.

Sanz, J. Á. (2015). Problemas de localización y distribución: Modelos de optimización y algoritmos.

García, G. A. (2017). Modelo multiobjetivo de asignación sostenible de usos del suelo [Tesis de maestría, Universidad Autónoma de Nuevo León]. Repositorio institucional UANL. http://eprints.uanl.mx/14194/

Olivares, H. S. (2013). Gestión del sistema de distribución en la logística portuaria [Tesis doctoral, Universitat Politècnica de Catalunya]. TDX. http://www.tdx.cat/handle/10803/129191

Balcázar, M. I., Agila, R. D., & Burgos, J. E. (2018). Costos de producción: Estimación y proyección de ingresos.

Corengia, R., Weigel, E. P., Villalba, D., García, D. L., & Sak, L. S. (2019). Desarrollo de corredores en el Mercosur-Chile y perspectivas del transporte intermodal.

Coronel, R. F. (2013). Diseño del modelo scor en un operador logístico, aplicado a los procesos de almacenamiento, recolección y despacho de productos perecibles, para mejorar la eficacia de la gestión de la cadena de suministro y mejorar el nivel de servicio al cliente [Tesis de maestría, Universidad de Chile]. Repositorio académico UCHILE. http://repositorio.uchile.cl/handle/2250/115010

Márquez, M. (2000). El Fondo de Estabilización de Precios del Petróleo (FEPP) y el mercado de los derivados en Chile [Tesis de magíster, Universidad de Chile]. Repositorio académico UCHILE. http://repositorio.uchile.cl/handle/2250/106403

Galeano, E., & Montoya, V. (2008). Optimización multiobjetivo de la operación en sistemas automatizados de distribución de energía eléctrica. https://repositorio.utp.edu.co/handle/11059/837

Herrnán, L. (2008). Modelo de optimización multiobjetivo para la programación de la producción agrícola en Santander, Colombia. http://tangara.uis.edu.co/biblioweb/tesis/2018/173926.pdf

Todman, L. C., Coleman, K., Milne, A. E., Gil, J. D. B., Reidsma, P., Schwoob, M. H., Treyer, S., & Whitmore, A. P. (2019). Multi-objective optimization as a tool to identify possibilities for future agricultural landscapes. The Science of the Total Environment, 687, 535-545. https://doi.org/10.1016/j.scitotenv.2019.06.070

Ge, H., Goetz, S. J., Canning, P., Perez, A., & Liang, H. (2018). Optimal locations of fresh produce aggregation facilities in the United States with scale economies. International Journal of Production Economics, 197, 143-157. https://doi.org/10.1016/j.ijpe.2017.12.020

Martín-Hernández, E., Hu, Y., Martín, M., Zavala, V. M., & Ruiz-Mercado, G. (2020). Optimal selection and location of nutrient recovery systems considering standalone and coordinated strategies. Water Research, 170, 115366. https://doi.org/10.1016/j.watres.2019.115366

Hodrick, R. J., & Prescott, E. S. (1981). Post-War U.S. Business Cycles: An Empirical investigation. RePEc: Research Papers in Economics. https://econpapers.repec.org/RePEc:nwu:cmsems:451

De Riesgo Compartido, F. (n.d.). ¿Conoces el proceso del ganado dentro de un Rastro TIF? gob.mx. https://www.gob.mx/firco/articulos/conoces-el-proceso-del-ganado-dentro-de-un-rastro-tif?idiom=es

Mavani, N. R., Ali, J. M., Othman, S., Hussain, M. A., Hashim, H., & Rahman, N. A. (2022). Application of Artificial Intelligence in Food Industry—a Guideline. *Food Engineering Reviews*, *14*(1), 134–175. https://doi.org/10.1007/s12393-021-09290-z

Gupta, R., & Nanda, S. J. (2022). Solving time varying many-objective TSP with dynamic θ-NSGA-III algorithm. Applied Soft Computing, 118, 108493. https://doi.org/10.1016/j.asoc.2022.108493

Wang, Y., Shi, L., Cai, Z., Fu, L., & Jin, X. (2020). NSGA-II algorithm and application for multi-objective flexible workshop scheduling. Journal of Algorithms & Computational Technology, 14, 174830262094246. https://doi.org/10.1177/1748302620942467

Los hechos y solo los hechos: la inteligencia artificial. (n.d.) https://www.hpe.com/mx/es/what-is/artificial-intelligence.html

Lange, R. T., Schaul, T., Chen, Y., Lu, C. X., Zahavy, T., Dalibard, V., & Flennerhag, S. (2023). Discovering Attention-Based genetic Algorithms via Meta-Black-Box optimization. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2304.03995

Jaramillo-Puebla et al. (2020b). ANÁLISIS POST INVERSIÓN DE UN RASTRO PRIVADO DE BOVINOS y PORCINOS EN TEJUPILCO, ESTADO DE MÉXICO. https://www.redalyc.org/journal/141/14165939002/

ANEXO A: CÓDIGO 1

Código del desarrollo del modelo de la predicción de las cabezas de ganado bovino

```
# Carga el archivo .xlsx
df = pd.read_excel('/content/drive/MyDrive/val.xlsx')

# Define los datos a usar
columnas_produccion = ['Variable 1', 'Variable 2', 'Variable 5', 'Variable 6', 'Variable 9 (Objetivo)', 'Variable 10', 'Variable 11']
data = df[columnas_produccion]

columnas_totales = ['Municipio', 'Año', 'Variable 1', 'Variable 2', 'Variable 5', 'Variable 6', 'Variable 9 (Objetivo)', 'Variable 10', 'Variable 11']
datos = df[columnas_totales]

# Limpia los datos
data = data.replace(' ', '', regex=True)
data = data.replace(',', '.', regex=True)

data = data.astype(float)

# Define la cantidad de municipios de entrenamiento
num_municipios_entrenamiento = 80

# Define municipios aleatorios
municipios_unicos = datos['Municipio'].unique()
np.random.shuffle(municipios_unicos)

# Define los municipios de entrenamiento y prueba
municipios_entrenamiento = municipios_unicos[:num_municipios_entrenamiento]
municipios_prueba = municipios_unicos[num_municipios_entrenamiento:]

train = datos[datos['Municipio'].isin(municipios_entrenamiento)]
```

```
test = datos[datos['Municipio'].isin(municipios_prueba)]

data_train = data.loc[train.index]
data_test = data.loc[test.index]

# Define la clase para la transformación Box-Cox
class BoxCoxTransformer(BaseEstimator, TransformerMixin):
    def __init__(self, features, lmbda=0.15):
        self.features = features
        self.lmbda = lmbda

    def fit(self, X, y=None):
        return self

    def boxcox1p(self, x, lmbda):
        if lmbda != 0:
            return (x + 1)**lmbda - 1
        else:
            return np.log(x + 1)

    def transform(self, X, y=None):
        X = X.copy()
        for feature in self.features:
            X.loc[:, feature] = self.boxcox1p(X[feature], self.lmbda)
        return X


# Procesamiento de la Data
categorical_features = ['Municipio']
categorical_transformer      =      OneHotEncoder(handle_unknown='ignore',
sparse_output=False)

polynomial_transformer = PolynomialFeatures(degree=1)

preprocessor = ColumnTransformer(
```

```
    transformers=[
        ('cat', categorical_transformer, categorical_features)],
    remainder=StandardScaler())

# Aplica tranformación Box-Cox
num_features = ['Variable 1', 'Variable 2', 'Variable 5', 'Variable 6',
'Variable 10', 'Variable 11']
boxcox_transformer = BoxCoxTransformer(features=num_features)

# Selecciona las mejoras características
k = 7
feature_selector = SelectKBest(f_regression, k=k)

# Inicia el modelo
model1 = GradientBoostingRegressor()
model2 = RandomForestRegressor()
model3 = ElasticNet(max_iter=50000, l1_ratio=1)
model4 = XGBRegressor()
bagging_model     =     BaggingRegressor(estimator=DecisionTreeRegressor(),
n_estimators=10)

# Stackea multiples modelos
stacked_models   =   StackingRegressor(estimators=[('gb',   model1),   ('rf',
model2), ('en', model3), ('xgb', model4), ('bagging', bagging_model)])

# Define el pipeline
pipe = Pipeline(steps=[('boxcox_transformer', boxcox_transformer),
                       ('preprocessor', preprocessor),
                       ('polynomial_transformer', polynomial_transformer),
                       ('feature_selector', feature_selector),
                       ('model', stacked_models)])

# Ajusta los hiperparámetros del modelo
parameters = {
    'model__gb__max_depth': [1, 2, 3, 4],
```

```
    'model__gb__min_samples_split': [2, 3, 5, 7, 10],
    'model__rf__n_estimators': [10, 30, 50, 70, 100, 150, 200],
    'model__rf__max_features': ['sqrt', 'log2', None],
    'model__en__alpha': np.logspace(-5, 0, num=20),
    'model__en__l1_ratio': np.linspace(0, 1, num=20),
    'model__xgb__learning_rate': np.logspace(-3, 0, num=10),
    'model__xgb__max_depth': [3, 5, 7, 9],
    'model__xgb__n_estimators': [50, 100, 150, 200],
    'model__xgb__min_child_weight': [1, 3, 5],
    'model__xgb__gamma': [0, 0.1, 0.2],
    'model__xgb__subsample': [0.5, 0.7, 0.9],
    'model__xgb__colsample_bytree': [0.5, 0.7, 0.9]
}


group_kfold = GroupKFold(n_splits=10)

# Rendimiento de búsqueda bayesiana con cross validation
search = BayesSearchCV(pipe, parameters, n_iter=50,
scoring='neg_mean_absolute_error', cv=group_kfold, n_jobs=-1)
search.fit(train.drop('Variable 9 (Objetivo)', axis=1), train['Variable 9
(Objetivo)'], groups=train['Municipio'])

print("Best parameters found: ", search.best_params_)
print("Lowest MAE found: ", np.abs(search.best_score_))

pipe.set_params(**search.best_params_)
pipe.fit(train.drop('Variable 9 (Objetivo)', axis=1), train['Variable 9
(Objetivo)'])

predictions = pipe.predict(test.drop('Variable 9 (Objetivo)', axis=1))
mae = mean_absolute_error(test['Variable 9 (Objetivo)'], predictions)
print("MAE: ", mae)

# Hacer predicciones para cada municipio
```

```
predictions = []
for municipio in municipios_unicos:
    municipio_data = datos[datos['Municipio'] == municipio]
    municipio_data = municipio_data[columnas_totales]

    x_test = []
    for i in range(7, len(municipio_data)):
        x_test.append(municipio_data.iloc[i-7:i, :])
    x_test = pd.concat(x_test)

    if x_test.shape[0] > 0:
        predicted_values     =     pipe.predict(x_test.drop('Variable     9
(Objetivo)', axis=1))

        for i in range(min(len(predicted_values), 25)):
            year = 2006 + i
            value = predicted_values[i]
            predictions.append([municipio, year, value])
    else:
        print(f"No  hay  suficientes  datos  para  hacer  una  predicción  para
{municipio}.")

# Convierte las predicciones en un DataFrame
predictions_df = pd.DataFrame(predictions)


predictions_df  =  pd.DataFrame(predictions,  columns=['Municipio',  'Año',
'Valor'])
predictions_df.to_excel('predicciones.xlsx', index=False)
```

ANEXO B: CÓDIGO 2

Código de algoritmo evolutivo

```
from deap import base, creator, tools, algorithms
import random
import numpy as np
from math import radians, sin, cos, sqrt, atan2
import folium
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# datos
instalaciones = [
    {"Municipio": "Chapantongo", "Latitud": 20.28559483, "Longitud": -99.41319277, "Producción": 6602.24979},
    {"Municipio": "Chapulhuacán", "Latitud": 21.15787458, "Longitud": -98.90435814, "Producción": 6602.24979},
    {"Municipio": "San Bartolo Tutotepec", "Latitud": 20.3991586, "Longitud": -98.20205941, "Producción": 6602.24979},
    {"Municipio": "Jacala de Ledezma", "Latitud": 21.00849043, "Longitud": -99.18853709, "Producción": 6602.24979},
    {"Municipio": "Huejutla de Reyes", "Latitud": 21.13959058, "Longitud": -98.42044493, "Producción": 6602.24979},
    {"Municipio": "Molango de Escamilla", "Latitud": 20.78658031, "Longitud": -98.7306035, "Producción": 6602.24979},
    {"Municipio": "Alfajayucan", "Latitud": 20.41015828, "Longitud": -99.34946458, "Producción": 6602.24979},
    {"Municipio": "Huichapan", "Latitud": 20.37553905, "Longitud": -99.6510293, "Producción": 6602.24979},
    {"Municipio": "Villa de Tezontepec", "Latitud": 19.879986, "Longitud": -98.81930736, "Producción": 6602.24979},
    {"Municipio": "Francisco I. Madero", "Latitud": 20.24540323, "Longitud": -99.08881409, "Producción": 6602.24979},
```

```
    {"Municipio": "Acatlán", "Latitud": 20.14599753, "Longitud": -98.43835281, "Producción": 6602.24979},
    {"Municipio": "Nicolás Flores", "Latitud": 20.76804674, "Longitud": -99.15056418, "Producción": 6602.24979},
    {"Municipio": "San Felipe Orizatlán", "Latitud": 21.17106016, "Longitud": -98.60758905, "Producción": 6602.24979},
    {"Municipio": "Tecozautla", "Latitud": 20.53415978, "Longitud": -99.6349425, "Producción": 6602.24979},
    {"Municipio": "Metztitlán", "Latitud": 20.5948676, "Longitud": -98.7642084, "Producción": 6602.24979}
]

clientes = [
    {"Municipio": "Actopan", "Latitud": 20.26918743, "Longitud": -98.94286206, "Demanda": 1339.69499},
    {"Municipio": "Apan", "Latitud": 19.71003178, "Longitud": -98.45236722, "Demanda": 2471.989578},
    {"Municipio": "Atitalaquia", "Latitud": 20.05929791, "Longitud": -99.22112947, "Demanda": 557.1169201},
    {"Municipio": "Atotonilco de Tula", "Latitud": 20.00039212, "Longitud": -99.21807773, "Demanda": 2713.440279},
    {"Municipio": "Emiliano Zapata", "Latitud": 19.65736929, "Longitud": -98.54730232, "Demanda": 449.384075},
    {"Municipio": "Epazoyucan", "Latitud": 20.01811094, "Longitud": -98.63604826, "Demanda": 602.1081288},
    {"Municipio": "Francisco I. Madero", "Latitud": 20.24540323, "Longitud": -99.08881409, "Demanda": 1237.102658},
    {"Municipio": "Huejutla de Reyes", "Latitud": 21.13959058, "Longitud": -98.42044493, "Demanda": 4979.113861},
    {"Municipio": "Ixmiquilpan", "Latitud": 20.4874612, "Longitud": -99.21584985, "Demanda": 2838.446607},
    {"Municipio": "Mineral de La Reforma", "Latitud": 20.07243478, "Longitud": -98.69588487, "Demanda": 15063.60747},
    {"Municipio": "Mineral del Monte", "Latitud": 20.14055695, "Longitud": -98.67147452, "Demanda": 408.9543905},
```

```
    {"Municipio": "Mixquiahuala de Juárez", "Latitud": 20.22971536, "Longitud": -99.21397966, "Demanda": 1416.818176},
    {"Municipio": "Pachuca de Soto", "Latitud": 20.12699971, "Longitud": -98.73005493, "Demanda": 22317.94682},
    {"Municipio": "Progreso de Obregón", "Latitud": 20.24868014, "Longitud": -99.18942359, "Demanda": 1158.828115},
    {"Municipio": "San Agustín Tlaxiaca", "Latitud": 20.1157998, "Longitud": -98.88675497, "Demanda": 1756.706179},
    {"Municipio": "San Salvador", "Latitud": 20.28495895, "Longitud": -99.0154511, "Demanda": 1244.233287},
    {"Municipio": "Santiago Tulantepec de Lugo Guerrero", "Latitud": 20.04050747, "Longitud": -98.35736774, "Demanda": 1837.631258},
    {"Municipio": "Tepeapulco", "Latitud": 19.7858894, "Longitud": -98.5530995, "Demanda": 2769.985565},
    {"Municipio": "Tepeji del Río de Ocampo", "Latitud": 19.90548931, "Longitud": -99.34182441, "Demanda": 2968.837365},
    {"Municipio": "Tezontepec de Aldama", "Latitud": 20.19289391, "Longitud": -99.27271678, "Demanda": 1921.076482},
    {"Municipio": "Tizayuca", "Latitud": 19.84126223, "Longitud": -98.98151696, "Demanda": 7102.832774},
    {"Municipio": "Tlahuelilpan", "Latitud": 20.13082664, "Longitud": -99.23456837, "Demanda": 389.8091821},
    {"Municipio": "Tlanalapa", "Latitud": 19.81786009, "Longitud": -98.60380072, "Demanda": 278.6898165},
    {"Municipio": "Tlaxcoapan", "Latitud": 20.09155632, "Longitud": -99.2213648, "Demanda": 353.5982297},
    {"Municipio": "Tolcayuca", "Latitud": 19.95673828, "Longitud": -98.9216934, "Demanda": 847.0972813},
    {"Municipio": "Tula de Allende", "Latitud": 20.07145808, "Longitud": -99.34480352, "Demanda": 4899.476703},
    {"Municipio": "Tulancingo de Bravo", "Latitud": 20.10754951, "Longitud": -98.38196463, "Demanda": 9277.88531},
    {"Municipio": "Zacualtipán de Ángeles", "Latitud": 20.64546431, "Longitud": -98.6535594, "Demanda": 2064.969542},
```

```
    {"Municipio": "Zapotlán de Juárez", "Latitud": 19.97414785, "Longitud":
-98.8619187, "Demanda": 695.7076297},
    {"Municipio":  "Zempoala",  "Latitud":  19.91562419,  "Longitud":  -
98.66811157, "Demanda": 3070.658113}
]

# Constantes
COST_MULTIPLIER = 1000

# Crear el tipo de fitness y el tipo de individuo
creator.create("FitnessMulti", base.Fitness, weights=(-1.0, -1.0))
creator.create("Individual", list, fitness=creator.FitnessMulti)

# Definición de las funciones
def haversine(lat1, lon1, lat2, lon2):
    R = 6371.0  # radio de la Tierra en km
    dlon = radians(lon2) - radians(lon1)
    dlat = radians(lat2) - radians(lat1)
    a = sin(dlat / 2)**2 + cos(radians(lat1)) * cos(radians(lat2)) * sin(dlon
/ 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    return R * c

def calcular_aptitud(individual):
    distancia_total,         demanda_total,         costo_total         =
calcular_distancia_demanda_costo(individual)
    return distancia_total, costo_total

def calcular_distancia_demanda_costo(individual):
    distancia_total = 0
    demanda_total = [0] * len(instalaciones)
    capacidad_total  =  [instalacion["Producción"]  for  instalacion  in
instalaciones]
    costo_total = 0
    for i, cliente in enumerate(clientes):
```

```
        instalacion_seleccionada = instalaciones[individual[i] %
len(instalaciones)]
        distancia_total += haversine(instalacion_seleccionada["Latitud"],
                                     instalacion_seleccionada["Longitud"],
                                     cliente["Latitud"],
                                     cliente["Longitud"])
        demanda_total[individual[i] % len(instalaciones)] +=
cliente["Demanda"]

    for i in range(len(instalaciones)):
        if demanda_total[i] > capacidad_total[i]:
            costo_total += (demanda_total[i] - capacidad_total[i]) *
COST_MULTIPLIER
    return distancia_total, demanda_total, costo_total

def proximidad_geo():
    X = [[cliente["Latitud"], cliente["Longitud"]] for cliente in clientes]
    kmeans = KMeans(n_clusters=len(instalaciones), n_init=10).fit(X)
    individual = kmeans.labels_
    return individual.tolist()

def crear_individuo():
    return proximidad_geo()

def custom_mutate(individual, indpb):
    for i in range(len(individual)):
        if random.random() < indpb:
            individual[i] = random.randint(0, len(instalaciones) - 1)
    return individual,

def custom_mate(ind1, ind2):
    for i in range(len(ind1)):
        if abs(instalaciones[ind1[i]]["Latitud"] -
instalaciones[ind2[i]]["Latitud"]) < 0.05 and
```

```
abs(instalaciones[ind1[i]]["Longitud"]                                    -
instalaciones[ind2[i]]["Longitud"]) < 0.05:
            if                  clientes[i]["Demanda"]                  <=
instalaciones[ind2[i]]['Producción']    and    clientes[i]["Demanda"]    <=
instalaciones[ind1[i]]['Producción']:
                ind1[i], ind2[i] = ind2[i], ind1[i]
    return ind1, ind2
toolbox = base.Toolbox()
toolbox.register("individual",    tools.initIterate,    creator.Individual,
crear_individuo)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
toolbox.register("mate", custom_mate)
toolbox.register("mutate", custom_mutate, indpb=0.05)
toolbox.register("select", tools.selNSGA2)
toolbox.register("evaluate", calcular_aptitud)
tamanio_poblacion = 500
probabilidad_cruce = 0.9
probabilidad_mutacion = 0.01
generaciones = 1000

def init_population():
    return toolbox.population(n=tamanio_poblacion)

def evaluate_population(poblacion):
    fitnesses = list(map(toolbox.evaluate, poblacion))
    for ind, fit in zip(poblacion, fitnesses):
        ind.fitness.values = fit
    return poblacion

def perform_evolution(poblacion):
    hof = tools.ParetoFront()
    stats = tools.Statistics(lambda ind: ind.fitness.values)
    stats.register("avg", np.mean, axis=0)
    stats.register("std", np.std, axis=0)
    stats.register("min", np.min, axis=0)
```

```
    stats.register("max", np.max, axis=0)
    logbook = tools.Logbook()
    logbook.header = "gen", "avg", "std", "min", "max"

    # Evaluar la población inicial
    poblacion = evaluate_population(poblacion)

    # Actualizar la logbook y el Frente de Pareto con la población inicial
    record = stats.compile(poblacion)
    logbook.record(gen=0, **record)
    hof.update(poblacion)

    # Comenzar la evolución
    for gen in range(1, generaciones + 1):
        offspring = algorithms.varOr(poblacion, toolbox,
lambda_=tamanio_poblacion, cxpb=probabilidad_cruce,
mutpb=probabilidad_mutacion)

        # Evaluar los individuos
        invalid_ind = [ind for ind in offspring if not ind.fitness.valid]
        fitnesses = map(toolbox.evaluate, invalid_ind)
        for ind, fit in zip(invalid_ind, fitnesses):
            ind.fitness.values = fit

        # Actualizar la población
        poblacion[:] = toolbox.select(offspring + poblacion,
k=tamanio_poblacion)

        # Actualizar la logbook y el Frente de Pareto
        hof.update(poblacion)
        record = stats.compile(poblacion)
        logbook.record(gen=gen, **record)

    return poblacion, hof, logbook
```

```
def main():
    poblacion = init_population()
    poblacion, hof, logbook = perform_evolution(poblacion)

    # Imprimir el mejor individuo encontrado
    mejor_ind = tools.selBest(poblacion, 1)[0]
    print("Mejor individuo es ", mejor_ind, " con aptitud: ",
mejor_ind.fitness.values[0])

    # Imprimir la mejor solución encontrada
    print("La mejor solución es:")
    for i, cliente in enumerate(clientes):
        print("Cliente", i, "asignado a la instalación", mejor_ind[i])

    # Mostrar las rutas de la solución
    for i, cliente in enumerate(clientes):
        print(f"Cliente {cliente['Municipio']} atendido por
{instalaciones[mejor_ind[i]]['Municipio']}")

 # Grafica la evolución
    gen, avg, min_, max_ = logbook.select("gen", "avg", "min", "max")
    avg_dist = [a[0] for a in avg]
    avg_cost = [a[1] for a in avg]
    max_dist = [m[0] for m in max_]
    max_cost = [m[1] for m in max_]
    plt.figure()
    plt.semilogy(gen, avg_dist, label="Promedio Distancia")
    plt.semilogy(gen, avg_cost, label="Promedio Costo")
    plt.semilogy(gen, max_dist, label="Máxima Distancia")
    plt.semilogy(gen, max_cost, label="Máximo Costo")
    plt.xlabel("Generación")
    plt.ylabel("Valor")
    plt.legend()
    plt.show()
```

```
    return poblacion, logbook, hof
if __name__ == "__main__":
    main()

def visualizar_resultados(mejor_ind):

   # Crear un mapa centrado en la ubicación media de todas las instalaciones
y clientes
    lat_media = sum(inst["Latitud"] for inst in instalaciones + clientes) /
len(instalaciones + clientes)
    lon_media = sum(inst["Longitud"] for inst in instalaciones + clientes)
/ len(instalaciones + clientes)
    mapa = folium.Map(location=[lat_media, lon_media], zoom_start=6)

   # Agregar marcadores para las instalaciones
    for i, inst in enumerate(instalaciones):
        folium.Marker([inst["Latitud"],                 inst["Longitud"]],
popup=f"Instalación {i}", icon=folium.Icon(color="green")).add_to(mapa)

 # Agregar marcadores para los clientes y líneas para representar las
asignaciones de clientes a instalaciones
    for i, cliente in enumerate(clientes):
        folium.Marker([cliente["Latitud"],           cliente["Longitud"]],
popup=f"Cliente {i}").add_to(mapa)
        folium.PolyLine([(cliente["Latitud"],        cliente["Longitud"]),
(instalaciones[mejor_ind[i]]["Latitud"],
instalaciones[mejor_ind[i]]["Longitud"])], color="blue").add_to(mapa)

    # Mostrar el mapa
    return mapa
if __name__ == "__main__":
    poblacion, logbook, hof = main()
    mejor_ind = hof[0]
    mapa = visualizar_resultados(mejor_ind)
    display(mapa)
```

ANEXO C: CÓDIGO 3

Código de para resolver el problema TSP y VRP

```
from ortools.constraint_solver import routing_enums_pb2
from ortools.constraint_solver import pywrapcp
from math import radians, cos, sin, asin, sqrt
import numpy as np
import folium
instalaciones = [
    {"Municipio": "Chapantongo", "Latitud": 20.28559483, "Longitud": -
99.41319277},
    {"Municipio": "Chapulhuacán", "Latitud": 21.15787458, "Longitud": -
98.90435814},
    {"Municipio": "San Bartolo Tutotepec", "Latitud": 20.3991586,
"Longitud": -98.20205941},
    {"Municipio": "Jacala de Ledezma", "Latitud": 21.00849043, "Longitud":
-99.18853709},
    {"Municipio": "Huejutla de Reyes", "Latitud": 21.13959058, "Longitud":
-98.42044493},
    {"Municipio": "Molango de Escamilla", "Latitud": 20.78658031,
"Longitud": -98.7306035},
    {"Municipio": "Alfajayucan", "Latitud": 20.41015828, "Longitud": -
99.34946458},
    {"Municipio": "Huichapan", "Latitud": 20.37553905, "Longitud": -
99.6510293},
    {"Municipio": "Villa de Tezontepec", "Latitud": 19.879986, "Longitud":
-98.81930736},
    {"Municipio": "Francisco I. Madero", "Latitud": 20.24540323, "Longitud":
-99.08881409},
    {"Municipio": "Acatlán", "Latitud": 20.14599753, "Longitud": -
98.43835281},
    {"Municipio": "Nicolás Flores", "Latitud": 20.76804674, "Longitud": -
99.15056418},
```

```
    {"Municipio": "San Felipe Orizatlán", "Latitud": 21.17106016, "Longitud": -98.60758905},
    {"Municipio": "Tecozautla", "Latitud": 20.53415978, "Longitud": -99.6349425},
    {"Municipio": "Metztitlán", "Latitud": 20.5948676, "Longitud": -98.7642084}
]
clientes = [
    {"Municipio": "Actopan", "Latitud": 20.26918743, "Longitud": -98.94286206},
    {"Municipio": "Apan", "Latitud": 19.71003178, "Longitud": -98.45236722},
    {"Municipio": "Atitalaquia", "Latitud": 20.05929791, "Longitud": -99.22112947},
    {"Municipio": "Atotonilco de Tula", "Latitud": 20.00039212, "Longitud": -99.21807773},
    {"Municipio": "Emiliano Zapata", "Latitud": 19.65736929, "Longitud": -98.54730232},
    {"Municipio": "Epazoyucan", "Latitud": 20.01811094, "Longitud": -98.63604826},
    {"Municipio": "Francisco I. Madero", "Latitud": 20.24540323, "Longitud": -99.08881409},
    {"Municipio": "Huejutla de Reyes", "Latitud": 21.13959058, "Longitud": -98.42044493},
    {"Municipio": "Ixmiquilpan", "Latitud": 20.4874612, "Longitud": -99.21584985},
    {"Municipio": "Mineral de La Reforma", "Latitud": 20.07243478, "Longitud": -98.69588487},
    {"Municipio": "Mineral del Monte", "Latitud": 20.14055695, "Longitud": -98.67147452},
    {"Municipio": "Mixquiahuala de Juárez", "Latitud": 20.22971536, "Longitud": -99.21397966},
    {"Municipio": "Pachuca de Soto", "Latitud": 20.12699971, "Longitud": -98.73005493},
```

```
    {"Municipio": "Progreso de Obregón", "Latitud": 20.24868014, "Longitud": -99.18942359},
    {"Municipio": "San Agustín Tlaxiaca", "Latitud": 20.1157998, "Longitud": -98.88675497},
    {"Municipio": "San Salvador", "Latitud": 20.28495895, "Longitud": -99.0154511},
    {"Municipio": "Santiago Tulantepec de Lugo Guerrero", "Latitud": 20.04050747, "Longitud": -98.35736774},
    {"Municipio": "Tepeapulco", "Latitud": 19.7858894, "Longitud": -98.5530995},
    {"Municipio": "Tepeji del Río de Ocampo", "Latitud": 19.90548931, "Longitud": -99.34182441},
    {"Municipio": "Tezontepec de Aldama", "Latitud": 20.19289391, "Longitud": -99.27271678},
    {"Municipio": "Tizayuca", "Latitud": 19.84126223, "Longitud": -98.98151696},
    {"Municipio": "Tlahuelilpan", "Latitud": 20.13082664, "Longitud": -99.23456837},
    {"Municipio": "Tlanalapa", "Latitud": 19.81786009, "Longitud": -98.60380072},
    {"Municipio": "Tlaxcoapan", "Latitud": 20.09155632, "Longitud": -99.2213648},
    {"Municipio": "Tolcayuca", "Latitud": 19.95673828, "Longitud": -98.9216934},
    {"Municipio": "Tula de Allende", "Latitud": 20.07145808, "Longitud": -99.34480352},
    {"Municipio": "Tulancingo de Bravo", "Latitud": 20.10754951, "Longitud": -98.38196463},
    {"Municipio": "Zacualtipán de Ángeles", "Latitud": 20.64546431, "Longitud": -98.6535594},
    {"Municipio": "Zapotlán de Juárez", "Latitud": 19.97414785, "Longitud": -98.8619187},
    {"Municipio": "Zempoala", "Latitud": 19.91562419, "Longitud": -98.66811157}
]
```

```
# Funciones para crear los datos del problema.
def haversine(loc1, loc2):
    lon1, lat1 = loc1["Longitud"], loc1["Latitud"]
    lon2, lat2 = loc2["Longitud"], loc2["Latitud"]
    # Convertir grados a radianes.
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
    # Fórmula de haversine.
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    r = 6371 # Radio de la tierra en kilómetros.
    return c * r
def create_data_model(locations):
    data = {}
    data['distance_matrix']    =    [[0]*len(locations)    for    _    in
range(len(locations))]
    for i in range(len(locations)):
        for j in range(len(locations)):
            data['distance_matrix'][i][j]   =   int(haversine(locations[i],
locations[j]))
    return data
def main():
    best_instalacion = None  # Variable para almacenar la mejor instalación
    min_distance = float('inf')   # Inicializar la distancia mínima como
infinito
    for i in range(len(instalaciones)):
        total_distance = 0
        for j in range(len(clientes)):
            locations = [instalaciones[i], clientes[j], instalaciones[i]]
            # Crear el modelo de datos.
            data = create_data_model(locations)
            # Crear el problema de enrutamiento.
            manager                                                    =
pywrapcp.RoutingIndexManager(len(data['distance_matrix']), 1, 0)
```

```
            routing = pywrapcp.RoutingModel(manager)
            def distance_callback(from_index, to_index):
                return
data['distance_matrix'][manager.IndexToNode(from_index)][manager.IndexToNo
de(to_index)]
            transit_callback_index                                    =
routing.RegisterTransitCallback(distance_callback)

routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)
            search_parameters = pywrapcp.DefaultRoutingSearchParameters()
            search_parameters.first_solution_strategy                 =
(routing_enums_pb2.FirstSolutionStrategy.PATH_CHEAPEST_ARC)
            # Resolver el problema.
            solution = routing.SolveWithParameters(search_parameters)
            # Calcular la distancia total de la ruta.
            if solution:
                total_distance += solution.ObjectiveValue()
        # Actualizar la mejor instalación si se encuentra una distancia
menor.
        if total_distance < min_distance:
            min_distance = total_distance
            best_instalacion = instalaciones[i]
        # Imprimir la instalación y la distancia total de la ruta.
        print(f"Instalación:   {instalaciones[i]['Municipio']},   Distancia
total de la ruta: {total_distance} km")
    if best_instalacion is not None:
        print(f"\nMejor    instalación:    {best_instalacion['Municipio']},
Distancia total de la ruta: {min_distance} km")
    else:
        print("No se encontró una mejor instalación.")
    # Crear un mapa centrado en la mejor instalación
    mapa        =        folium.Map(location=[best_instalacion["Latitud"],
best_instalacion["Longitud"]], zoom_start=10)
    # Agregar marcadores para las instalaciones
    for instalacion in instalaciones:
```

```
        distancia = haversine(best_instalacion, instalacion)
        folium.Marker(
            location=[instalacion["Latitud"], instalacion["Longitud"]],
            tooltip=f"Distancia: {distancia:.2f} km"
        ).add_to(mapa)
    # Agregar marcadores para los clientes
    for cliente in clientes:
        folium.Marker(
            location=[cliente["Latitud"], cliente["Longitud"]],
            tooltip=cliente["Municipio"]
        ).add_to(mapa)
    # Conectar la mejor instalación con los clientes utilizando líneas
    for cliente in clientes:
        folium.PolyLine(
            locations=[
                [best_instalacion["Latitud"],
best_instalacion["Longitud"]],
                [cliente["Latitud"], cliente["Longitud"]]
            ],
            color="blue",
            weight=2,
            opacity=0.7
        ).add_to(mapa)
    # Mostrar el mapa
    display(mapa)
main()
```