Sistema de análisis y predicción del crimen ‘Precrimen’

DANIEL LORENZO MEDINA SALCEDO
Autor Trabajo de Grado

Nota de Aceptación

Certificamos que el presente Trabajo de Grado Satisface, en alcances y calidad, todos los requisitos que demanda un Trabajo de Grado de Maestría.

DAVID ARANGO RODRIGUEZ
Director

VALENTINA CORCHUELO GUZMAN
Jurado

MARIO JULIAN MORA CARDONA
Jurado

Aprobado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali, para optar el título de Magister en Ciencia de Datos.

HERNÁN CAMILO ROCHA NIÑO Ph. D.
Decano Facultad de Ingeniería y Ciencias

JUAN CARLOS MARTÍNEZ ARIAS
Director Posgrados de Ingeniería y Ciencias

Santiago de Cali, 1 de agosto del 2024

Santiago de Cali, 28 de junio del 2024

Doctor

Diego Linares.

Director Maestría en Ciencia de Datos

Facultad de Ingeniería y Ciencias

Pontificia Universidad Javeriana de Cali

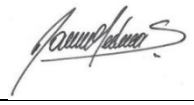
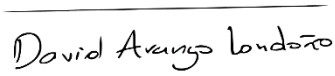
Asunto: Presentación para evaluación del proyecto aplicado

Cordial Saludo,

Con el fin de cumplir con los requisitos exigidos por la Universidad para optar por el título de Magíster en Ciencia de Datos, nos permitimos presentar a su consideración el proyecto denominado Sistema de análisis y predicción del crimen "Precrimen" el cual fue realizado por el estudiante Daniel Lorenzo Medina Salcedo pertenecientes a la Maestría en Ciencia de Datos, bajo la dirección de David Arango Londoño.

El suscrito director del Proyecto Aplicado autoriza para que se proceda a hacer la evaluación de este proyecto, toda vez que ha revisado cuidadosamente el documento y avala que ya se encuentra listo para ser presentado y sustentado oficialmente.

Atentamente,

	
Estudiante Daniel Lorenzo Medina Salcedo CC. 80.793.180 de Bogotá	Director David Arango Londoño CC. 1.130.586.950

Documentación anexa:

Resumen del Proyecto Aplicado en formato digital.

Una copia digital (PDF) del documento del proyecto aplicado

FICHA RESUMEN

PROYECTO APLICADO – MAESTRÍA EN CIENCIA DE DATOS

TÍTULO: Sistema de análisis y predicción del crimen “PRECRIMEN”

1. ÁREA DE TRABAJO: Ciencia de datos, Seguridad ciudadana
2. TIPO DE PROYECTO (Aplicado, Innovación, Investigación): Aplicado
3. ESTUDIANTE(S): Daniel Lorenzo Medina Salcedo
4. CORREO ELECTRÓNICO: medinada@javerianacali.edu.co
5. DIRECCIÓN Y TELEFONO: Km 19 Autopista Norte, Chía, Costado Occidental, 3167508302
6. DIRECTOR: David Arango Londoño
7. VINCULACIÓN DEL DIRECTOR: Profesor
8. CORREO ELECTRÓNICO DEL DIRECTOR: david.arango@javerianacali.edu.co
9. CODIRECTOR : No aplica
10. GRUPO O EMPRESA QUE LO AVALA (Si aplica): EMAP, Estadística y Matemática Aplicada
11. OTROS GRUPOS O EMPRESAS:
12. PALABRAS CLAVE (al menos 5): Crimen, Predicción, Delito, Seguridad Ciudadana, Machine Learning, Series de tiempo, Geo celdas, datos espaciales.
13. FECHA DE INICIO: 30/07/2023
14. FECHA DE FINALIZACIÓN: 27/06/2024
15. RESUMEN:

El proyecto "Sistema de análisis y predicción del crimen ‘Precrimen’" presenta una solución innovadora para abordar la criminalidad. Utilizando técnicas avanzadas de ciencia de datos se procesan datos espaciales y hechos delictivos buscando mejorar la seguridad ciudadana mediante la predicción de cantidad de eventos delictivos y la implementación de estrategias preventivas.

El objetivo principal del proyecto es construir una herramienta tecnológica que permita predecir la posible comisión de delitos. Los objetivos específicos incluyen la identificación, clasificación y visualización de datos conectando fuentes abiertas y oficiales que disponen información sobre delitos, la construcción de un modelo predictivo y la apropiación de conocimientos prácticos en gestión de datos, clasificación, visualización y modelos de predicción.

La metodología empleada en el proyecto incluye varias fases:

Recopilación y preparación de datos: Se investigaron diferentes fuentes de datos nacionales e internacionales, destacando la base de datos Departamento de Policía de Los Ángeles y de la Policía Nacional de Colombia.

Análisis exploratorio de datos: Permitió comprender la estructura de las bases de datos, identificar patrones y correlaciones, y generar hipótesis.

Feature Engineering: Se crearon nuevas características para mejorar la capacidad predictiva de los modelos incluyendo variables de tiempo y datos espaciales segmentados.

Construcción del modelo predictivo: Se realizaron varias pruebas con diferentes algoritmos de Machine Learning tales cómo redes LTSM y Arboles aleatorios de regresión.

Evaluación y ajuste del modelo: Se evalúan métrica para evaluar la precisión de las series de tiempos evaluando el modelo con el 20% de los datos.

Implementación de interfaz de visualización: Utilizando Power BI y Python, se desarrolló una interfaz para visualizar los datos y los resultados del análisis.

El análisis descriptivo reveló patrones significativos en la criminalidad, como la alta frecuencia de delitos de robo y asalto en la ciudad de Los Ángeles y un crecimiento de delitos sexuales en Bogotá, Medellín y Cali. Los modelos predictivos desarrollados demostraron ser altamente efectivos para anticipar eventos delictivos, permitiendo a las autoridades actuar proactivamente.

El proyecto concluye que la integración de técnicas de ciencia de datos en la gestión de seguridad pública es esencial para mejorar la prevención del crimen. Se recomienda ampliar el proyecto para incluir el comportamiento de los victimarios para enriquecer la capacidad predictiva. La implementación de este sistema tiene el potencial de transformar la dinámica de seguridad en Colombia y otros países en desarrollo.

SISTEMA DE ANÁLISIS Y PREDICCIÓN DEL CRIMEN “PRECRIMEN”

DANIEL LORENZO MEDINA SALCEDO

Código 8972920

Proyecto Aplicado para optar al título de

Magister en Ciencia de Datos

Director

DAVID ARANGO LONDOÑO

FACULTAD DE INGENIERÍA Y CIENCIAS

MAESTRÍA EN CIENCIA DE DATOS

SANTIAGO DE CALI, JUNIO 28 DE 2024

TABLA DE CONTENIDO

Lista de Tablas

Lista de ilustraciones

Lista de gráficas

INTRODUCCIÓN

El crimen constituye uno de los desafíos más persistentes y complejos que enfrentan las sociedades contemporáneas, afectando de manera desproporcionada a los países en vías de desarrollo [1, 2]. En este contexto, Colombia se presenta como un estudio de caso emblemático, donde los elevados índices de criminalidad no solo impactan la seguridad y bienestar de sus ciudadanos, sino que también representan un obstáculo significativo para atraer inversión extranjera y fomentar el desarrollo económico sostenible [3, 4].

Ante este panorama, la emergencia de métodos innovadores para la predicción, gestión y minimización del crimen es más que una necesidad; es una oportunidad para transformar la dinámica de seguridad y abrir nuevos caminos hacia la paz y el desarrollo sostenible [5]. La ciencia de datos, con su capacidad para analizar e interpretar grandes volúmenes de información, emerge como una herramienta poderosa en este esfuerzo. La visualización de datos, en particular a través de mapas de calor y sistemas de videovigilancia integrados, ha demostrado ser un avance significativo en la identificación de "puntos calientes" de actividad delictiva [6, 7].

Este proyecto busca ir más allá del análisis descriptivo tradicional, proponiendo el uso de técnicas avanzadas de ciencia de datos para desarrollar modelos predictivos que puedan anticipar eventos delictivos antes de que ocurran. La implementación de un enfoque multidimensional que incluye la recolección de datos a través de automatización contra fuentes abiertas, análisis descriptivo y modelos predictivos, tiene el potencial de revolucionar la forma en que las ciudades gestionan la seguridad pública.

El componente de recolección de datos se centró en la agregación de información a partir de fuentes abiertas y oficiales, proporcionando una base rica y diversa para el análisis. La analítica descriptiva permitió una comprensión más profunda de los patrones de criminalidad existentes, facilitando la identificación de áreas críticas y tendencias emergentes. Más importante aún, el núcleo del proyecto, la predicción, se basó en modelos avanzados para proyectar futuras incidencias delictivas, permitiendo a las autoridades actuar proactivamente en lugar de reactivamente.

Este enfoque no solo es innovador sino también estratégicamente vital para el desarrollo de políticas de seguridad efectivas y eficientes. Al proporcionar una herramienta tecnológica que integra la gestión de datos, clasificación, visualización y predicción, este proyecto se posiciona en la vanguardia de la lucha contra el crimen.

El compromiso con la apropiación de conocimientos prácticos en las áreas de gestión de datos, clasificación, visualización, y modelos de predicción es fundamental. Este proyecto no solo buscó abordar un problema crítico de seguridad sino también avanzar en la comprensión y aplicación de las ciencias de datos para el bien social. En última instancia, el objetivo es impactar positivamente la calidad de vida de los ciudadanos y contribuir de manera significativa a los objetivos de desarrollo sostenible de la humanidad, marcando un hito en la forma en que los países en desarrollo se enfrentan y manejan los desafíos del crimen en el siglo XXI.

1. DEFINICIÓN DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

En el mundo actual, con una alta tasa de urbanización, la seguridad pública sigue siendo una preocupación importante. Los métodos tradicionales de aplicación de la ley generalmente se enfocan en responder a incidentes criminales después que estos han ocurrido, en lugar de la prevención. Este enfoque reactivo, no solo agota los recursos, sino que también hace muy poco por aliviar el miedo y la perturbación que el crimen genera en las comunidades. Adicionalmente, con el creciente volumen y la complejidad de los datos disponibles, las instituciones encargadas de hacer cumplir la ley les resulta difícil procesar y utilizar eficientemente esta información para la prevención del delito.

La insuficiencia de las estrategias tradicionales de prevención del delito, junto con la subutilización de los avances tecnológicos y el análisis de datos presentan un problema crítico: la capacidad de anticipar y prevenir actividades delictivas de manera efectiva. Este problema se agrava en zonas con recursos limitados para hacer cumplir la ley, donde la capacidad de analizar patrones delictivos y aplicar medidas preventivas muchas veces se ve superada por la dinámica del comportamiento delictivo.

Además, la falta de conocimientos precisos y predictivos sobre los posibles focos de delincuencia obstaculiza el desarrollo de intervenciones específicas y la asignación optima de recursos policiales. La ausencia de un enfoque proactivo y basado en datos para la prevención del delito y la asignación de recursos genera ineficiencias en la gestión de la seguridad pública, lo que potencialmente aumenta el riesgo de delincuencia y reduce la calidad de vida general de los ciudadanos.

Para abordar estos desafíos, existe una necesidad apremiante de soluciones innovadoras que aprovechen la ciencia de datos y los avances tecnológicos para pasar de una estrategia de seguridad pública reactiva a una positiva. El objetivo no es solo mejorar la capacidad de los organismos encargados de hacer cumplir la ley para predecir y prevenir el delito, sino también dotar a las comunidades de información y herramientas para contribuir a su propia seguridad. Este cambio requiere el desarrollo de una herramienta tecnológica que pueda analizar grandes cantidades de datos, identificar patrones, predecir futuras actividades delictivas y simular estrategias de mitigación, fomentando en última instancia una sociedad más segura y resiliente.

1.2 FORMULACION DEL PROBLEMA

Pregunta de alto nivel

¿Usando técnicas de ciencia de datos se podrían recomendar acciones para disminuir o incluso predecir posibles hechos delictivos?

Preguntas especificas

- ¿Se podría clasificar, visualizar denuncias de posibles crímenes en una ciudad conectando fuentes abiertas y/o fuentes oficiales.?
- ¿La visualización de datos permitiría describir las zonas de mayor criminalidad?
- ¿A partir de la información colectada se podría generar un modelo predictivo que permita predecir nuevos crímenes o acciones para la mitigación de nuevos crímenes?

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

Construir una herramienta tecnológica “Precrimen”, basada en técnicas de las ciencias de datos que permita predecir la posible comisión de delitos o la simulación de acciones para mitigar el crimen.

2.2 OBJETIVO ESPECÍFICO

1. Identificar, clasificar y visualizar los datos conectando fuentes abiertas y/o fuentes oficiales que disponen información de delitos.
2. Construir un modelo de predicción y/o recomendación que permita predecir la comisión delitos.
3. Apropiar conocimientos prácticos de gestión de datos, clasificación, visualización y modelos de predicción.

3. MARCO TEÓRICO Y ANTECEDENTES

Para realizar una contextualización sobre el crimen como un fenómeno de naturaleza social, se debe primero aclarar lo que es el contexto social y dónde se origina éste; es así entonces, que el origen de dicho contexto parte de la idea que los seres humanos a diferencia de los animales, poseen una naturaleza específica e individual en su comportamiento, la cual les otorga una personalidad que les hace más manejables y flexibles en su quehacer diario en situaciones de convivencia, y esto, a través de la historia le ha facilitado la composición de agrupaciones de personas con intereses socio culturales similares lo que conllevó al establecimiento de asentamientos humanos y ha facilitado la convivencia dentro de ellos.

En este contexto, es pertinente argumentar que el ser humano es inherentemente parte de un tejido social, el cual está modelado por las relaciones derivadas de sus interacciones con otros, más que por experiencias individuales. Esto indica que, para comprender a una persona, es esencial analizar cómo interactúa y coexiste con otros dentro de un contexto social. Como señala Huertas Díaz [8], *"La estructura y la cualidad constitutiva de la dirección del comportamiento de un individuo dependen de la estructura de las relaciones entre individuos".* Esto subraya que la relación entre el individuo y la sociedad está intrínsecamente ligada a la propia sociedad y que dicha relación tiene una estructura y normativas definidas colectivamente, no individualmente. Este marco de relaciones y regulaciones establecidas grupalmente es crucial para lograr la cohesión social y asegurar la convivencia armoniosa de los individuos, ahora considerados ciudadanos, dentro de las comunidades.

Para alcanzar la cohesión social o la unidad en una comunidad, es crucial la implementación de procesos regulatorios. A lo largo de la historia, desde la formación de los primeros estados y los diversos tipos de gobiernos, estos procesos han evolucionado significativamente. Este desarrollo ha dado lugar a la creación de las Constituciones Políticas, que, según el estado colombiano, se definen como “la ley máxima y suprema de un país o estado”[9]. Estas constituciones desempeñan un papel fundamental en la regulación de la vida de los ciudadanos, estableciendo un marco de normas y leyes que guían y disciplinan sus comportamientos en diversos contextos.

Este modelo de control es de tipo socio-jurídico y las leyes que lo componen se refuerzan con la amenaza de sancionar de forma represiva al individuo por su incumplimiento, esta infracción se determina así por realizar algún tipo de acción delictiva contraria a la ley, y se les denomina cómo crímenes [10]. Y se evidencia la necesidad de un control sancionatorio a la perpetración de crímenes dado el impacto de estos a la sociedad, cómo lo plantea Gonzáles Andrade [11]: “Las tasas de criminalidad altas perjudican el clima de inversión privada y desvían los recursos públicos al

fortalecimiento del sistema policial en lugar de fomentar las actividades económicas provocando la erosión de la calidad de vida. Además de tener impactos en el corto y largo plazos sobre el desarrollo local, pues afecta los niveles de capital físico y el clima de inversión, limitando el desarrollo del capital humano, erosionando el capital social, y en la pérdida de confianza en el gobierno".

3.1 PREDICCIÓN DEL CRIMEN Y GEORREFERENCIACIÓN

Entendiendo el crimen como una problemática pública que afecta a la sociedad en general, podemos suponer que, para el gobierno local una estrategia adecuada en la forma que se deben gestionar los recursos económicos destinados para la seguridad es en la prevención criminal a partir de la estimación del número de crímenes que podrían ser cometidos en una zona determinada. Y para lograr dicha estimación se hace necesario el uso de herramientas que faciliten visualizar los datos estadísticos históricos que se tienen de una zona y asociarlos a espacios físicos reales en los mismos territorios. Para lograr esto último debemos tener en cuenta la ciencia geográfica, específicamente la línea de la geografía humana, la cual permite estudiar y analizar los fenómenos de origen humano que se asocian a los espacios físicos en los que habitan[12].

El comportamiento antisocial está intrínsecamente vinculado a un lugar y un tiempo específicos. Analizar detalladamente estas variables espaciotemporales, junto con otros factores geográficos, es crucial porque proporciona información valiosa que puede aumentar la eficacia y eficiencia de las organizaciones encargadas de brindar servicios de seguridad [13].

En el análisis de patrones criminales, se hace especial énfasis en el uso de Sistemas de Información Geográfica (SIG). Estos sistemas están diseñados para gestionar y administrar datos que pueden ser georreferenciados, incluyendo datos geográficos, metadatos y mapas temáticos [14]. Los SIG permiten al usuario integrar una amplia variedad y volumen de información, asociándola con puntos específicos en los mapas de áreas que requieren intervención. Esta capacidad de correlación facilita una interpretación efectiva de los patrones criminales en las zonas delimitadas, lo cual es crucial para el mantenimiento de la seguridad y el orden[15].

Estos patrones concretamente se dan con base en el análisis relacional entre los hechos criminales, el territorio y la sociedad (economía, cultura, etc.), dando la oportunidad de tener una representación visual, que a su vez facilita un análisis comparativo y conjunto del fenómeno criminal.

3.2 CONTEXTO HISTÓRICO DE LA PREDICCIÓN DEL CRIMEN

A través de la Historia el análisis de los crímenes y las conductas criminales se han analizado desde distintas disciplinas humanas cómo la psicología y la sociología, no obstante la policía en su

trayectoria histórica de lucha contra el crimen, ha hecho uso de la zonificación de los territorios para identificar los puntos de ocurrencia de crímenes dentro de los límites de su rango de actuación, pues estos mapas que se usaban eran representaciones cartográficas de los limites jurisdiccionales o administrativos establecidos por los gobiernos locales.

Lo anterior evidencia la importancia del uso del elemento geográfico contra el crimen, aunque su uso sólo se diera para identificar el punto geográfico de la ocurrencia de este, y no para hacer un análisis más detallado del mismo.

Harries K. (1999) plantea que históricamente en el proceso de análisis de crímenes surgieron tres "escuelas" principales:

- La Cartográfica o geográfica
- La tipológica
- La Ecología Social

Harries, también argumenta que es altamente probable que el primer uso de la cartografía digital y computarizada para abordar y analizar reportes de hechos criminales se haya producido en 1960 en San Luis, y sorprendentemente la participación de los geógrafos profesionales sólo se dio en la década de los 70's cuando Lloyd Haring organizó un seminario en la Universidad Estatal de Arizona sobre la geografía de la delincuencia[16].

3.3 TECNICAS DE PREDICCION DEL CRIMEN

La prevención del crimen en la historia ha sido el objetivo de diferentes instituciones encargadas de mantener el orden público y la seguridad de los ciudadanos de un territorio, esto hecho a partir del análisis de datos y su relación con los sitios de ocurrencia. Esta problematización ha llevado a la formulación de diferentes intervenciones en materia política, económica y social, como también de intervenciones académicas, dando origen a la formulación de teorías y técnicas de análisis cómo la Sociología Criminal de Enrico Ferri, la Intervención Multiagencial de Jock Young, Roger Matthews y John Lea, y finalmente la Escuela de Chicago centrada en el trabajo de Frederic Trasher, trabajo que a su vez se puede enmarcar en la escuela de la Ecología Social mencionada anteriormente[17].

Con referencia a lo planteado anteriormente, en el libro: La Criminología que Viene [18] se plantea qué: "la investigación criminal a través de la historia ha buscado métodos y herramientas que fueran capaces de predecir la comisión de un delito para actuar con anterioridad. Inicialmente con la predicción de la reincidencia, para luego abordar las circunstancian ambientales que se relacionaban

con la comisión del delito (Espacio; Tiempo y Distribución). Desde la introducción de los SIG se ha logrado desarrollar estrategias de predicción dentro de la denominada Predictive Policing. Y en la actualidad esto se refuerza con el análisis del comportamiento social a través de la data mining" [18].

3.4 TABLEROS DE VISUALIZACIÓN DE DATOS

Podemos decir que la visualización de datos es la ilustración gráfica a través de elementos visuales (gráficos y/o mapas, entre otros.) que se hace de un evento o series de eventos o fenómenos y de cómo ha sido su comportamiento y así poder de manera más clara evidenciar y comprender las tendencias, los valores atípicos y los patrones en los datos. Todo esto permite hacer análisis a grandes cantidades de información y facilitar la toma de decisiones.

Históricamente, la necesidad de obtener una respuesta rápida en el análisis e interpretación gráfica de los datos origina la demanda de medios o herramientas que permitan la expresión visual, por ello se dio paso a la creación de herramientas de autoservicio y auto consulta (en la nube), las cuales, a partir de un software dotado con una variedad de filtros regulables le permitiera al usuario explorar sus datos y realizar informes de forma inmediata, estas herramientas inicialmente se enfocaron en la línea de la Inteligencia de Negocios (BI), pero en la actualidad su campo de uso es más amplio, incluyendo las ciencias básicas y las humanidades. La herramienta que para el caso se va a utilizar es Power BI, la cual fue lanzada en el mes de julio del año 2011 como un complemento de Microsoft Excel llamado: Power Query, Power Pivot y Power view.

3.5 POWER BI COMO HERRAMIENTA DE ANÁLISIS

Cómo lo plantea Microsoft Corporation en su página, la herramienta de Power BI es conjunto ordenado de servicios cómo: software, aplicaciones y conectores que permiten trabajar con una colección de datos almacenados en una computadora, los cuales no se encuentran relacionados directamente, y convertirlos en información coherente e interactiva, la cual se representa de forma atractiva en representación visual. En consecuencia, permite almacenar los datos en la nube para trabajarlos en línea para incluir nuevos hallazgos y así poder hacer cambios en tiempo real de forma individual o junto con varios usuarios.

Los análisis realizados se pueden presentar a modo de informes y paneles los cuales son una hoja o cómo lo denomina la herramienta "un lienzo" dónde se alojan elementos visuales que provienen de un conjunto de datos que a su vez provienen de los informes, planteado un resumen estructurado y de fácil comprensión debido a la interfaz gráfica allí mostrada.

3.6 POWER BI EN EL ANÁLISIS DE DATOS ESPACIALES

El uso de la herramienta POWER BI en conjunto con herramientas SIG para el análisis espacial, ofrece la capacidad de presentar mapas que facilitan el entendimiento de la información allí presentada, ya que permite mostrar las ubicaciones pertinentes e identificar las relaciones de los datos y los patrones contenidos en los mismos. Un claro ejemplo de esto es la facilidad al presentar la información en paneles dónde se alojan mapas de calor presentados de manera ágil, o el poder agrupar los datos en diferentes clústeres que se han generado en el procesamiento previo, y así identificar patrones ocultos en los datos.

Todo esto, en función de las diferentes cualidades espaciales de la zona física de estudio, visualizando finalmente los datos en mapas de diferentes características cómo, por ejemplo:

- Datos espaciales con precisión.
- Presentación de datos en mapas inteligentes.
- Selecciones basadas en áreas.
- Análisis de la demografía de áreas de interés.

Por lo tanto, la integración de las dos herramientas proporciona información más compleja de manera más clara que facilita la elaboración de conclusiones y la toma de decisiones.

3.7 MACHINE LEARNING

El aprendizaje automático o Machine Learning (ML), es una rama de la informática que estudia algoritmos y técnicas para automatizar soluciones a problemas complejos que son difíciles de programar mediante métodos convencionales, considerado un subconjunto de la inteligencia artificial (IA), emula el "aprendizaje" experiencial asociado con la inteligencia humana. Utiliza algoritmos computacionales para analizar y mejorar sus capacidades mediante grandes conjuntos de datos de entrada y salida. Estos algoritmos reconocen patrones en los datos y "aprenden" de manera efectiva para entrenar a la máquina a hacer recomendaciones o tomar decisiones autónomas. Cuanto mayor sea el conjunto de datos, más precisos se vuelven los algoritmos. El objetivo de un algoritmo de ML es aprender un modelo o conjunto de reglas a partir de un conjunto de datos etiquetados para predecir correctamente las etiquetas de nuevos datos que no están en el conjunto inicial. Tras suficientes repeticiones y modificaciones del algoritmo, la máquina puede tomar una entrada y predecir una salida. Estas predicciones se comparan con resultados conocidos para evaluar la precisión del algoritmo, que se ajusta iterativamente para mejorar su capacidad predictiva [19], [20].

Modelos de aprendizaje

Aprendizaje Supervisado

En el aprendizaje supervisado, se proporciona a la máquina un conjunto de datos junto con las respuestas correctas correspondientes a esos datos. Los pares de datos de entrada y salida en el conjunto de entrenamiento se utilizan para calibrar los parámetros del modelo de ML. El algoritmo de aprendizaje recibe un gran conjunto de datos etiquetados con respuestas. El algoritmo debe aprender las características clave de cada punto de datos en el conjunto para determinar la respuesta. De esta manera, cuando se le proporciona un nuevo punto de datos, el algoritmo debería poder predecir el resultado o la respuesta correcta basándose en las características clave aprendidas. Una vez que el modelo ha sido entrenado con éxito, puede usarse para predecir la variable objetivo y con nuevos datos de las características de entrada x. En el aprendizaje supervisado, se distinguen dos tipos de problemas: regresión, donde se predice un valor numérico (por ejemplo, número de usuarios), y clasificación, donde el resultado de la predicción es una categoría, como "observadores" o "compradores"[20], [21].

Aprendizaje no supervisado

El aprendizaje no supervisado ocurre cuando el sistema de aprendizaje detecta patrones sin etiquetas o especificaciones preexistentes. Los datos de entrenamiento solo consisten en variables x, con el objetivo de encontrar información estructural de interés, como grupos de elementos que comparten propiedades comunes (conocido como clustering) o representaciones de datos proyectadas de un espacio de alta dimensión a uno más bajo (conocido como reducción de dimensionalidad) [20], [21].

Aprendizaje por refuerzo

En un sistema de aprendizaje por refuerzo, en lugar de proporcionar pares de entrada y salida, se describe el estado actual del sistema, se especifica un objetivo, se proporciona una lista de acciones permitidas y sus restricciones ambientales, y se deja que el modelo de ML experimente el proceso de alcanzar el objetivo por sí mismo utilizando el principio de prueba y error para maximizar una recompensa. Este enfoque es especialmente útil en situaciones donde el entorno cambia constantemente, como en la conducción de vehículos o juegos como el ajedrez y el backgammon, donde la máquina debe adaptar sus respuestas a un entorno en constante cambio. Además, es eficaz en espacios de estado enormes, como los juegos multijugador, donde las configuraciones posibles son casi infinitas y el aprendizaje debe permitir que la máquina perciba el entorno y elija acciones basadas en su estado interno y el ambiente externo, con el objetivo de maximizar un objetivo específico predefinido, como mantenerse en el carril al conducir [20], [21].

Conceptos fundamentales

Algoritmos y modelos

Un "algoritmo" en aprendizaje automático es un procedimiento aplicado a datos para crear un "modelo". Los algoritmos se ajustan a un conjunto de datos y existen varios tipos con diferentes funciones, como regresión (predicción de valores continuos), clasificación (asignación de valores categóricos) y clustering (agrupación de elementos similares).

Un "modelo" es el resultado de aplicar un algoritmo a los datos. Representa lo aprendido y puede guardar la funcionalidad del algoritmo para hacer nuevas predicciones. Un modelo entrenado eficientemente puede predecir con precisión datos similares en el futuro.

La principal diferencia entre un algoritmo y un modelo es que el algoritmo es el procedimiento utilizado para crear el modelo. El modelo incluye los datos y un procedimiento para predecir nuevos datos [22].

Feature Engineering

La ingeniería de características es el proceso de utilizar conocimientos del dominio para seleccionar y transformar las variables más relevantes de los datos brutos al crear un modelo predictivo con aprendizaje automático o modelado estadístico. El objetivo es mejorar el rendimiento de los algoritmos de ML. El proceso para implementar la ingeniería de características incluye pasos de preprocesamiento que transforman los datos brutos en características utilizables en modelos predictivos, que consisten en una variable de resultado y variables predictoras. Desde 2016, algunos software de ML ofrecen ingeniería de características automatizada [23].

La ingeniería de características en ML comprende cuatro pasos principales: creación, transformación, extracción y selección de características:

1. **Creación de Características**: Identificación y combinación de variables para crear nuevas características derivadas que tengan mayor poder predictivo.
2. **Transformaciones**: Manipulación de variables predictoras para mejorar el rendimiento del modelo, asegurando que las variables estén en la misma escala y dentro de un rango aceptable.

3. **Extracción de Características**: Creación automática de nuevas variables a partir de datos brutos para reducir el volumen de datos a un conjunto más manejable, utilizando métodos como análisis de componentes principales y análisis de texto.
4. **Selección de Características**: Análisis y ranking de características para determinar cuáles son irrelevantes o redundantes y cuáles deben priorizarse en el modelo.

Estos pasos son cruciales para crear algoritmos de ML precisos y eficientes.

Entrenamiento y evaluación del modelo

Entender la diferencia entre datos de entrenamiento y datos de prueba es crucial porque uno entrena al modelo y el otro confirma su precisión con datos nuevos. Usar incorrectamente estos conjuntos puede llevar a resultados engañosos y decisiones incorrectas.

Datos de Entrenamiento: Los datos de entrenamiento en el aprendizaje automático son un subconjunto del conjunto de datos utilizado para enseñar al modelo a reconocer patrones y aprender. Este conjunto es generalmente más grande que el de prueba, ya que se necesita proporcionar la mayor cantidad de datos posible al modelo para que aprenda de manera significativa. Los algoritmos de ML analizan estos datos para encontrar patrones y tomar decisiones, mejorando con más datos relevantes. Los datos de entrenamiento varían según el tipo de aprendizaje utilizado: supervisado o no supervisado.

Datos de Prueba: Una vez que el modelo de ML está entrenado con los datos de entrenamiento, se utilizan datos no vistos, llamados datos de prueba, para evaluar su rendimiento. Estos datos deben representar el conjunto de datos real y ser lo suficientemente grandes para generar predicciones significativas. La finalidad es verificar si el modelo funciona correctamente con datos nuevos y ajustar u optimizar el modelo si es necesario. En aprendizaje supervisado, los resultados conocidos se eliminan del conjunto de datos al crear el conjunto de prueba y se comparan con las predicciones del modelo entrenado.

3.8 ANTECEDENTES Y TRABAJOS RELACIONADOS

Los métodos de prevención del crimen han evolucionado de los métodos reactivos, es decir cuando el crimen ya ha ocurrido a métodos que utilizan datos que se producen diariamente. En este sentido, los métodos estadísticos y los sistemas de información geográfica han sido fundamentales para ubicar los “puntos calientes” de crimen. Actualmente la emergencia de la inteligencia artificial (AI) en el campo de la seguridad pública ha permitido la transformación de las estrategias de prevención del

crimen. Avances en el Aprendizaje Automático, reconocimiento de patrones y la analítica predictiva han permitido esta transformación.

3.8.1 Modelos de Aprendizaje Automático (ML) por Machine Learning

La aplicación del aprendizaje automático (ML) en la predicción y prevención de delitos ha experimentado avances significativos, aprovechando conocimientos basados en datos para comprender y abordar mejor diversos aspectos del comportamiento delictivo. El estado del arte en este ámbito abarca un amplio espectro de enfoques, cada uno de ellos adaptado a tipos específicos de delitos o aspectos de la aplicación de la ley y la seguridad pública.

Uno de sus usos ha sido, por ejemplo, en la prevención del crimen juvenil y el crimen basado en el género. En un estudio sobre predictores de delitos juveniles se usó ML para construir un sistema de apoyo a la decisión que predice diferentes tipos de delitos juveniles. Al incorporar factores de riesgo criminógenos de una variedad de teorías criminológicas, el estudio demuestra la aplicabilidad de estas teorías a los delitos juveniles tradicionales y cibernéticos, mejorando la precisión predictiva y ofreciendo conocimientos valiosos para los programas de intervención temprana[24]. También se aplican métodos híbridos de ML para predecir la reincidencia en delitos de género utilizando datos del sistema español Vio Gen[25]. Este enfoque innovador supera significativamente a los sistemas tradicionales de evaluación de riesgos, destacando la eficacia del ML para mejorar la protección policial y la gestión de recursos[26].

También se está usando ML para construir patrones que permitan predecir diferentes aspectos del crimen. Centrándose en Londres, un sistema predictivo desarrollado utilizando diversos conjuntos de datos de fuente abierta, incluidos registros policiales y redes sociales, revela patrones en el comportamiento humano y la demografía cruciales para la predicción del crimen. Este sistema basado en ML no solo pronostica delitos potenciales sino que también permite medidas preventivas específicas, mejorando significativamente las estrategias de seguridad pública[27].

La integración del aprendizaje automático en la predicción y prevención de delitos ha revolucionado el campo, ofreciendo diversas metodologías que atienden diversos aspectos del comportamiento delictivo y las necesidades de aplicación de la ley. Desde mejorar los sistemas de justicia juvenil hasta mejorar la seguridad urbana y abordar la violencia de género, los modelos de aprendizaje automático brindan conocimientos y herramientas fundamentales que refuerzan significativamente los esfuerzos de prevención del delito.

3.8.2 Vigilancia con IA

La integración de tecnologías de vigilancia con ayuda de IA en los espacios públicos es común hoy en día, generalmente haciendo uso para mejorar la seguridad pública y el control de la salud. Se destaca el despliegue de tecnologías de reconocimiento facial en tiempo real y aplicaciones de rastreo de contactos, que ayudan en la aplicación de la ley y el seguimiento de comportamientos sociales[28]. Estas herramientas han demostrado ser especialmente valiosas durante crisis como pandemias y aumentos de la delincuencia[28]. Para evaluar los posibles beneficios y riesgos de estas tecnologías de vigilancia, se propone un marco multidimensional que evalúa su funcionalidad, el manejo del consentimiento del usuario y su impacto social. Este marco ayuda a determinar cómo se pueden diseñar e implementar estas herramientas de manera responsable, garantizando que respeten la autonomía del usuario y al mismo tiempo mejoren las capacidades de las autoridades públicas[28].

También se ha reportado el diseño de un sistema de monitoreo de delitos (CMS) en tiempo real que emplea cámaras de vigilancia integradas con técnicas de aprendizaje profundo (DL – Deep Learning) para detectar actividades delictivas al instante y notificar a las autoridades. Este sistema tiene como objetivo abordar las limitaciones del monitoreo humano, como los tiempos de reacción lentos y la falta de atención. El CMS opera a través de etapas de detección de armas, detección de violencia y reconocimiento facial, utilizando algoritmos avanzados como YOLOv5 y MobileNetv2 para garantizar una alta precisión y capacidad de respuesta en tiempo real. El CMS ha demostrado una eficacia significativa en varios escenarios del mundo real, mostrando su potencial para mejorar la seguridad y la protección pública [29].

4. METODOLOGIA

Para el desarrollo de la herramienta "Precrimen" se utilizan las metodologías de análisis predictivo e Ingeniería de datos. Específicamente, se trata de una metodología de predicción de crimen. Las fases de esta metodología son:

1. **Recopilación y preparación de datos:** En esta fase se investigaron las diferentes fuentes de datos relacionados con la comisión de crímenes a nivel nacional e internacional. A nivel nacional se accedió a la base de datos de la Policía Nacional sobre Delitos sexuales y homicidios por accidentes de tránsito. A nivel internacional, una de las bases de datos más completas es la de delitos violentos y delitos contra la propiedad del Departamento de Policía de los Ángeles, que, como ventaja para el estudio predictivo de crímenes, esta georreferenciada.
2. **Análisis exploratorio de datos:** Este análisis permitió comprender la estructura de la base de datos, la identificación de patrones y correlaciones, detectar los datos extremos, revisar los datos faltantes y generar hipótesis y posibles técnicas a utilizar para la predicción.
3. **Preprocesamiento:** En esta etapa se realizaron actividades y cómputos geoespaciales para segmentar los datos por ubicaciones geográficas. Lo anterior con el objetivo de producir un modelo más preciso y evitar pronósticos que no tengan impacto para la toma de decisiones. Se resalta que el uso de datos de geoposicionamiento del delito permitió construir un algoritmo que construye geo celdas para segmentar en porciones más pequeñas el conjunto de datos y agrega una nueva característica llamada Zona, que permite ser usada en la construcción del modelo y con ello enriquecer y mejorar la precisión del modelo.
4. **Feature Engineering:** Con este proceso se crean nuevas características que mejoran la capacidad predictiva de los modelos. Para este estudio se crearon nuevas variables a partir de la variable fecha.
5. **Construcción del modelo predictivo:** Para la construcción del modelo predictivo se utilizó el servicio de Colab de Google. Inicialmente se hicieron pruebas con las zonas preestablecidas en la base de datos, después se redujeron esas zonas a cuadriculas más pequeñas.
6. **Evaluación y ajuste del modelo:** La evaluación el modelo se realizó usando el 20 % de los datos y calculando la medida de error absoluto medio

7. **Implementación de interfaz de visualización:** La interfaz de visualización se implementó en Power BI y en Python usando las librerías folium, matplotlib y seaborn. En el caso de Power Bi, se importaron los datos desde archivos Excel, se realizó la transformación o ajuste de datos en el formato requerido utilizando Power Query Editor. Seguidamente se definieron las relaciones entre los datos del modelo y se crearon columnas calculadas (por ejemplo, agregación por departamentos) usando DAX (Data Analysis Expressions). Finalmente, se diseñó la interfaz seleccionando las visualizaciones adecuadas, se configuraron las propiedades y se realizan los ajustes de forma de la visualización (Temas y estilo).

Ilustración 1 Metodología del proyecto

5. ACTIVIDADES Y RESULTADOS

5.1 OBJETIVO IDENTIFICAR, CLASIFICAR Y VISUALIZAR LOS DATOS CONECTANDO FUENTES ABIERTAS Y/O FUENTES OFICIALES QUE DISPONEN INFORMACIÓN DE DELITOS EN COLOMBIA.1.

5.1.1 Identificación de datos en fuentes abiertas

Ciudad de Los Ángeles, California: Para desarrollar el proyecto se seleccionó el conjunto de datos de crímenes en Los Ángeles, del 1 de enero de 2020 al 4 de marzo de 2024. Estos datos se obtuvieron de la página oficial de datos abiertos del Gobierno de Estados Unidos (https://catalog.data.gov/dataset/crime-data-from-2020-to-present).

La base de datos tiene 908 K registros y 26 columnas, incluyendo la georreferenciación del lugar donde se cometieron los delitos en la ciudad de Los Ángeles. La base de datos cuenta con una licencia Creative Commons que permite el acceso y uso público bajo algunos términos y condiciones [30].

La estructura de la base de datos es:

Tabla 1 Variables de la base de datos de crímenes de Los Ángeles

Nombre de la variable	**Descripción**	**Tipo**	**Categorías**
Date Rptd	Fecha de reporte del crimen	Cuantitativo discreto	fecha
DATE OCC	Fecha de ocurrencia del delito	Cuantitativo discreto	fecha
TIME OCC	Hora de ocurrencia del delito	Cuantitativo discreto	fecha
AREA	Comunidad donde ocurre el delito	Cualitativo nominal	Una de las 21 estaciones de policía en las comunidades de LA. Se numeran de 1 a 21
AREA NAME	Nombre de la comunidad donde ocurre el delito	Cualitativo nominal	Una de las 21 estaciones de policía en las comunidades de LA

Rpt Dist No	Código de la subárea dentro de la comunidad	Cuantitativo nominal	
Part 1-2	Clasificación por gravedad del crimen del FBI	Cualitativo nominal	Part I-Crimenes serios Part II- Crímenes menos serios
Crm Cd	Código del crimen cometido	Cuantitativo	
Crm Cd Desc	Definición del crimen cometido	Cualitativo nominal	
Mocodes	Modus operandi	Cualitativo nominal	
Vict Age	Edad de la victima	Cuantitativo discreto	
Vict Sex	Sexo de la victima	Cualitativo	
Vict Descent	Etnia/origen	Cualitativo nominal	A - Other Asian B - Black C - Chinese D - Cambodian F - Filipino G - Guamanian H - Hispanic/Latin/Mexican I - American Indian/Alaskan Native J - Japanese K - Korean L - Laotian O - Other P - Pacific Islander S - Samoan U - Hawaiian V - Vietnamese W - White X - Unknown Z - Asian Indian
Premis Cd	Código del tipo de estructura, vehículo o locación donde ocurrió el crimen	Cuantitativo nominal	

Premis Desc	Definición del tipo de estructura, vehículo o locación donde ocurrió el crimen	Cualitativo nominal	
Weapon Used Cd	Código del tipo de arma usada para el crimen	Cualitativo nominal	
Weapon Desc	Definición del tipo de crimen usada para el crimen	Cualitativo nominal	
Status	Código del estado del caso	Cualitativo nominal	
Status Desc	Definición del estado del caso	Cualitativo nominal	
Crm Cd 1	Código del crimen cometido	Cualitativo nominal	
Crm Cd 2	Código de un delito adicional cometido, menos serio que el Crm Cd 1	Cualitativo nominal	
Crm Cd 3	Código de un delito adicional cometido, menos serio que el Crm Cd 1	Cualitativo nominal	
Crm Cd 4	Código de un delito adicional cometido, menos serio que el Crm Cd 1	Cualitativo nominal	
LOCATION	Dirección del crimen	Cualitativo nominal	
Cross Street	Calle transversal de la dirección aproximada	Cualitativo nominal	
LAT	Latitud	Cuantitativo	

LON	Longitud	Cuantitativo	

Policía Nacional (fuente secundaria sin datos espaciales): La institución cuenta con información accesible en su página web, que se puede descargar en archivos CSV, así mismo publica los conjuntos de datos en el portal de datos abiertos oficial de Colombia que gestiona del Ministerio de Tecnologías de la Información y Comunicaciones (datos.gov.co). En los conjuntos de datos publicados por la Policía Nacional se incluyen los delitos de: homicidio, homicidio en accidente tránsito, lesiones en accidente de tránsito, lesiones personales, hurto a comercio, hurto de automotores, hurto a personas, hurto a residencias, hurto de motocicletas, piratería terrestre, Hurto a cabezas de ganado (abigeato), hurto a entidades financieras, terrorismo, delitos sexuales, violencia intrafamiliar y amenazas. Estos datos se encuentran desagregados por tiempo, modo, lugar y en algunos casos, datos demográficos de la víctima.

En el caso de la base de datos de víctimas de delitos sexuales, la estructura es la siguiente:

Nombre de la base de datos: Reporte__Delitos_sexuales_Polic_a_Nacional_20240310.csv

Fuente: https://www.datos.gov.co/Seguridad-y-Defensa/Reporte-Delitos-sexuales-Polic-a-Nacional/fpe5-yrmw/data_preview

Propósito: Estadísticas de delitos de impacto en Colombia

Tamaño: La base de datos de *delitos sexuales* contiene 9 columnas y 319.534 registros.

Alcance: casos de delitos sexuales a nivel municipal.

Descripción de variables

Tabla 2 Variables de la base de datos de Colombia

Nombre de la variable	**Descripción**	**Tipo**	**Categorías**
Departamento	Departamento donde ocurrió el delito	Cualitativa nominal	Los 32 departamentos del país
Municipio	Municipio donde ocurrió el delito	Cualitativa nominal	Municipios en el departamento

Código DANE	Corresponde a la nomenclatura estandarizada para identificación de los municipios	Cualitativa nominal	Códigos de cada municipio de acuerdo con el DANE
Armas medios	Arma o medio utilizado para cometer el delito	Cualitativa nominal	Arma blanca cortopunzante; armas de fuego; cintas/cinturón; contundentes; escopolamina; esposas; no reportado; sin empleo de armas
Fecha hecho	Fecha en la que ocurrió el delito	Cuantitativa discreta	Fecha en el formato DD/MM/AA
Delito	Tipo de delito de acuerdo con lo establecido en el código penal	Cualitativa nominal	Delitos descritos en el capítulo I del título IV del código penal Colombiano
Genero	Genero de la víctima del delito	Cualitativa nominal	Femenino, Masculino; No reportado
Agrupa edad persona	Grupo de edad al cual pertenece la victima	Cualitativa nominal	Adolescentes (13 a 17 años); Adultos (18 años en adelante); Menores (0 a 12 años); No reportado
Cantidad	Número de víctimas del mismo delito en el mismo municipio en la misma fecha	Cuantitativo discreto	números de 1 en adelante

La base de datos contiene algunos campos vacíos que son tomados como datos perdidos. El período de los datos para este trabajo es desde el 1 de enero de 2010 hasta el 31 de diciembre de 2023.La estructura de la base de datos de homicidios por accidente de tránsito es la siguiente:

Nombre de la base de datos: Homicidios_accidente_de_tr_nsito_Polic_a_Nacional_20240310 .csv

Fuente: https://www.datos.gov.co/Seguridad-y-Defensa/Homicidios-accidente-de-tr-nsito-Polic-a-Nacional/ha6j-pa2r/about_data

Propósito: Estadísticas de delitos de impacto en Colombia

Tamaño: La base de datos de *homicidios por accidente de tránsito* contiene 8 columnas y 77.223 registros.

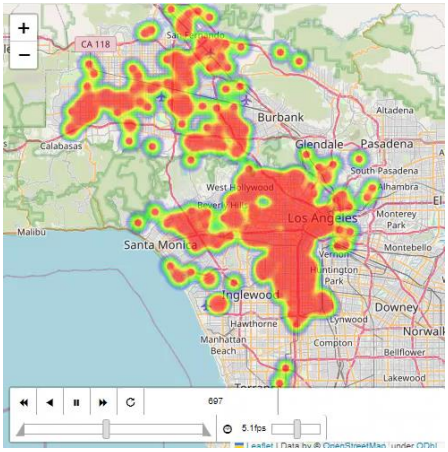
Alcance: casos de homicidios por accidentes de tránsito a nivel municipal, las bases de datos se descargan por año y se pueden agregar en una sola base de datos

Las variables corresponden a las mismas que la base de datos de delitos sexuales, a excepción de la columna delito (subtipo de delito), que no está presente en esta base de datos y la variable "arma medios" se refiere al vehículo en el cual se transportaba la victima de homicidio.

5.1.2 EXTRACCIÓN DE DATOS

Para esta actividad se utilizó la Liberia de Python pandas para realizar lectura del conjunto de datos inicial. Debido a que el conjunto de datos de la ciudad de Los Ángeles de un tamaño superior a 200Mb, fue necesario realizar el almacenamiento del conjunto de datos en los servicios de Drive de Google para así poder interactuar desde un cuaderno desarrolla en Google Collab.

Gráfica 1 Mapa de calor de delitos de la ciudad de Los Ángeles con serie de tiempo diaria



5.1.3 LIMPIEZA Y PROCESAMIENTO DE DATOS

El conjunto de datos publicado de la ciudad de Los Ángeles cuenta con excelente calidad de datos y menos del 2% de los registros presentan los campos Latitud y Longitud en 0 o en nulo. Para realizar la limpieza de datos se filtraron los registros de la siguiente manera.

```
data.shape
```

```
(910707, 28)
```

```
filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
```

```
filtered_df.shape
```

```
(908443, 28)
```

Ilustración 2. Comandos para filtrar los registros sin datos espaciales

Posteriormente fue necesario procesar todas las ubicaciones (latitud y longitud) para calcular una zona o geo celda que permitiera obtener mayor segmentación de crímenes por proximidad en una región de 7.7 kilómetros cuadrados por Zona y así producir un conjunto de datos en formato de serie de tiempo.

Gráfica 2 Segmentación de crímenes en geo celdas y creación de nuevo conjunto de datos para serie de tiempos

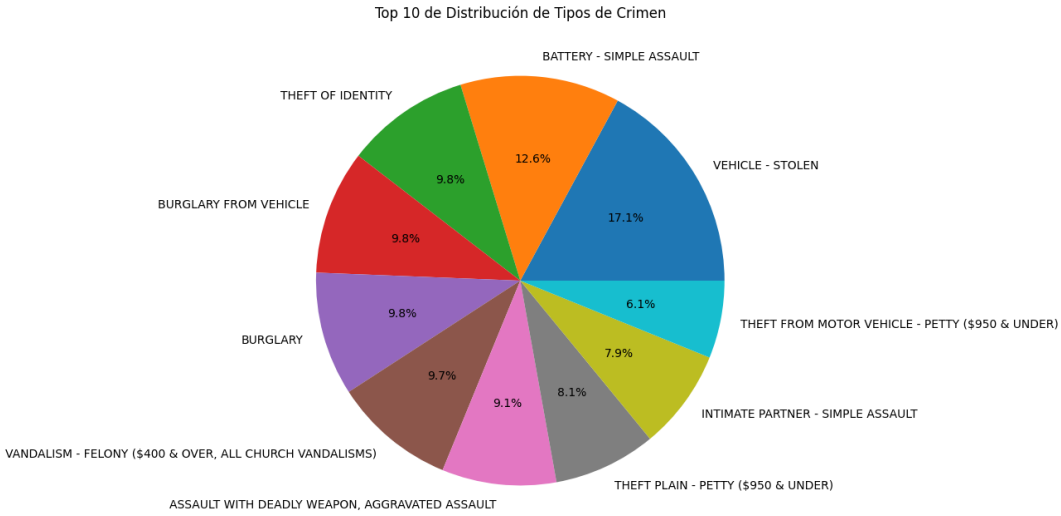
	A	B	C
1	Fecha	Zona	Cantidad de Crimenes
2	1/01/2020	1	0
3	1/01/2020	2	0
4	1/01/2020	3	0
5	1/01/2020	4	0
6	1/01/2020	5	0
2467	....	....	....
2468	5/01/2020	285	4
2469	5/01/2020	286	3
2470	5/01/2020	287	4
2471	5/01/2020	288	4
2472	5/01/2020	289	11
2473	5/01/2020	290	25
2474	5/01/2020	291	5
2475	5/01/2020	292	12
831122	....	....	....
831123	4/03/2024	539	0
831124	4/03/2024	540	0
831125	4/03/2024	541	0
831126	4/03/2024	542	0
831127	4/03/2024	543	0
831128	4/03/2024	544	0
831129	4/03/2024	545	0

5.1.4 RESULTADOS

Análisis descriptivo de la base de datos de Los Ángeles

La base de datos incluye alrededor de 139 tipos de delitos categorizados como crímenes violentos y crímenes contra la propiedad. Los que mayor frecuencia tienen en la base de datos analizada se presentan en la Gráfica 3.

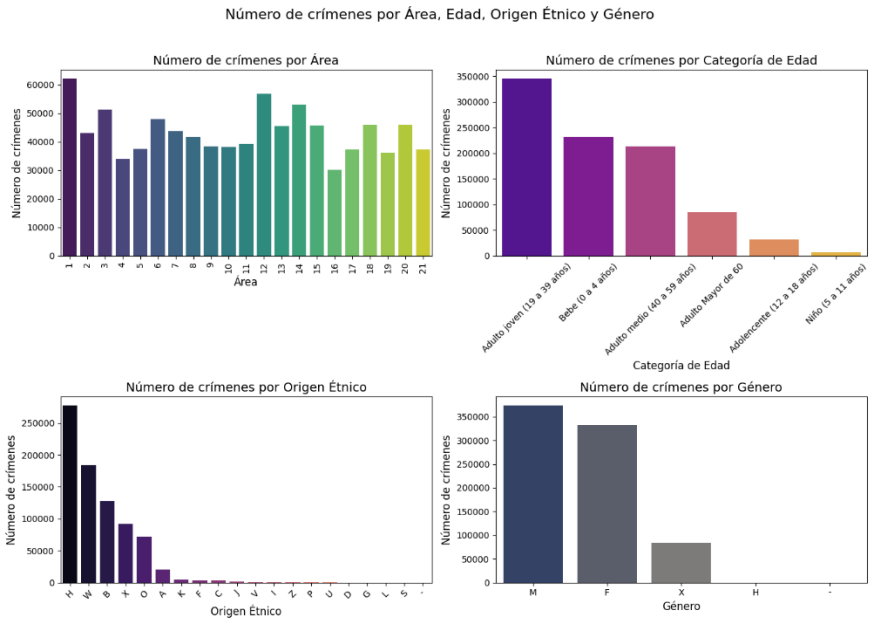
Gráfica 3 10 delitos que se presentaron con mayor frecuencia en Los Ángeles, entre el 1 de enero de 2020 y el 4 de marzo de 2024



En primer lugar, se encuentra el robo de vehículos que es un delito contra la propiedad y en segundo lugar está el asalto simple que se categoriza como crimen violento.

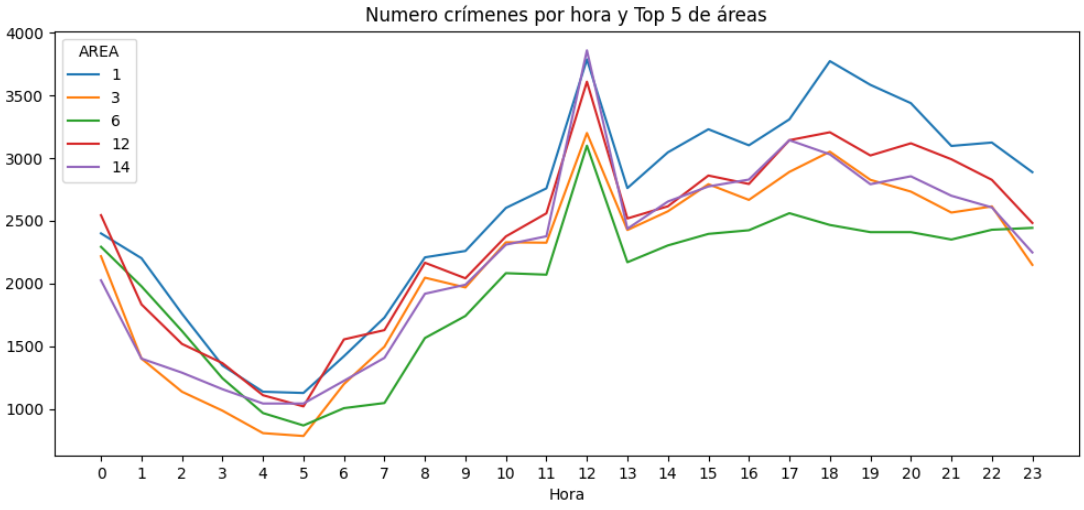
En la Gráfica 4 se presenta una caracterización del tipo de víctimas de crímenes. Las mujeres son víctimas de delitos con mayor frecuencia que los hombres; el grupo étnico que aparece como víctima con mayor frecuencia es el categorizado como “Hispanic/Latin/Mexican”; y los adultos jóvenes entre 19 y 39 años son quienes más oportunidad tienen de ser víctimas de crímenes.

Gráfica 4 Caracterización de los crímenes en Los Ángeles



En cuanto a distribución temporal durante el día, se realizó un análisis utilizando las 5 áreas con mayor cantidad de delitos. En la Gráfica 5 se puede observar que el comportamiento es idéntico en las 5 áreas, con un pico a las 12 del mediodía y con mayor ocurrencia de crímenes entre las 12 del mediodía y las 12 de la noche, comparado con la madrugada (12 de la media noche a 6 am).

Gráfica 5 Frecuencia de crímenes durante el día en las 5 áreas con mayor frecuencia



En el anexo 1. se presentan visualizaciones y resultados adicionales del análisis descriptivo.

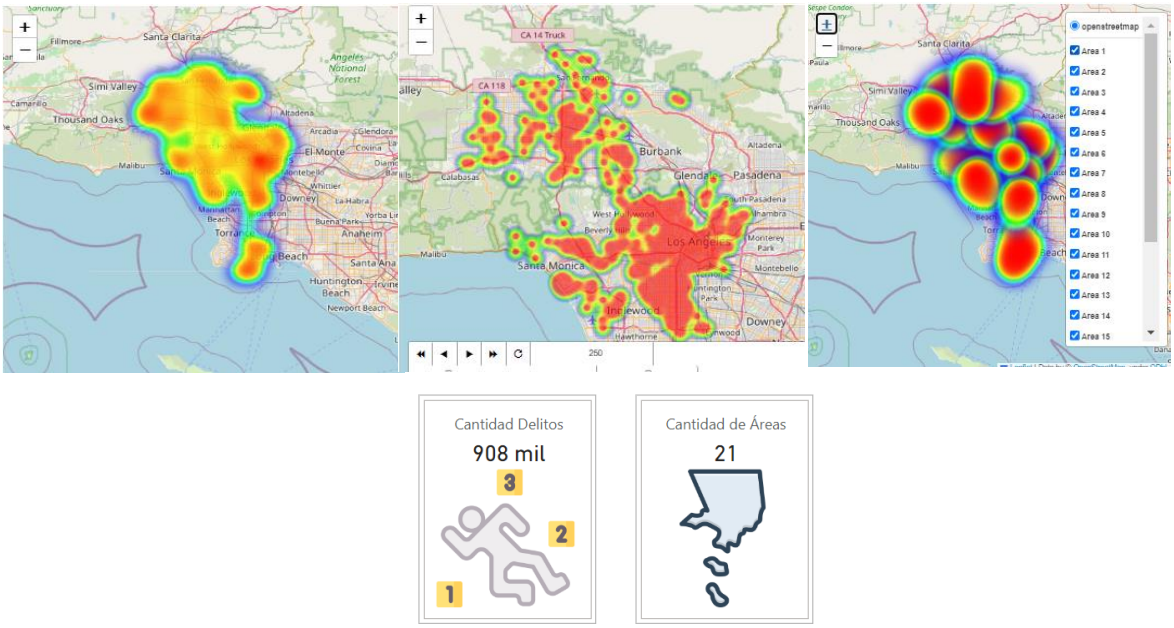
Mapas de calor por áreas

Como primer paso se realizó un análisis de los datos espaciales utilizando mapas de calor que permitió observar la distribución espacial y temporal de los delitos en las diferentes zonas.

Los siguientes mapas se realizaron usando el paquete folium de Python y permiten visualizar los mapas de calor acumulados, por área, por delito y una serie animada de tiempo que permite comprender el comportamiento espacial del delito en el tiempo.

Fundamentados en estos mapas fue posible la identificación de estrategias alternativas de complementar el conjunto de datos con una nueva columna que permita geo posicionar el delito en zonas más pequeñas que faciliten el estudio y procesamiento de los delitos.

Gráfica 6 Mapas de calor de crímenes y áreas en Los Ángeles



Durante el análisis exploratorio de datos se procesaron todos los hechos presentes en las 21 áreas de la Ciudad de Los Ángeles y se concluyó que dado la magnitud de las áreas estudias era más conveniente reducir las unidades de estudio a geo-celdas más pequeñas, para con ello fraccionar un problema complejo en subproblemas más pequeños y manejables.

Análisis descriptivo de los delitos sexuales de Colombia

El análisis de la base de datos de Colombia arrojó resultados importantes que vale la pena resaltar. El departamento con mayor cantidad de denuncias por delitos sexuales es Cundinamarca (62.802) y en cuanto a los municipios de Bogotá, Medellín y Cali son los que tienen mayor reporte. En su mayoría,

los delitos sexuales se cometen sin el uso de armas (53.75%) y las mujeres (84.81%) y los menores (46.42%) son las principales víctimas. En concordancia con lo anterior, el delito que presenta mayor frecuencia en la base de datos es el de "actos sexuales con menores de 14 años".

Análisis descriptivo de los homicidios por accidente de tránsito de Colombia

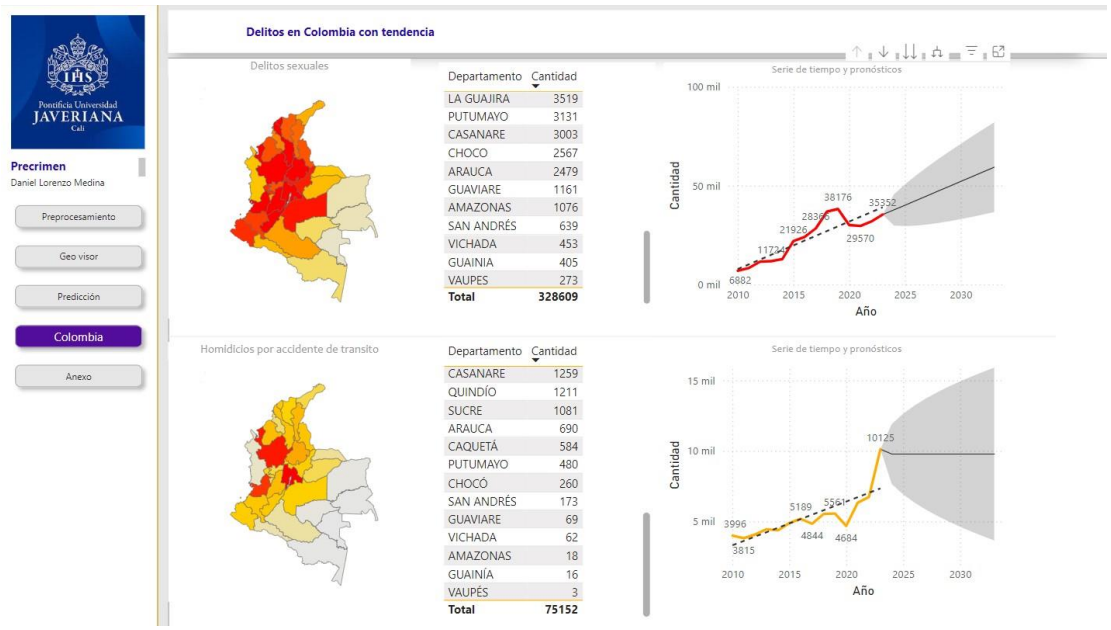


Ilustración 3. Tablero del módulo de datos de delitos sexuales y homicidios por accidente de tránsito de Colombia

La base de datos de delitos en Colombia tiene un nivel espacial municipal, lo que permite realizar análisis a este nivel y tomar decisiones generales para generar estrategias de reducción de delitos. Sin embargo, dada la complejidad del fenómeno de la criminalidad, es requerido el uso de escalas espaciales más precisas, requiriendo los datos de la latitud y la longitud para mejorar significativamente la capacidad de predecir y prevenir delitos.

Por esta razón, para cumplir con los objetivos de este trabajo, se decidió utilizar la base de datos de delitos de la Policía de Los Ángeles. Esta base de datos no solo proporciona la geolocalización precisa de los incidentes, sino que también incluye múltiples variables sobre los perpetradores y las víctimas del delito. Este nivel de detalle permite un análisis más profundo y la elaboración de estrategias más efectivas y específicas para la prevención del crimen.

Al integrar datos geoespaciales detallados con información sobre los individuos involucrados, se espera desarrollar modelos predictivos más robustos que puedan ser aplicados en diversos contextos,

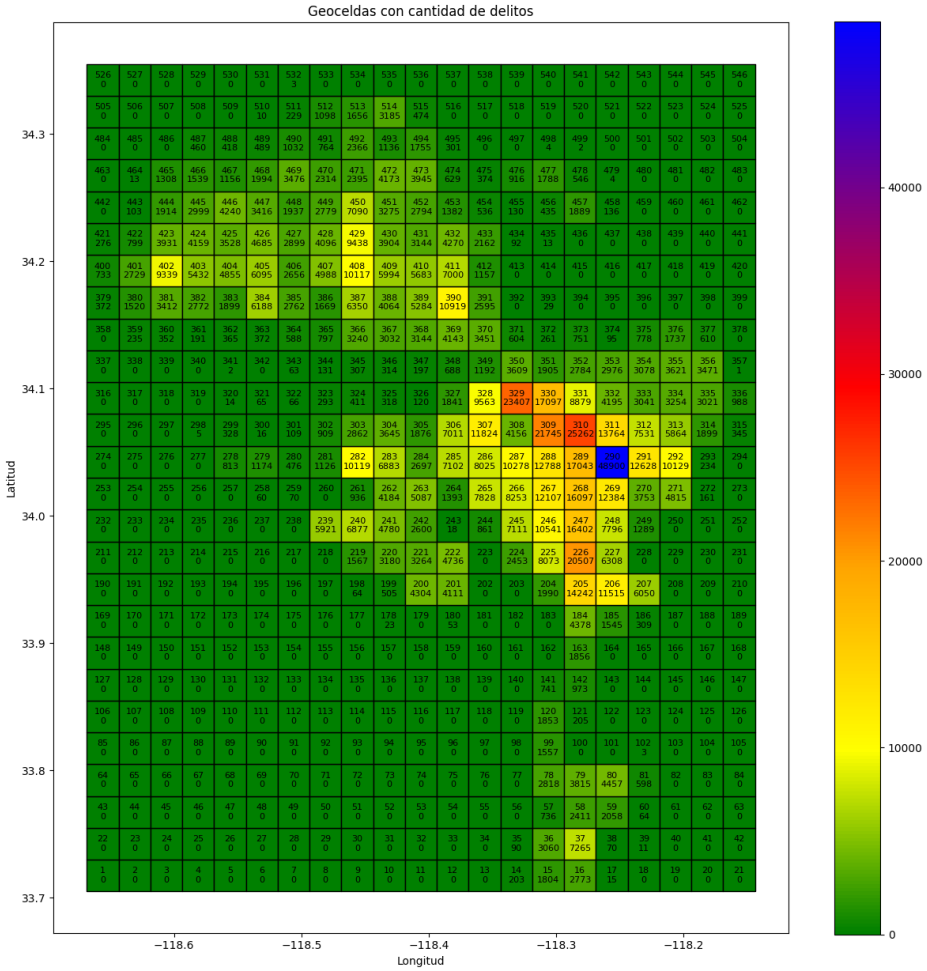
mejorando así las intervenciones y políticas de seguridad pública tanto en Colombia como en otros lugares que enfrenten desafíos similares.

5.2 OBJETIVO 2. CONSTRUIR UN MODELO DE PREDICCIÓN Y/O RECOMENDACIÓN QUE PERMITA PREDECIR LA COMISIÓN DELITOS.

Subdivisión de área de estudio en geo-celdas

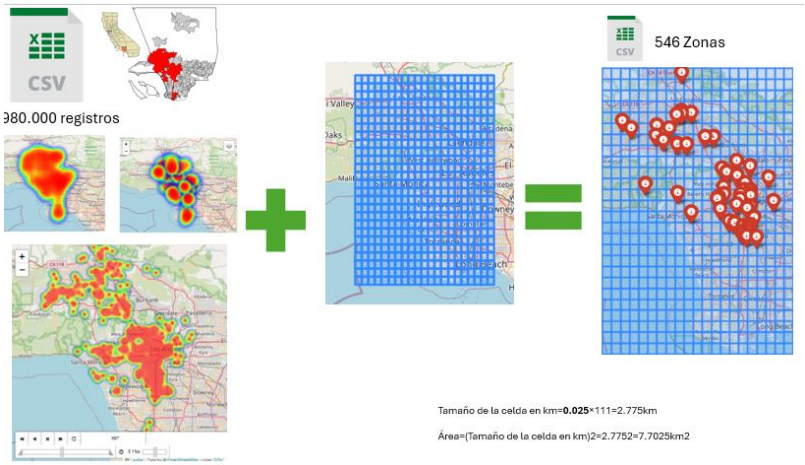
Se crearon 526 zonas con tamaño de 2.77 km² cada una, utilizando la librería folium [31] que permite la visualización de los datos en un mapa interactivo. Se importaron los datos a folium para crear un mapa base, especificando la ubicación inicial y el nivel de zoom. El archivo de geo celdas, creado previamente, se agregó al mapa como una capa y se usó el método **'choropleth'** para crear un mapa coroplético de acuerdo con el número de delitos cometidos en cada celda, tal como se observa en la Gráfica 7.

Gráfica 7 División del área de estudio en 526 geo celdas



Cada geo-celda tiene las coordenadas asociadas, al igual que el resto de los datos georreferenciados en la base de datos.

Gráfica 8 Esquema de creación de conjunto de datos por zonas y serie de tiempo



Construcción de serie de tiempo

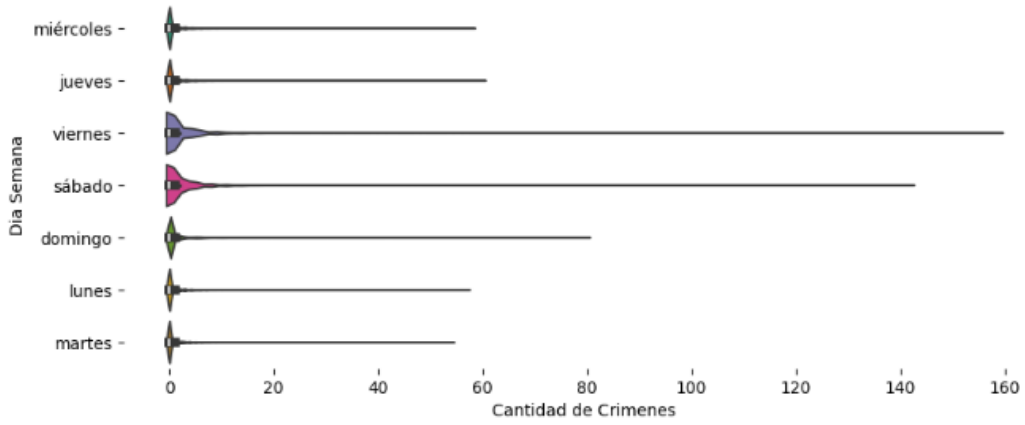
La base de datos se ajustó para su análisis como serie de tiempo, adicionando los días en los cuales no se reportaron delitos. A partir de la serie de tiempo y haciendo uso del proceso de Feature Engineering se crearon nuevas variables (features) a partir de las variables existentes (fecha) para mejorar el rendimiento del modelo. Se crearon las variables agrupadas por día de la semana y agrupada por mes y adicionalmente se creó una variable que define si la fecha en particular se trata de un día festivo o no.

Uno de los resultados se presenta en la Gráfica 9. La gráfica mostrada es un gráfico de violín que representa la distribución de la cantidad de crímenes en una base de datos desde el 1 de enero de 2020 hasta el 4 de marzo de 2024, agrupada por día de la semana. Cada "violín" muestra la densidad de la distribución de la cantidad de crímenes para cada día de la semana. La anchura de cada violín en cualquier punto muestra la densidad de los datos en ese valor. Cuanto más ancho es el violín, mayor es la densidad de datos en ese punto.

Los martes, miércoles, jueves y lunes tienen violines más estrechos, lo que indica una menor variabilidad en la cantidad de crímenes y una densidad más concentrada en valores más bajos. El viernes tiene un violín muy ancho y extendido hacia el lado derecho, indicando una alta variabilidad y una gran cantidad de crímenes, con algunos días alcanzando hasta 160 crímenes. El sábado, similar al viernes, pero con un poco menos de variabilidad. Aun así, muestra una alta cantidad de crímenes,

con un máximo que también se extiende a valores altos, aunque ligeramente menos que el viernes. El domingo muestra una variabilidad intermedia, con una distribución que también se extiende, pero no tanto como los viernes y sábados. Se observa un patrón claro donde los crímenes aumentan significativamente hacia el fin de semana (viernes y sábado) y disminuyen durante la semana laboral.

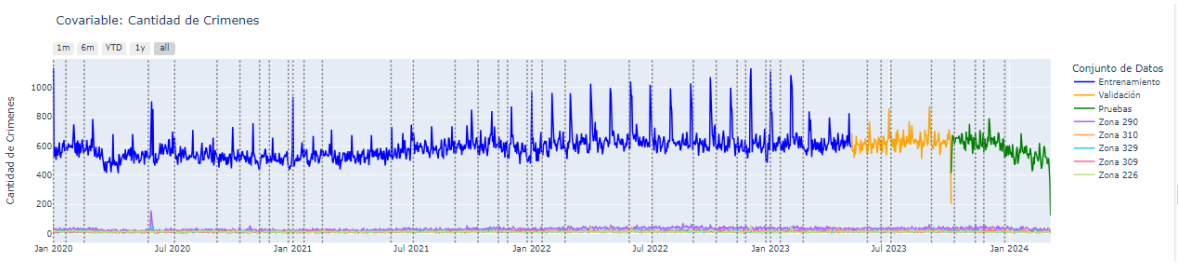
Gráfica 9 Gráfico de violín con la agrupación de la base de datos en días de la semana



Entrenamiento y validación del modelo

Para el entrenamiento del modelo con ML se usó el 80% de los datos y para la validación el 20% de los datos. La gran cantidad de datos en el segmento de entrenamiento ayuda al modelo a aprender patrones significativos y representativos de la actividad delictiva en Los Ángeles, mientras que la validación y prueba del modelo con datos separados aseguran que el modelo tenga un buen rendimiento y sea capaz de generalizar a datos nuevos. La Gráfica 10 presenta la segmentación de los datos para realizar el entrenamiento, validación y prueba del modelo.

Gráfica 10 Serie de tiempo con la segmentación de datos para entrenamiento, validación y pruebas



Se realizaron pruebas con redes neuronales LTSM y los resultados obtenidos no eran de precisión aceptable, dado que la predicción que obtenía era menor al 40%, principalmente por las desviaciones que generan cada una de las zonas donde no hay delitos.

La imagen muestra la definición de un modelo secuencial de Keras con una capa LSTM y una capa densa de salida. Se utilizaron los datos de entrenamiento y se validó con el 20% de los datos y finalmente se generó la predicción encontrando que las métricas de evaluación no favorecían el uso de este tipo de red neuronal recurrente. Las métricas utilizadas fueron RMSE, MAE y errores.

```
INPUT_SHAPE = (x_tr_s.shape[1], x_tr_s.shape[2])
modelo = Sequential()
modelo.add(LSTM(N_UNITS, input_shape=INPUT_SHAPE))
modelo.add(Dense(OUTPUT_LENGTH, activation='linear'))
```

Ilustración 4. Definición de un modelo secuencial de Keras con una capa LSTM y una capa densa de salida

A lo largo del período observado, hay una tendencia general de variabilidad constante en los crímenes, con algunos ciclos que podrían corresponder a factores estacionales o eventos específicos. Los picos más altos y recurrentes en la serie temporal sugieren momentos de alta incidencia delictiva que podrían ser objeto de un análisis más detallado.

Creación de modelos a partir de geo celdas – Modelo seleccionado

Para la construcción del modelo se usó la librería Skforecast [32] que ofrece herramientas útiles para la modelación estadística y el aprendizaje automático (machine learning). En la Ilustración 5 se representa el arreglo de modelos y la forma de los vectores en la serie de tiempo.

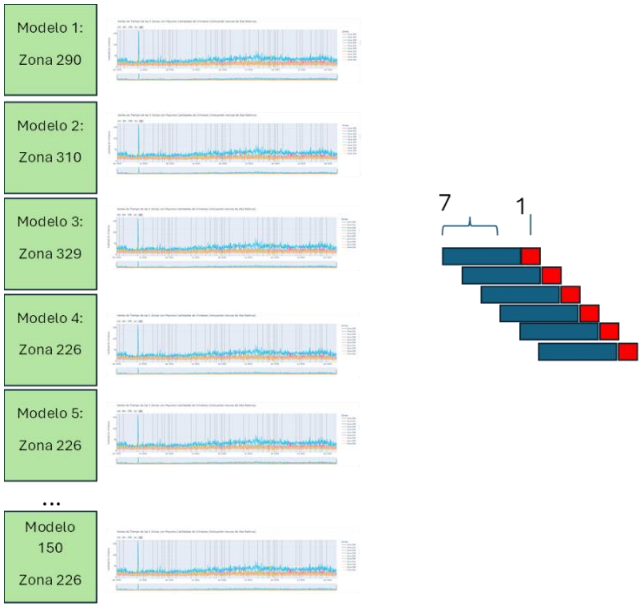


Ilustración 5 Esquema de la modelación realizada

```
# creación del arreglo de modelos de predicción
for i, zona in enumerate(zonas):
    print(f"Entrenando para la zona: {zona}")
    zona_ds = data_to_train[data_to_train['Zona'] == zona].set_index('Fecha')
    # Creación de conjuntos de entrenamiento, validación y test
    N = zona_ds.shape[0]
    Ntrain = int(0.8 * N)
    Nval = int(0.1 * N)
    Ntst = N - Ntrain - Nval
    train_ds = zona_ds.iloc[:Ntrain]
    test_ds = zona_ds.iloc[Ntrain:]
    train_ds.index = pd.to_datetime(train_ds.index)
    test_ds.index = pd.to_datetime(test_ds.index)
    zona_ds = zona_ds.resample("D").sum()
    train_ds = train_ds.resample("D").sum()
    test_ds = test_ds.resample("D").sum()
    # Creación del modelo predictor para la zona
    forecaster = ForecasterAutoreg(regressor=RandomForestRegressor(n_estimators=10), lags=7)
    forecaster.fit(y=train_ds['Cantidad de Crimenes'], exog=train_ds[variables_exogenas])
    print(f"Forecaster para la zona {zona} entrenado")
    # Guardar el forecaster en el diccionario
    forecasters[zona] = forecaster
    forecaster = forecasters[zona]
    # Realizar el backtesting
    metricas, predicciones = backtesting_forecaster(forecaster=forecaster, y=zona_ds['Cantidad de Crimenes'],
        exog=zona_ds[variables_exogenas],  initial_train_size=Ntrain,  steps=7,  metric='mean_absolute_error',
        refit=True, verbose=False )
    print(f"Backtesting para la zona {zona} completado")
    print(f"Zona {zona} - Mean Absolute Error: {metricas}")
    # Almacenamiento en memoria de Conjuntos de entrenamiento, predicción y metrica
    zona_ds['pred'] = predicciones
    predicciones_array[zona] = predicciones
    entrenamiento_array[zona] = zona_ds
    metricas_array.append(metricas)
```

Ilustración 6 Código fuente que crea el arreglo de modelos predictivos

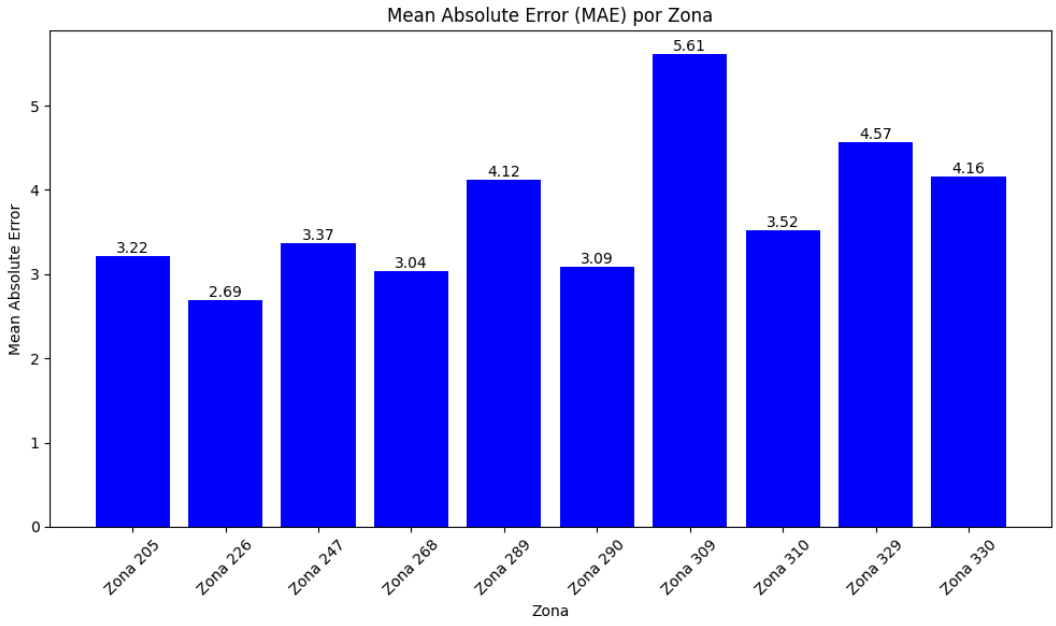
Los resultados de los modelos tuvieron una alta precisión, de 0.99, es decir que una gran proporción de las predicciones coinciden con los datos reales. Aunque la precisión es una métrica útil no siempre es suficiente para evaluar el modelo, por esta razón también se calculó la medida de error absoluto medio que también evidenció que el modelo tiene una buena precisión. En la Ilustración 7 se presentan las métricas agregadas de los 150 modelos predictivos generados y en la Gráfica 11 se presentan algunos errores absolutos de los modelos para algunas zonas.

Total Datos	Entrenamiento	Test	Predicción	Delta	Precisión
866.928	693.542	173.386	171.770	1.616	0,990679755

Error Absoluto Medio (MAE)	=	[1 , 5.88]
Precisión acumulada	=	99,06%

Ilustración 7 Métricas agregadas de los 150 modelos predictivos generados

Gráfica 11 Error Absoluto Medio para la predicción de crimen de 10 zonas con más delitos



La zona 226 tiene el menor MAE con un valor de 2.69, sugiriendo que el modelo de predicción es más preciso en esta zona en comparación con las otras. La Zona 309 presenta el mayor MAE con un valor de 5.61, indicando que el modelo de predicción tiene mayores errores en esta zona. Zonas como la 226 y 268 tienen mejores rendimientos, lo que podría estar relacionado con características específicas de esas áreas, como patrones de crimen más consistentes o datos de mayor calidad. La alta MAE en zonas como la 309 y 329 puede indicar una mayor dificultad para el modelo de capturar los patrones de crimen en estas áreas. Esto podría deberse a una mayor variabilidad en los datos de crimen, cambios rápidos en los patrones de crimen, o datos insuficientes o de baja calidad.

VISUALIZACIÓN

La visualización de los datos se realizó utilizando la herramienta Power Bi de Microsoft. En la Ilustración 8 se presenta la pantalla inicial del proyecto. La visualización contiene una pantalla de procesamiento donde se presenta gráficamente la información de la base de datos. El geo visor donde se aprecian algunas estadísticas resumen de la base de datos y la serie de tiempo. La predicción con el error del modelo se presenta en la pantalla de predicción. Adicionalmente se creó una pantalla para presentar los datos de Colombia

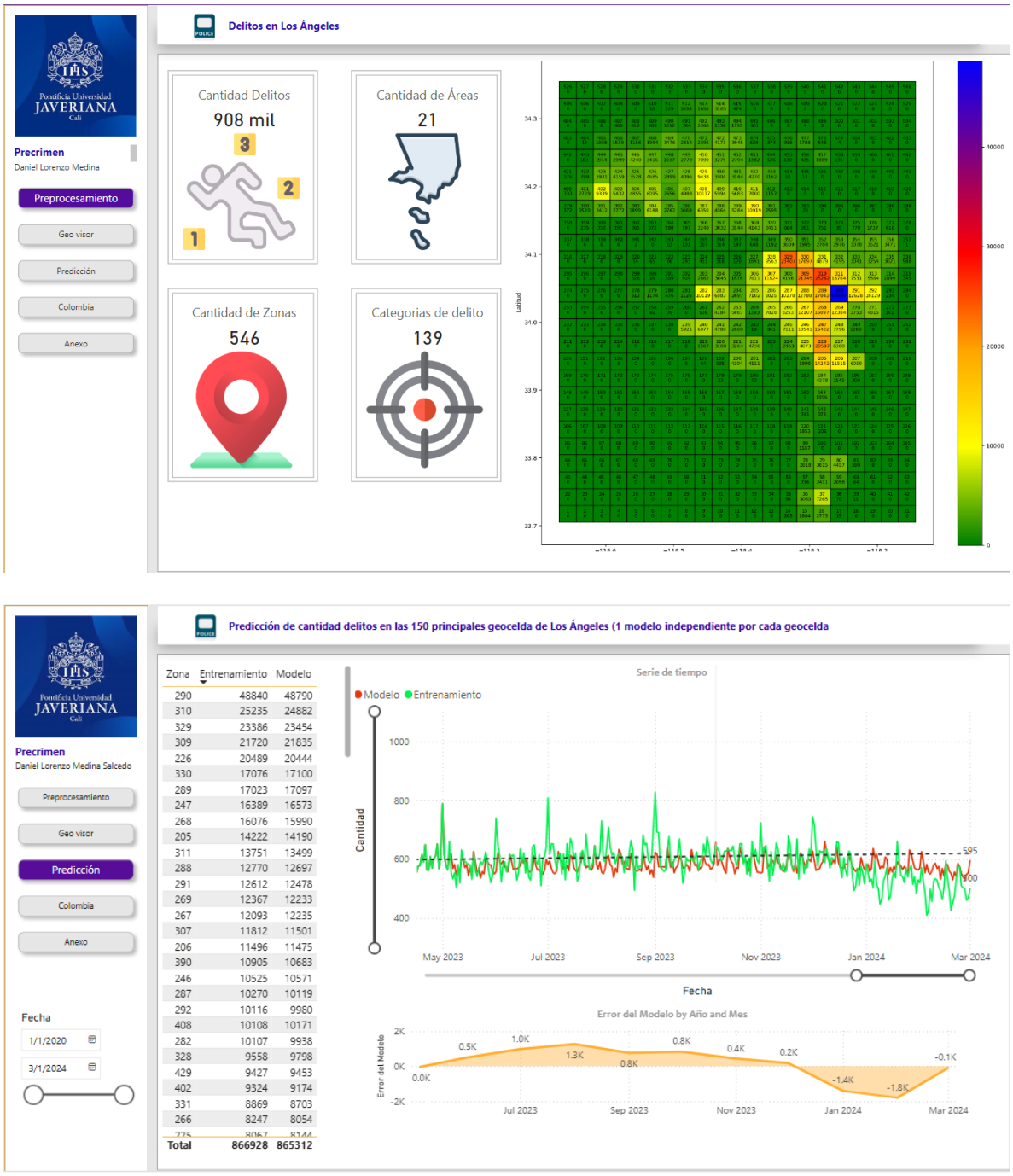


Ilustración 8. Página de visualización de la herramienta "Precrimen"

También se desarrolló una página de visualización en la dirección https://cuanticore.com/precrimen/. Esta última permite mayor interacción con los mapas.

5.3 OBJETIVO 3. APROPIAR CONOCIMIENTOS PRÁCTICOS DE GESTIÓN DE DATOS, CLASIFICACIÓN, VISUALIZACIÓN Y MODELOS DE PREDICCIÓN.

- En el presente proyecto se profundizo en elementos clave de las ciencias de datos tales como predicción de series de tiempo, visualización de datos con herramientas libres y de pago, gestión de datos espaciales y optimización de modelos de predicción de eventos posicionados espacialmente,
- La apropiación del conocimiento en el manejo de grandes volúmenes de datos, conocidos comúnmente como big data, ha representado un desafío crucial y una oportunidad significativa durante el desarrollo de este trabajo. Dominar este conocimiento no solo implica la recolección y almacenamiento de enormes cantidades de datos, sino también la capacidad de analizarlos de manera efectiva para descubrir patrones, tendencias y conexiones que puedan traducirse en decisiones estratégicas. Este proceso requiere una comprensión profunda de herramientas analíticas y algoritmos avanzados, así como una sólida habilidad para interpretar los resultados obtenidos. Al integrar esta inteligencia en sus operaciones, las empresas y entidades pueden optimizar procesos, anticipar necesidades del mercado y responder más ágilmente a los cambios del entorno, lo cual es esencial para mantener la competitividad y la innovación en un mercado cada vez más orientado hacia el análisis de datos.
- Adoptar y comprender los servicios en la nube no solo ha facilitado una infraestructura de TI más flexible y escalable, sino que también ha permite una colaboración más eficiente y un acceso seguro a datos y aplicaciones desde cualquier lugar. Esto implica entender las diversas opciones de servicio, como la infraestructura como servicio (IaaS), la plataforma como servicio (PaaS) y el software como servicio (SaaS), cada una ofreciendo distintos niveles de control, gestión y personalización. La correcta implementación de estos servicios puede conducir a una reducción significativa de costo y es un camino más claro hacia la innovación digital.
- La apropiación de conocimiento sobre métodos de entrenamiento en inteligencia artificial, tales como Random Forest, LTSM, SVM y KNN es esencial para los científicos de datos y desarrolladores de AI que buscamos implementar soluciones efectivas y eficientes. Cada uno de estos algoritmos tiene características únicas y aplicaciones específicas basadas en la naturaleza del problema y los datos disponibles. Por ejemplo, Random Forest es muy efectivo para manejar grandes conjuntos de datos con numerosas variables de entrada y es menos propenso al sobreajuste. SVM es preferido para problemas de clasificación y regresión donde

la claridad de la marginación entre las clases es un factor crucial. Por su parte, KNN es un método simple y fácil de implementar, útil para sistemas de recomendación y clasificación basada en la proximidad de los datos. Entender y saber cuándo aplicar cada uno de estos métodos permite a los profesionales maximizar la precisión y la eficiencia de sus modelos predictivos, adaptando las soluciones a las necesidades específicas del contexto y los datos con los que trabajan.

- La apropiación de conocimientos prácticos en gestión de datos, clasificación, visualización y modelos de predicción es fundamental para abordar de manera efectiva el análisis del crimen. Al integrar estas habilidades, es posible desarrollar modelos predictivos robustos que permitan anticipar eventos delictivos con mayor precisión. Estos modelos, basados en datos históricos y patrones identificables, ofrecen a las fuerzas de seguridad y a los responsables de políticas públicas herramientas valiosas para la prevención del crimen y la asignación eficiente de recursos. Además, la capacidad de visualizar estos datos de manera clara y comprensible facilita la toma de decisiones informadas y el diseño de estrategias preventivas más efectivas. En última instancia, el dominio de estas competencias no solo contribuye a una comprensión más profunda del comportamiento delictivo, sino que también mejora la seguridad y el bienestar de las comunidades al permitir intervenciones más proactivas y basadas en evidencia.

6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1 CONCLUSIONES

- Fue posible predecir la cantidad de crímenes usando técnicas de machine learning con una precisión de 99%, apalancados en la estrategia de dividir el problema de la predicción a zonas más pequeñas lo cual simplificó el problema y elevo la métrica.
- La predicción de series de tiempo de delitos presenta mejores métricas cuando se agrupan los delitos en áreas relativamente pequeñas (aproximadamente 7 Km^2) lo cual permite estandarizar los eventos y aislar anomalías.
- El crimen es un conjunto de factores y hechos complejos que requieren esfuerzos mayores por las autoridades para ser medido. La multidimensional del delito permite que pueda ser analizado en perspectivas de víctima, victimario, tiempo, modo, lugar entre otros, sin embargo, los conjuntos de datos tienden a ser insuficientes para medir integralmente el fenómeno del crimen.
- Para la ejecución de este proyecto fue necesario un presupuesto de cómputo de 500 dólares consumidos principalmente en el entrenamiento de 150 modelos. Con esto se evidencia la importancia de los recursos de nube para entrenar modelos con millones de registros e impactar positivamente la seguridad de un país.
- El cálculo de las ubicación de las zonas de los delitos tardo aproximadamente 4 horas para 1 millón de registros por consiguiente el tratamiento de datos espaciales suponen un esfuerzo computacional relevante y requiere ser considerado en el preprocesamiento de conjuntos de datos similares.
- Los conjuntos de datos pueden presentar sesgos debido al posible sub-reporte o menor grado de denuncias de los delitos en un periodo especifico, es decir, que las víctimas no denunciaran el hecho por temor o por algún factor externo. Es así cómo se evidencia que los delitos sexuales han crecido significativamente los últimos 10 años, lo cual se podría explicar cómo mayor inseguridad o mayor grado de denuncia.
- Recientemente, los conjuntos de datos de Los Ángeles han agregado los delitos cibernéticos en sus registros, permiten agregar una nueva perspectiva y dimensión cómo lo es el ciberespacio.
- Para el caso de Colombia, los conjuntos de datos no cuentan con información georreferenciada lo cual limita el análisis de variables relevantes para los modelos. Sin embargo, podría coordinarse por parte de las autorizades publicarse el número de cuadrante

en el cual se presentó el evento y así permitir el uso del aprendizaje de máquina para el entrenamiento de modelos de predicción para la policía de Colombia.

6.2 TRABAJOS FUTUROS

- El presente Proyecto puede ser ampliado a la predicción de los delitos que van a ser cometidos en una zona específica, utilizando las covariable de género, origen étnico y edad para dar así brindar una información oportuna a las autoridades que les permita prevenir el delito.
- Ampliar el estudio y conjuntos de datos a los comportamientos y características de los victimarios o delincuentes y con ello enriquecer la capacidad predictiva de los modelos e implementar medidas preventivas más eficaces.
- Realizar una reentrenamiento continuo y ajuste de los modelos predictivos para adaptarse a los cambios en los patrones delictivos y así mejorar constantemente la precisión de las predicciones.
- Utilizar los modelos predictivos para desarrollar estrategias específicas de intervención en áreas con alta incidencia delictiva, optimizando la asignación de recursos y mejorando la seguridad pública.

7. BIBLIOGRAFIA

[1] "Countries with the Highest Criminality rate in the World - The Organized Crime Index." Accessed: Apr. 08, 2024. [Online]. Available: https://ocindex.net/

[2] M. Buvinic, A. Morrison, and M. B. Orlando, "Violencia, crimen y desarrollo social en América Latina y el Caribe," *Papeles de población*, vol. 11, no. 43, pp. 167–214, Mar. 2005.

[3] A. C. Poveda, "Economic Development, Inequality and Poverty: An Analysis of Urban Violence in Colombia," *Oxford Development Studies*, vol. 39, no. 4, pp. 453–468, Dec. 2011, doi: 10.1080/13600818.2011.620085.

[4] G. Khanna, C. Medina, A. Nyshadham, C. Posso, and J. Tamayo, "Job Loss, Credit, and Crime in Colombia," *American Economic Review: Insights*, vol. 3, no. 1, pp. 97–114, Mar. 2021, doi: 10.1257/aeri.20190547.

[5] F. Dakalbab, M. Abu Talib, O. Abu Waraga, A. Bou Nassif, S. Abbas, and Q. Nasir, "Artificial intelligence & crime prediction: A systematic literature review," *Social Sciences & Humanities Open*, vol. 6, no. 1, p. 100342, Jan. 2022, doi: 10.1016/j.ssaho.2022.100342.

[6] Y. Liu, Z. Cheng, and X. Li, "How to prevent and control community risks?Identifying community burglary risk hotspots based on time-space characteristics," *Journal of Safety Science and Resilience*, vol. 4, no. 2, pp. 130–138, Jun. 2023, doi: 10.1016/j.jnlssr.2022.12.004.

[7] R. Arietti, "Do real-time crime centers improve case clearance? An examination of Chicago's strategic decision support centers," *Journal of Criminal Justice*, vol. 90, p. 102145, Jan. 2024, doi: 10.1016/j.jcrimjus.2023.102145.

[8] O. Huertas Díaz, "Durkheim: la perspectiva funcionalista del delito en la criminología," *Criminalidad*, vol. 51, no. 2, pp. 103–116, 2009.

[9] "Corte Constitucional de Colombia," Corte Constitucional de Colombia | Guardián de la Constitución. Accessed: Apr. 12, 2024. [Online]. Available: https://www.corteconstitucional.gov.co/

[10] RAE, "Definición de crimen - Diccionario panhispánico del español jurídico - RAE," Diccionario panhispánico del español jurídico - Real Academia Española. Accessed: Apr. 12, 2024. [Online]. Available: https://dpej.rae.es/lema/crimen

[11] S. González Andrade, "Criminalidad y crecimiento económico regional en México," *Frontera norte*, vol. 26, no. 51, pp. 75–111, Jun. 2014.

[12] H. Capel and H. C. Sáez, *Geografía Humana y Ciencias Sociales*. Editorial Montesinos, 1985.

[13] D. Salafranca Barreda and M. Rodríguez Herrera, "Sistemas de información geográfica aplicados a la investigación policial," in *Tecnologías de la información para nuevas formas de ver el territorio: XVI Congreso Nacional de Tecnologías de la Información Geográfica, 2014, ISBN 978-84-940784-4-6, págs. 721-736*, Universidad de Alicante / Universitat d'Alacant, 2014, pp. 721–736. Accessed: Apr. 12, 2024. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=5431178

[14] W. Moon, "Geographic Information Systems and Science (3rd Edition) by P. A. Longley, M. F. Goodchild, D. J. Maguire and D. W. Rhind (Book Review)," *The Leading Edge (Society of Exploration Geophysicists)*, vol. 31, pp. 975–976, Aug. 2012.

[15] D. O'Sullivan and D. Unwin, *Geographic Information Analysis*. John Wiley & Sons, 2003.

[16] K. Harries, *Mapping crime: principle and practice / Keith Harries.* in Research report. U.S. Dept. of Justice, Office of Justice Programs, National Institute of Justice, 1999.

[17] E. J. Ayos, "Prevención del delito y teorías criminológicas: tres problematizaciones sobre el presente," *Estudios Socio-Jurídicos*, vol. 16, no. 2, pp. 265–312, Dec. 2014, doi: 10.12804/esj16.02.2014.09.

[18] F. J. C. Toledo, A. B. G. Bellvís, and D. B. Gil, *La Criminología que viene: Resultados del I Encuentro de Jóvenes Investigadores en Criminología*. Red Española de Jóvenes

Investigadores en Criminología, 2019. Accessed: Apr. 12, 2024. [Online]. Available: https://dialnet.unirioja.es/servlet/libro?codigo=745952

[19] J. M. Helm *et al.*, "Machine Learning and Artificial Intelligence: Definitions, Applications, and Future Directions," *Curr Rev Musculoskelet Med*, vol. 13, no. 1, pp. 69–76, Feb. 2020, doi: 10.1007/s12178-020-09600-8.

[20] G. Rebala, A. Ravi, and S. Churiwala, "Machine Learning Definition and Basics," in *An Introduction to Machine Learning*, G. Rebala, A. Ravi, and S. Churiwala, Eds., Cham: Springer International Publishing, 2019, pp. 1–17. doi: 10.1007/978-3-030-15729-6_1.

[21] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[22] J. Brownlee, "Difference Between Algorithm and Model in Machine Learning," MachineLearningMastery.com. Accessed: Jun. 28, 2024. [Online]. Available: https://machinelearningmastery.com/difference-between-algorithm-and-model-in-machine-learning/

[23] "What is Feature Engineering? Definition and FAQs | HEAVY.AI." Accessed: Jun. 28, 2024. [Online]. Available: https://www.heavy.ai/technical-glossary/feature-engineering

[24] S. Guo and Y. Wang, "Investigating predictors of juvenile traditional and/or cyber offense using machine learning by constructing a decision support system," *Computers in Human Behavior*, vol. 152, p. 108079, Mar. 2024, doi: 10.1016/j.chb.2023.108079.

[25] "Ministerio del Interior | Sistema VioGén." Accessed: Apr. 12, 2024. [Online]. Available: https://www.interior.gob.es/opencms/ca/servicios-al-ciudadano/violencia-contra-la-mujer/sistema-viogen/

[26] Á. González-Prieto, A. Brú, J. C. Nuño, and J. L. González-Álvarez, "Hybrid machine learning methods for risk assessment in gender-based crime," *Knowledge-Based Systems*, vol. 260, p. 110130, Jan. 2023, doi: 10.1016/j.knosys.2022.110130.

[27] A. Yunus and J. Loo, "London street crime analysis and prediction using crowdsourced dataset," *Journal of Computational Mathematics and Data Science*, vol. 10, p. 100089, Mar. 2024, doi: 10.1016/j.jcmds.2023.100089.

[28] C. Fontes, E. Hohma, C. C. Corrigan, and C. Lütge, "AI-powered public surveillance systems: why we (might) need them and how we want them," *Technology in Society*, vol. 71, p. 102137, Nov. 2022, doi: 10.1016/j.techsoc.2022.102137.

[29] "Design of a real-time crime monitoring system using deep learning techniques - ScienceDirect." Accessed: Apr. 12, 2024. [Online]. Available: https://www-sciencedirect-com.bd.univalle.edu.co/science/article/pii/S2667305323001369

[30] Creative Commons, "CC0 1.0 Legal Code." Accessed: Jun. 27, 2024. [Online]. Available: https://creativecommons.org/publicdomain/zero/1.0/legalcode#copyright

[31] "Folium — Folium 0.16.1.dev76+g2921126e documentation." Accessed: Jun. 27, 2024. [Online]. Available: https://python-visualization.github.io/folium/latest/

[32] J. Amat Rodrigo and J. Escobar Ortiz, "skforecast." Zenodo, May 20, 2024. doi: 10.5281/zenodo.11222585.

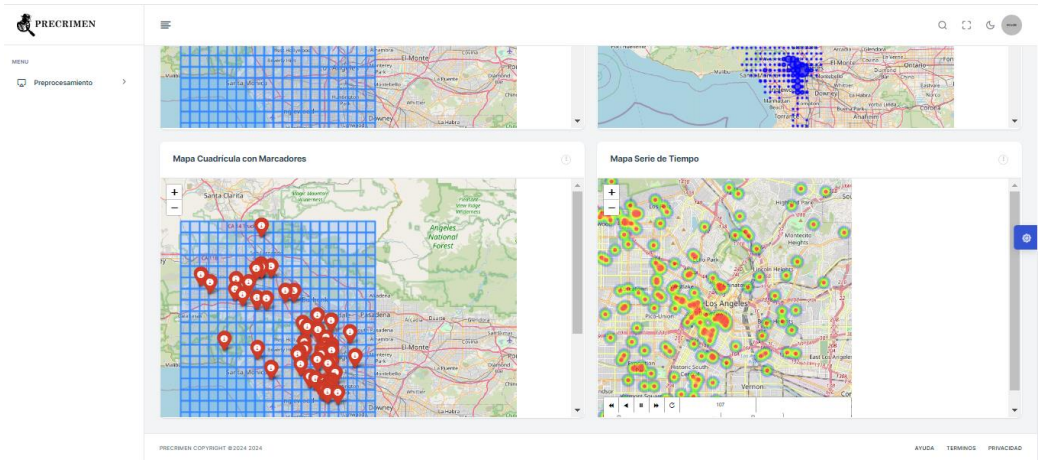
8. ANEXO 1- RESULTADOS Y CODIGO FUENTE

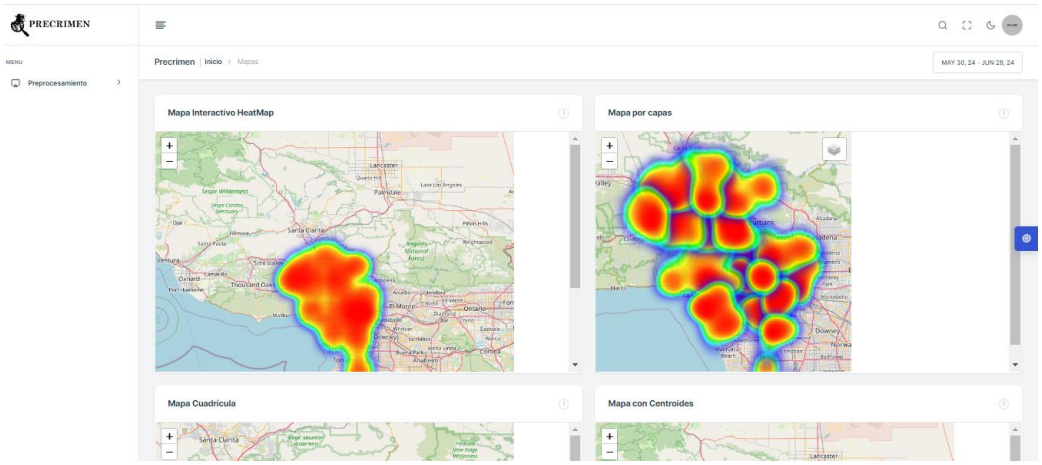
ANEXO. RESULTADOS Y CÓDIGO FUENTE

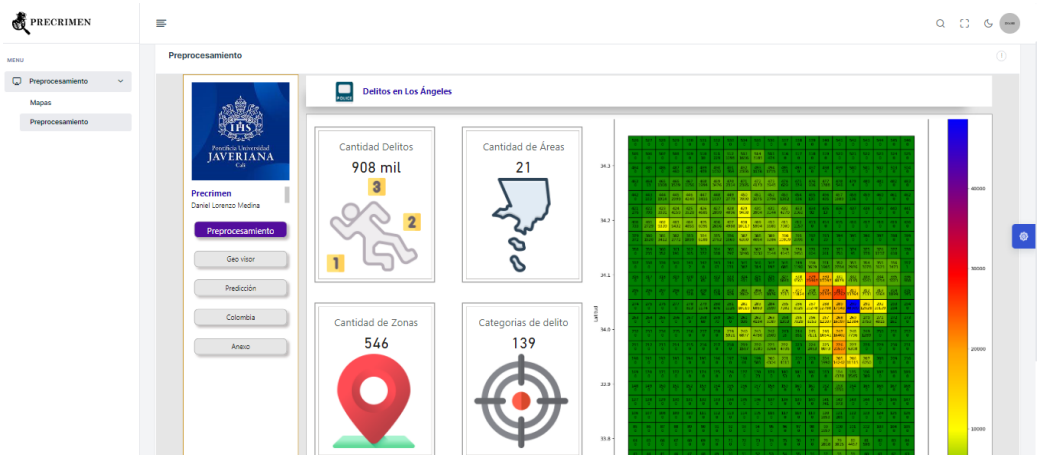
Nombre: Panel de control Herramienta de Precrimen

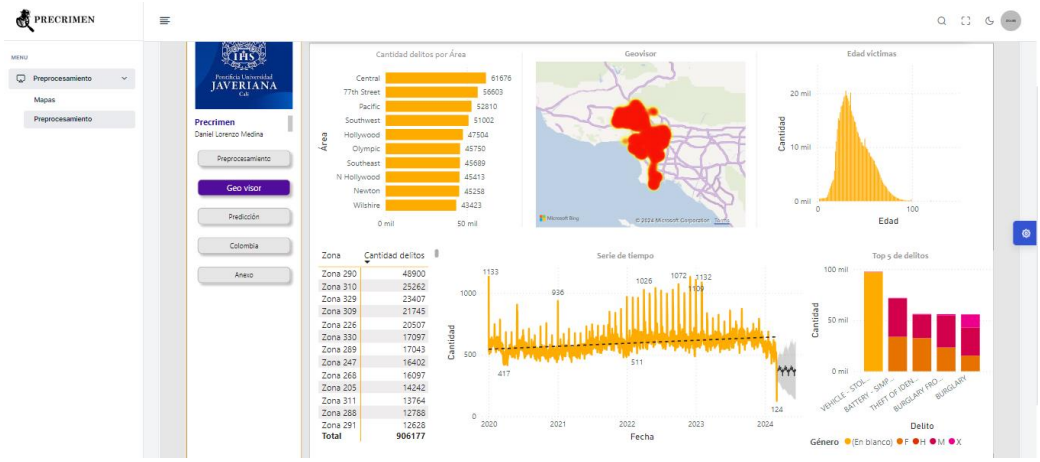
Enlace web sitio html 5: https://cuanticore.com/precrimen/

Enlace web de Power BI:
https://app.powerbi.com/view?r=eyJrIjoiOTM2MTViOTktYzFlOC00ZWQ3LTk5YjAtZjNlMmRhY2Q5ZTAwIiwidCI6IjRhZGU0NTlhLWRmNzQtNDdhYy04ODQ2LTdmYmQwZTZhYWQxYiIsImMiOjR9&pageName=4760a1cb2ecc17e6c25a






PRECRIMEN
MENU
Preprocesamiento
Mapas
Preprocesamiento
Preprocesamiento
Delitos en Los Ángeles
Precrimen
Daniel Lorenzo Medina
Preprocesamiento
Geo visor
Predicción
Colombia
Anexo
Cantidad Delitos
908 mil
Cantidad de Áreas
21
Cantidad de Zonas
546
Categorías de delito
139


PRECRIMEN
Precrimen
Daniel Lorenzo Medina
Preprocesamiento
Geo visor
Predicción
Colombia
Anexo
Cantidad delitos por Área
Geovisor
Edad víctimas
Serie de tiempo
Top 5 de delitos
Zona
Cantidad delitos
Total
906177
Género


PRECRIMEN
Precrimen
Daniel Lorenzo Medina
Preprocesamiento
Geo visor
Predicción
Colombia
Anexo
Fecha
01/01/2020
01/03/2024
Zona
Entrenamiento
Modelo
Total
866928
865312
Serie de tiempo
Modelo
Entrenamiento
Fecha
Error del Modelo por Año y Mes

Pontificia Universidad JAVERIANA Cali

Precrimen

Daniel Lorenzo Medina

- Preprocesamiento
- Geo visor
- Predicción
- Colombia
- Anexo

Delitos en Los Ángeles

Cantidad Delitos

908 mil

Cantidad de Áreas

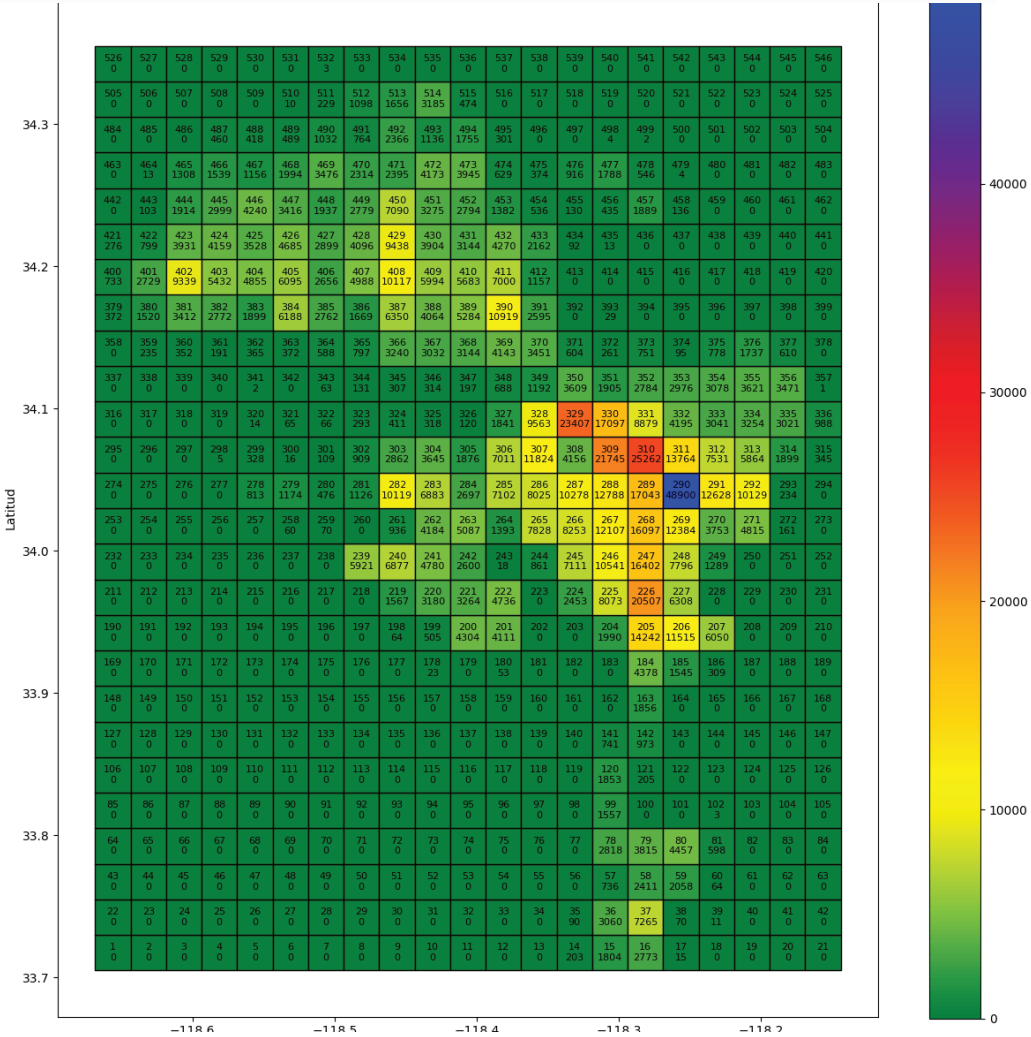
21

Cantidad de Zonas

546

Categorias de delito

139

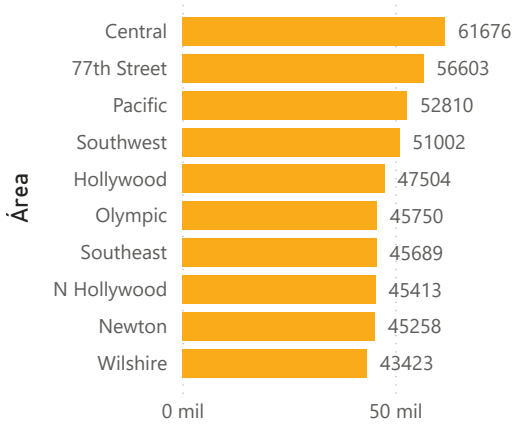


Precrimen
Daniel Lorenzo Medina

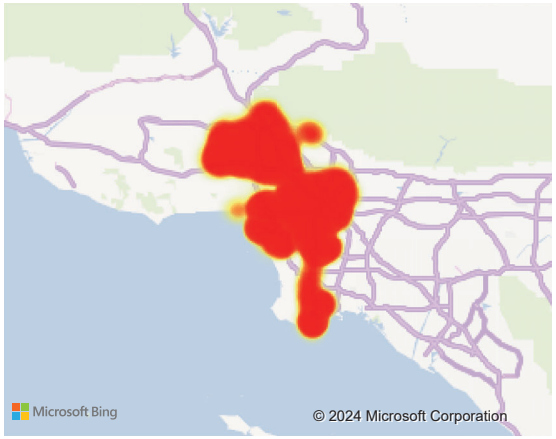
- Preprocesamiento
- Geo visor
- Predicción
- Colombia
- Anexo

Delitos en Los Ángeles

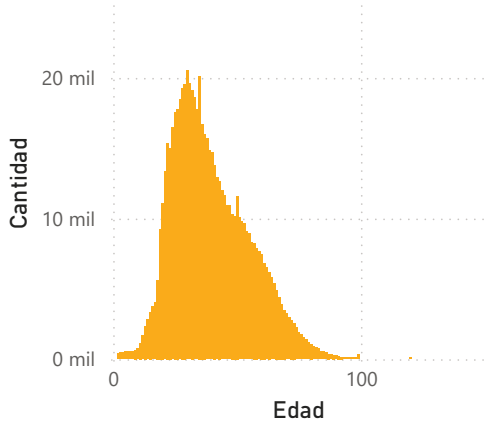
Cantidad delitos por Área



Geovisor

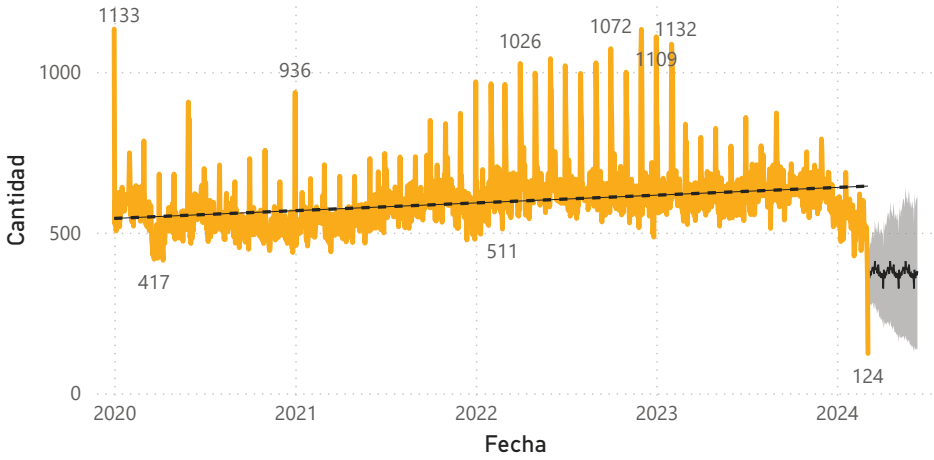


Edad victimas

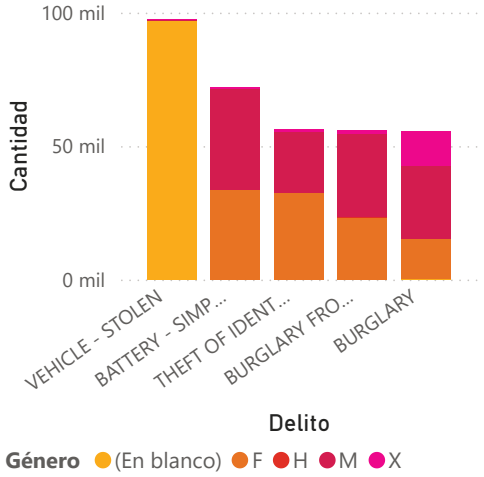


Zona	Cantidad delitos
Zona 290	48900
Zona 310	25262
Zona 329	23407
Zona 309	21745
Zona 226	20507
Zona 330	17097
Zona 289	17043
Zona 247	16402
Zona 268	16097
Zona 205	14242
Zona 311	13764
Zona 288	12788
Zona 291	12628
Total	**906177**

Serie de tiempo



Top 5 de delitos



Pontificia Universidad JAVERIANA Cali

Precrimen

Daniel Lorenzo Medina

- Preprocesamiento
- Geo visor
- Predicción
- Colombia
- Anexo

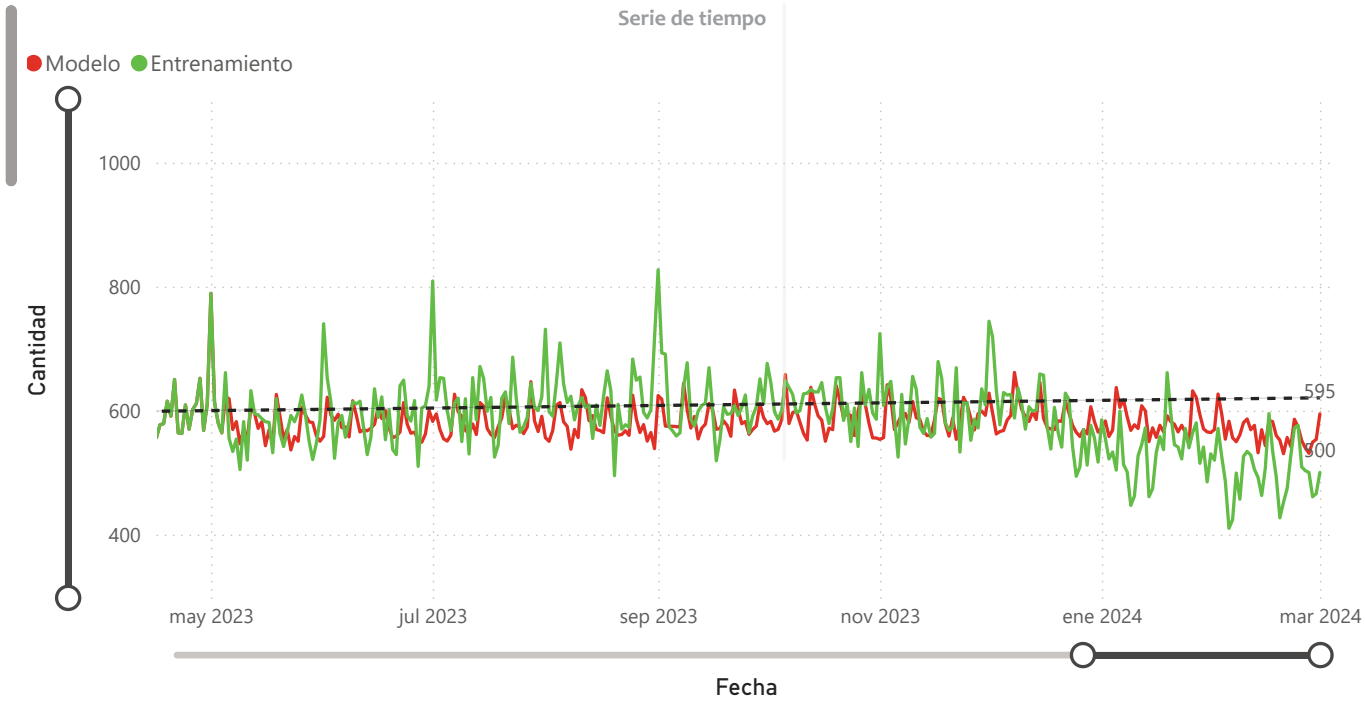
Fecha

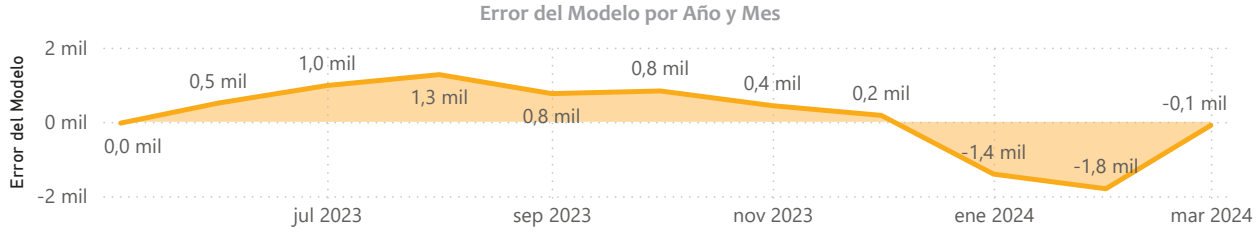
01/01/2020

01/03/2024

Predicción de cantidad delitos en las 150 principales geocelda de Los Ángeles (1 modelo independiente por cada geocelda

Zona	Entrenamiento	Modelo
290	48840	48790
310	25235	24882
329	23386	23454
309	21720	21835
226	20489	20444
330	17076	17100
289	17023	17097
247	16389	16573
268	16076	15990
205	14222	14190
311	13751	13499
288	12770	12697
291	12612	12478
269	12367	12233
267	12093	12235
307	11812	11501
206	11496	11475
390	10905	10683
246	10525	10571
287	10270	10119
292	10116	9980
408	10108	10171
282	10107	9938
328	9558	9798
429	9427	9453
402	9324	9174
331	8869	8703
Total	**866928**	**865312**



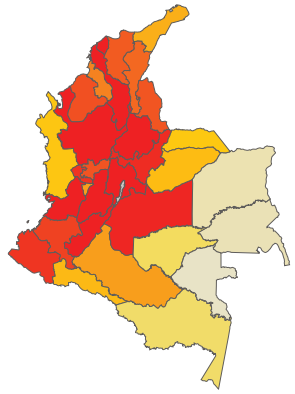


Precrimen
Daniel Lorenzo Medina

- Preprocesamiento
- Geo visor
- Predicción
- Colombia
- Anexo

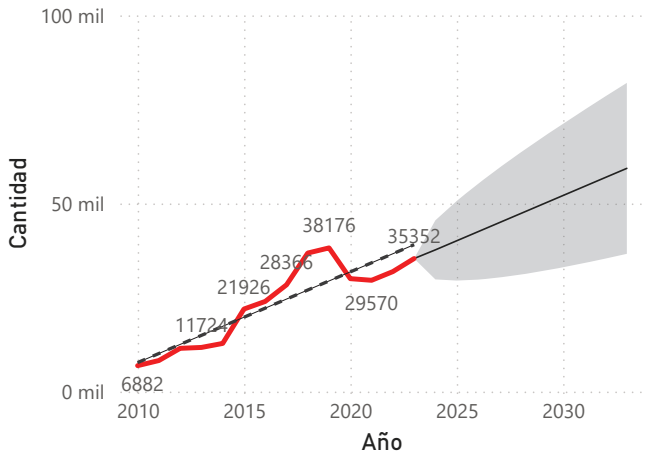
Delitos en Colombia con tendencia

Delitos sexuales

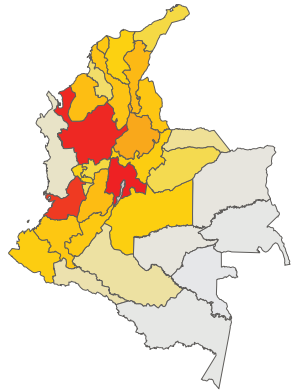


Departamento	Cantidad
CUNDINAMARCA	69610
ANTIOQUIA	39365
VALLE DEL CAUCA	32174
SANTANDER	20480
ATLANTICO	14138
HUILA	12407
TOLIMA	11897
BOLIVAR	11354
META	10321
BOYACA	9954
RISARALDA	9313
Total	**328609**

Serie de tiempo y pronósticos

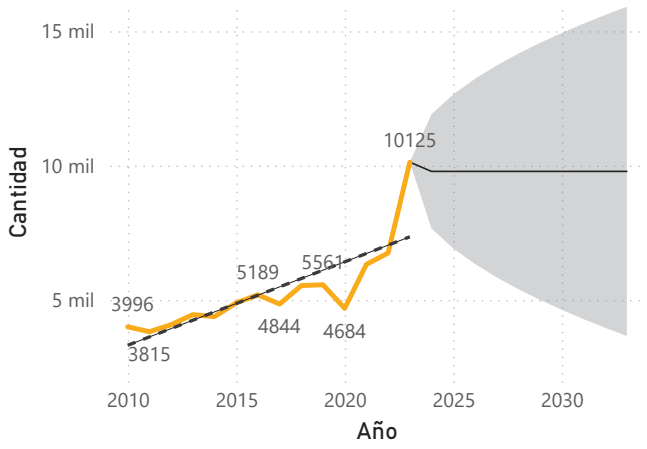


Homidicios por accidente de transito



Departamento	Cantidad
CUNDINAMARCA	11265
ANTIOQUIA	10218
VALLE	8900
SANTANDER	3850
CESAR	3102
TOLIMA	3012
CÓRDOBA	2909
BOLÍVAR	2651
NORTE DE SANTANDER	2565
CAUCA	2540
ATLÁNTICO	2509
HUILA	2427
NARIÑO	2247
Total	**75152**

Serie de tiempo y pronósticos



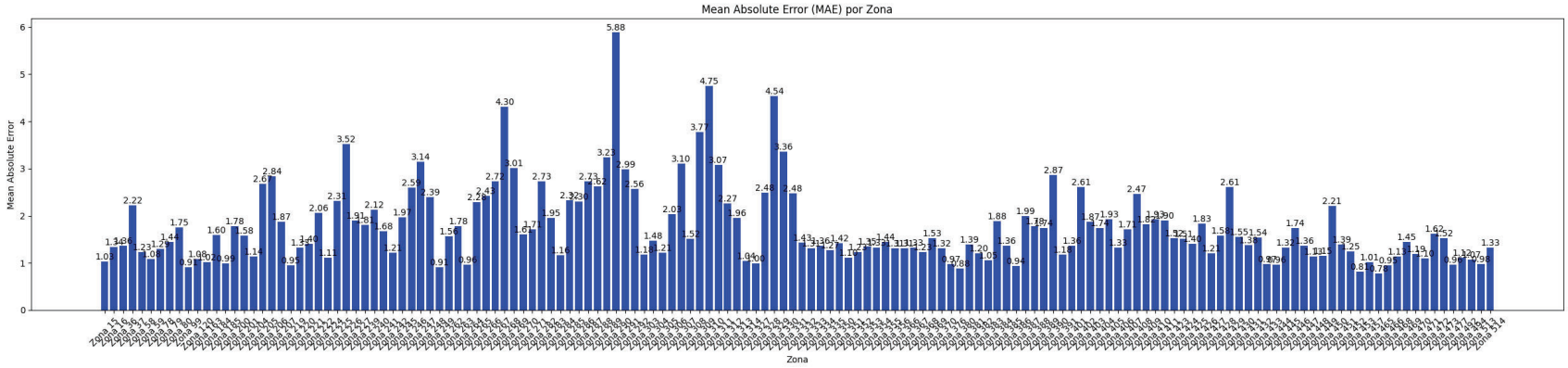
Pontificia Universidad JAVERIANA Cali

Precrimen

Daniel Lorenzo Medina

- Preprocesamiento
- Geo visor
- Predicción
- Colombia
- Anexo

Anexos



Precrimen

Daniel Lorenzo Medina Salcedo

Tutor: David Arango

Maestria en Ciencia de Datos

Universidad Javeriana de Cali

2024

Cargar de librerias

```
# Carga de librerías
import numpy as np
import pandas as pd
import sklearn as sk
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import datetime
import seaborn as sns
import matplotlib.style as style
import warnings
from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
from sklearn.metrics import f1_score
import folium
from folium.plugins import HeatMap

#%matplotlib inline
#sns.set(style='white', context='notebook', palette='deep')
#style.use('fivethirtyeight')
```

1. Obtención de datos

```
from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/My Drive/Precrimen/DataSets
```

```
Mounted at /content/drive
/content/drive/My Drive/Precrimen/DataSets
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
url_los_angeles='Crime_Data_from_2020_to_Present.csv'
data = pd.read_csv(url_los_angeles)
data['DATE OCC'] = pd.to_datetime(data['DATE OCC'])
data.head()
```

```
<ipython-input-4-f6b21f6065be>:3: UserWarning: Could not infer format, so each element will be parsed individual
  data['DATE OCC'] = pd.to_datetime(data['DATE OCC'])
```

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	...	Status	Status Desc	Crm Cd 1
0	190326475	03/01/2020 12:00:00 AM	2020-03-01	2130	7	Wilshire	784	1	510	VEHICLE - STOLEN	...	AA	Adult Arrest	510.0
1	200106753	02/09/2020 12:00:00 AM	2020-02-08	1800	1	Central	182	1	330	BURGLARY FROM VEHICLE	...	IC	Invest Cont	330.0
2	200320258	11/11/2020 12:00:00 AM	2020-11-04	1700	3	Southwest	356	1	480	BIKE - STOLEN	...	IC	Invest Cont	480.0
3	200907217	05/10/2023 12:00:00	2020-	2037	9	Van Nuys	964	1	343	SHOPLIFTING-GRAND THEFT		IC	Invest	343.0

Análisis Exploratorio de datos

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 910707 entries, 0 to 910706
Data columns (total 28 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   DR_NO           910707 non-null  int64
 1   Date Rptd       910707 non-null  object
 2   DATE OCC        910707 non-null  datetime64[ns]
 3   TIME OCC        910707 non-null  int64
 4   AREA            910707 non-null  int64
 5   AREA NAME       910707 non-null  object
 6   Rpt Dist No     910707 non-null  int64
 7   Part 1-2        910707 non-null  int64
 8   Crm Cd          910707 non-null  int64
 9   Crm Cd Desc     910707 non-null  object
 10  Mocodes         783696 non-null  object
 11  Vict Age        910707 non-null  int64
 12  Vict Sex        789672 non-null  object
 13  Vict Descent    789663 non-null  object
 14  Premis Cd       910697 non-null  float64
 15  Premis Desc     910153 non-null  object
 16  Weapon Used Cd  315247 non-null  float64
 17  Weapon Desc     315247 non-null  object
 18  Status          910707 non-null  object
 19  Status Desc     910707 non-null  object
 20  Crm Cd 1        910696 non-null  float64
 21  Crm Cd 2        66335 non-null   float64
 22  Crm Cd 3        2237 non-null    float64
 23  Crm Cd 4        64 non-null      float64
 24  LOCATION        910707 non-null  object
 25  Cross Street    143332 non-null  object
 26  LAT             910707 non-null  float64
 27  LON             910707 non-null  float64
dtypes: datetime64[ns](1), float64(8), int64(7), object(12)
memory usage: 194.5+ MB
```

```
data.shape
```

```
(910707, 28)
```

```
data.describe()
```

	DR_NO	DATE OCC	TIME OCC	AREA	Rpt Dist No	Part 1-2	Crm Cd
count	9.107070e+05	910707	910707.000000	910707.000000	910707.000000	910707.000000	910707.000000
mean	2.180575e+08	2022-02-23 09:23:03.686740736	1337.042061	10.698686	1116.307327	1.410830	500.809825
min	8.170000e+02	2020-01-01 00:00:00	1.000000	1.000000	101.000000	1.000000	110.000000
25%	2.104073e+08	2021-02-21 00:00:00	900.000000	5.000000	589.000000	1.000000	331.000000
50%	2.205108e+08	2022-03-16 00:00:00	1415.000000	11.000000	1141.000000	1.000000	442.000000
75%	2.304119e+08	2023-03-01 00:00:00	1900.000000	16.000000	1615.000000	2.000000	626.000000
max	2.499046e+08	2024-03-04 00:00:00	2359.000000	21.000000	2199.000000	2.000000	956.000000
std	1.191945e+07	NaN	652.903148	6.102210	610.237433	0.491985	207.606033

```
data.isnull().sum()
```

```
DR_NO                 0
Date Rptd             0
DATE OCC              0
TIME OCC              0
AREA                  0
AREA NAME             0
Rpt Dist No           0
Part 1-2              0
Crm Cd                0
Crm Cd Desc           0
Mocodes          127011
Vict Age              0
Vict Sex         121035
Vict Descent     121044
Premis Cd            10
Premis Desc         554
Weapon Used Cd   595460
Weapon Desc      595460
Status                0
Status Desc           0
Crm Cd 1             11
Crm Cd 2         844372
Crm Cd 3         908470
Crm Cd 4         910643
LOCATION              0
Cross Street     767375
LAT                   0
LON                   0
dtype: int64
```

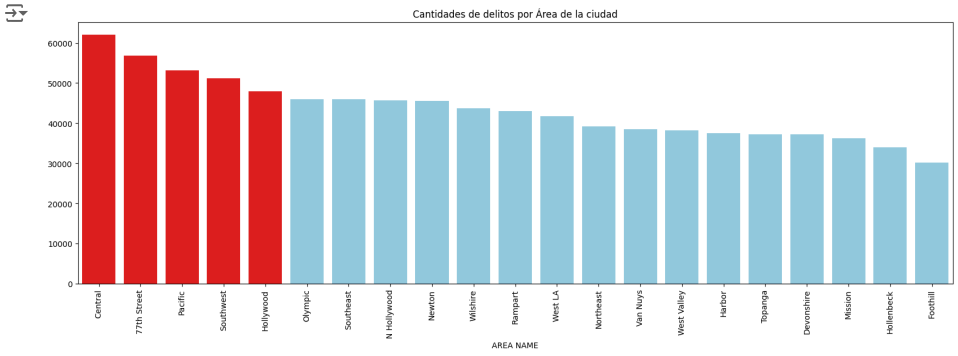
```
columnas = ['Date Rptd', 'DATE OCC', 'TIME OCC', 'AREA', 'AREA NAME', 'Vict Age', 'Vict Sex', 'Vict Descent', 'Crm Cd Desc', 'Crm Cd','LAT','LON']
df = data.copy()
df = df[columnas]
df.head()
```

	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Vict Age	Vict Sex	Vict Descent	Crm Cd Desc	Crm Cd	LAT	LON
0	03/01/2020 12:00:00 AM	2020-03-01	2130	7	Wilshire	0	M	O	VEHICLE - STOLEN	510	34.0375	-118.3506
1	02/09/2020 12:00:00 AM	2020-02-08	1800	1	Central	47	M	O	BURGLARY FROM VEHICLE	330	34.0444	-118.2628
2	11/11/2020 12:00:00 AM	2020-11-04	1700	3	Southwest	19	X	X	BIKE - STOLEN	480	34.0210	-118.3002

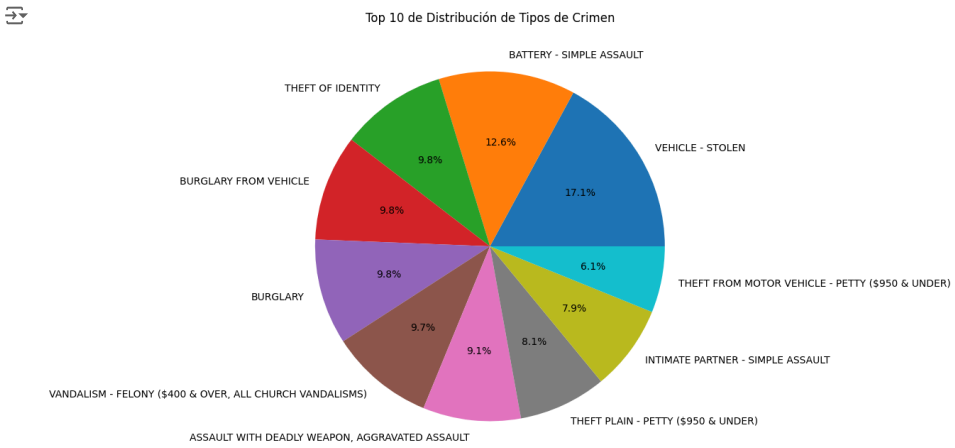
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 910707 entries, 0 to 910706
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Date Rptd     910707 non-null  object
 1   DATE OCC      910707 non-null  datetime64[ns]
 2   TIME OCC      910707 non-null  int64
 3   AREA          910707 non-null  int64
 4   AREA NAME     910707 non-null  object
 5   Vict Age      910707 non-null  int64
 6   Vict Sex      789672 non-null  object
 7   Vict Descent  789663 non-null  object
 8   Crm Cd Desc   910707 non-null  object
 9   Crm Cd        910707 non-null  int64
 10  LAT           910707 non-null  float64
 11  LON           910707 non-null  float64
dtypes: datetime64[ns](1), float64(2), int64(4), object(5)
memory usage: 83.4+ MB
```

```
# Crear el histograma usando countplot
fig = plt.figure(figsize=(20, 6))
plt.title('Cantidades de delitos por Área de la ciudad')

# Contar la cantidad de delitos por área
area_counts = df['AREA NAME'].value_counts()

# Identificar las 5 áreas con mayor cantidad de delitos
top_5_areas = area_counts.nlargest(5).index

# Crear una lista de colores para las barras
colors = ['red' if area in top_5_areas else 'skyblue' for area in area_counts.index]

# Crear el histograma con barras de colores
sns.barplot(x=area_counts.index, y=area_counts.values, palette=colors)

# Rotar las etiquetas del eje X
plt.xticks(rotation=90)

# Mostrar la gráfica
plt.show()
```



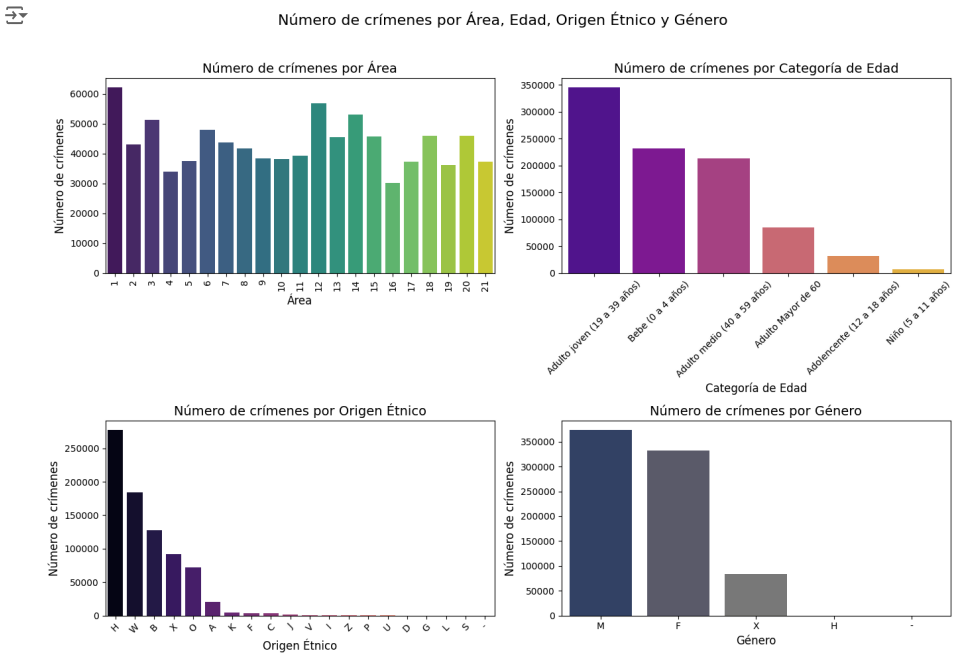
```
crime_type_counts = df['Crm Cd Desc'].value_counts()
top_10_crime_types = crime_type_counts.head(10)
plt.figure(figsize=(8, 8))
plt.pie(top_10_crime_types, labels=top_10_crime_types.index, autopct='%1.1f%%')
plt.title('Top 10 de Distribución de Tipos de Crimen')
plt.show()
```



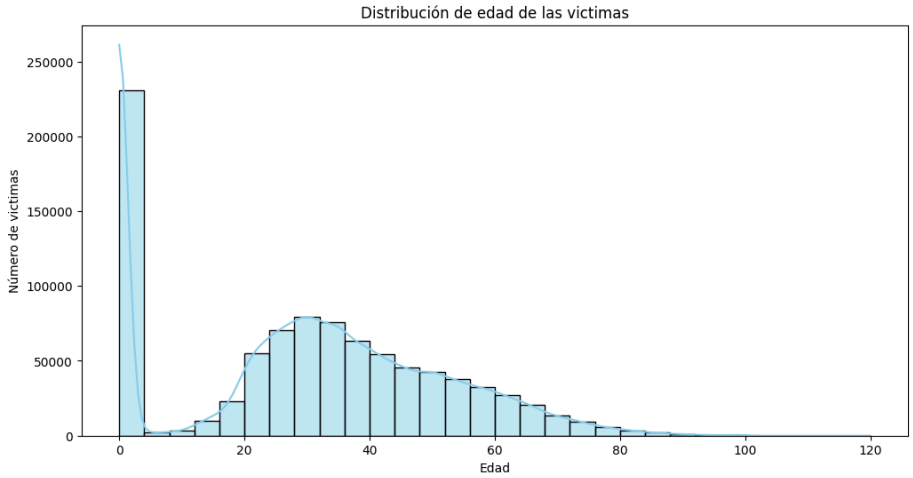
Haz doble clic (o pulsa Intro) para editar

```
time_field = 'DATE OCC' #time when the crime occurred
#time_field = 'Date Rptd' #time when the crime reported

df['Year'] = df[time_field].dt.year
df['Month'] = df[time_field].dt.month
df['Day'] = df[time_field].dt.day
df['Hour'] = df['TIME OCC'].apply(lambda x: x//100)


df['Vict Age'] = df['Vict Age'].where(df['Vict Age'] >= 0, other=pd.NA)
# Mean imputation for 'Vict Age'
mean_age = df['Vict Age'].mean()
df['Vict Age'] = df['Vict Age'].fillna(mean_age)

def GetAgeCategory(age):
    if age<=4: return 'Bebe (0 a 4 años)'
    if age<=12: return 'Niño (5 a 11 años)'
    elif age<=19: return 'Adolencente (12 a 18 años)'
    elif age<=39: return 'Adulto joven (19 a 39 años)'
    elif age<=59: return 'Adulto medio (40 a 59 años)'
    else: return 'Adulto Mayor de 60'

df['Vict Age Cat'] = df['Vict Age'].apply(lambda x: GetAgeCategory(x))


import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Crear datos de ejemplo

areas = ['Area_{}'.format(i) for i in range(1, 21)]
edades = ['Menor de 18', '18-25', '26-35', '36-45', '46-60', 'Mayor de 60']
origenes = ['Blanco', 'Negro', 'Latino', 'Asiático', 'Otro']
sexos = ['Masculino', 'Femenino']


# Crear la figura y los ejes
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(14, 10))

# Ajustes en los subgráficos
sns.barplot(x=df.groupby(['AREA'])['Day'].count().sort_values(ascending=False).index,
            y=df.groupby(['AREA'])['Day'].count().sort_values(ascending=False).values,
            palette='viridis', ax=axes[0,0])
axes[0,0].set_title('Número de crímenes por Área', fontsize=14)
axes[0,0].set_xlabel('Área', fontsize=12)
axes[0,0].set_ylabel('Número de crímenes', fontsize=12)
axes[0,0].tick_params(axis='x', rotation=90)

sns.barplot(x=df[df['Vict Age Cat'].notnull()].groupby(['Vict Age Cat'])['Day'].count().sort_values(ascending=False).index,
```

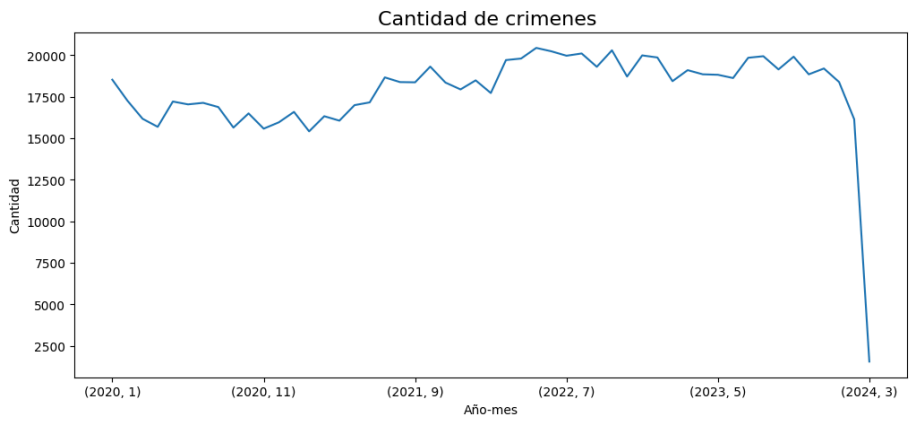
```
            y=df[df['Vict Age Cat'].notnull()].groupby(['Vict Age Cat'])['Day'].count().sort_values(ascending=False).values,
            palette='plasma', ax=axes[0,1])
axes[0,1].set_title('Número de crímenes por Categoría de Edad', fontsize=14)
axes[0,1].set_xlabel('Categoría de Edad', fontsize=12)
axes[0,1].set_ylabel('Número de crímenes', fontsize=12)
axes[0,1].tick_params(axis='x', rotation=45)

sns.barplot(x=df.groupby(['Vict Descent'])['Day'].count().sort_values(ascending=False).index,
            y=df.groupby(['Vict Descent'])['Day'].count().sort_values(ascending=False).values,
            palette='magma', ax=axes[1,0])
axes[1,0].set_title('Número de crímenes por Origen Étnico', fontsize=14)
axes[1,0].set_xlabel('Origen Étnico', fontsize=12)
axes[1,0].set_ylabel('Número de crímenes', fontsize=12)
axes[1,0].tick_params(axis='x', rotation=45)

sns.barplot(x=df.groupby(['Vict Sex'])['Day'].count().sort_values(ascending=False).index,
            y=df.groupby(['Vict Sex'])['Day'].count().sort_values(ascending=False).values,
            palette='cividis', ax=axes[1,1])
axes[1,1].set_title('Número de crímenes por Género', fontsize=14)
axes[1,1].set_xlabel('Género', fontsize=12)
axes[1,1].set_ylabel('Número de crímenes', fontsize=12)

# Título general
fig.suptitle('Número de crímenes por Área, Edad, Origen Étnico y Género', fontsize=16)

# Ajustar el diseño
plt.tight_layout(rect=[0, 0, 1, 0.96])

# Mostrar la gráfica
plt.show()
```




```
plt.figure(figsize=(12, 6))
sns.histplot(df['Vict Age'].dropna(), bins=30, kde=True, color='skyblue')
plt.title('Distribución de edad de las victimas')
plt.xlabel('Edad')
plt.ylabel('Número de victimas')
plt.show()
```

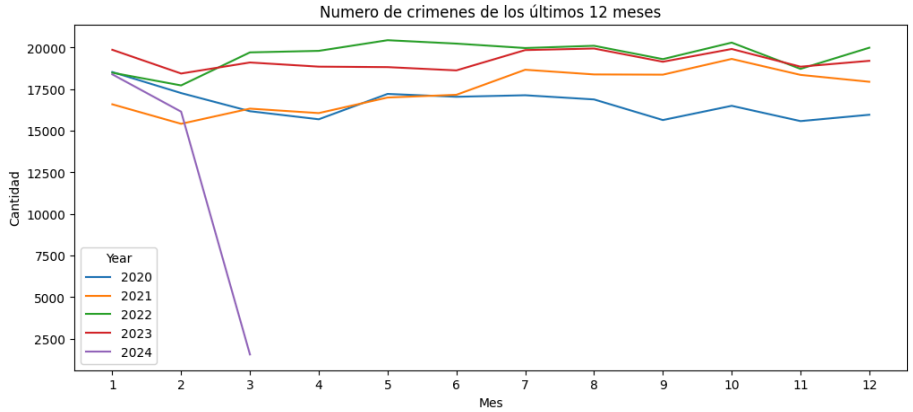



```
df.groupby(['Year', 'Month'])['Day'].count().plot(kind='line', figsize=(12, 5))
plt.ylabel('Cantidad')
plt.xlabel('Año-mes')
plt.title('Cantidad de crimenes ', loc='center', fontsize=16)
plt.show()
```



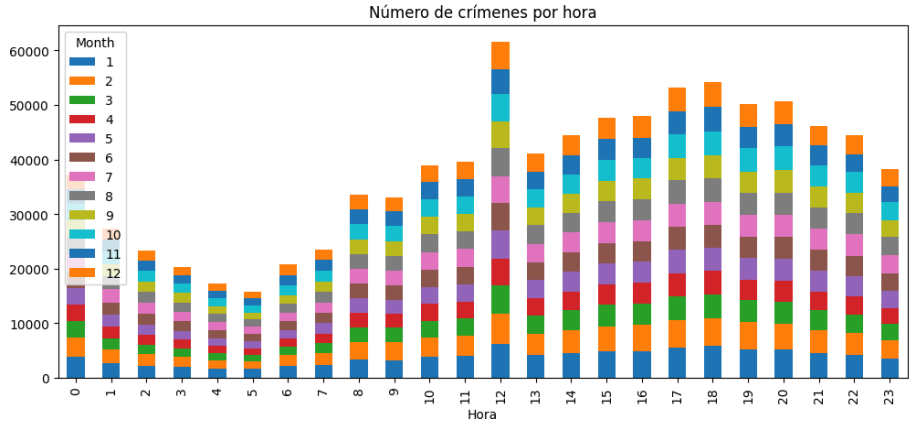

```
pltdata = df.groupby(['Month', 'Year'])['Day'].count().unstack()
pltdata.plot(kind='line', figsize=(12, 5), xticks=range(1, 13), title='Numero de crimenes de los últimos 12 meses')
plt.ylabel('Cantidad')
plt.xlabel('Mes')
```

```
Text(0.5, 0, 'Mes')
```



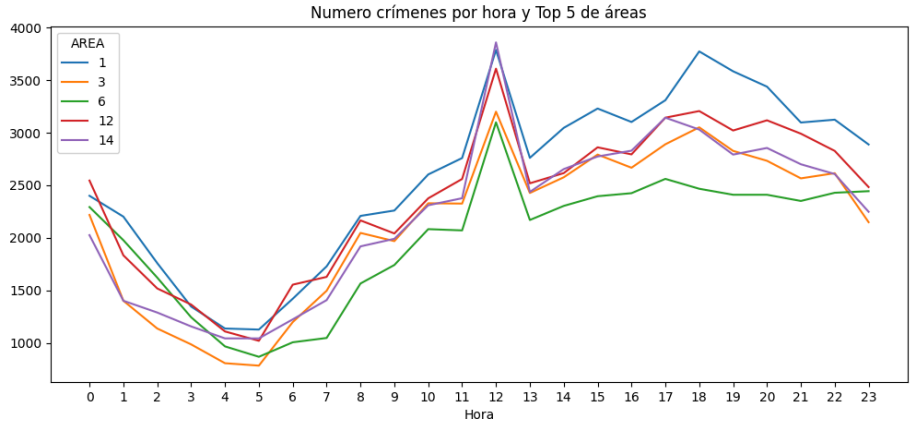
```
df.groupby(['Hour', 'Month'])['Day'].count().unstack().plot(kind='bar', figsize=(12, 5), stacked=True)
plt.title('Número de crímenes por hora')
plt.xlabel('Hora')
```

```
Text(0.5, 0, 'Hora')
```



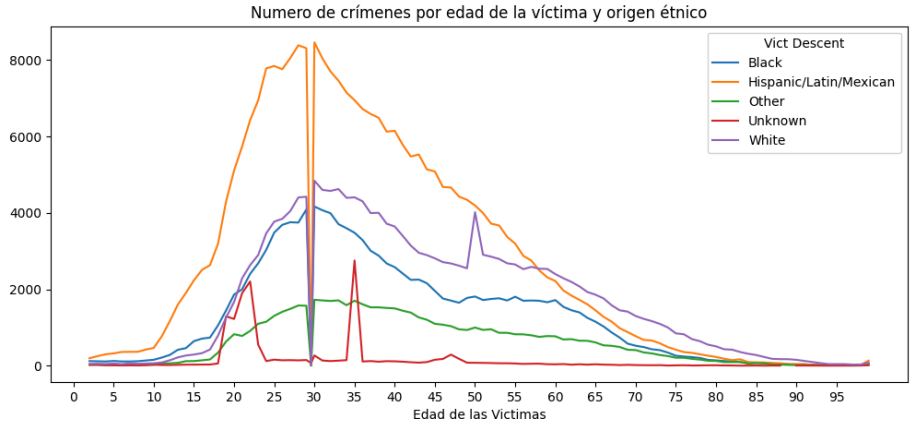

```
top5areas = df.groupby(['AREA'])['Day'].count().sort_values(ascending=False).index[:5]
pltdata = df[df['AREA'].isin(top5areas)]
pltdata.groupby(['Hour', 'AREA'])['Day'].count().unstack().plot(kind='line', figsize=(12, 5), xticks=range(0, 24))
plt.title('Numero crímenes por hora y Top 5 de áreas')
plt.xlabel('Hora')
```

```
Text(0.5, 0, 'Hora')
```



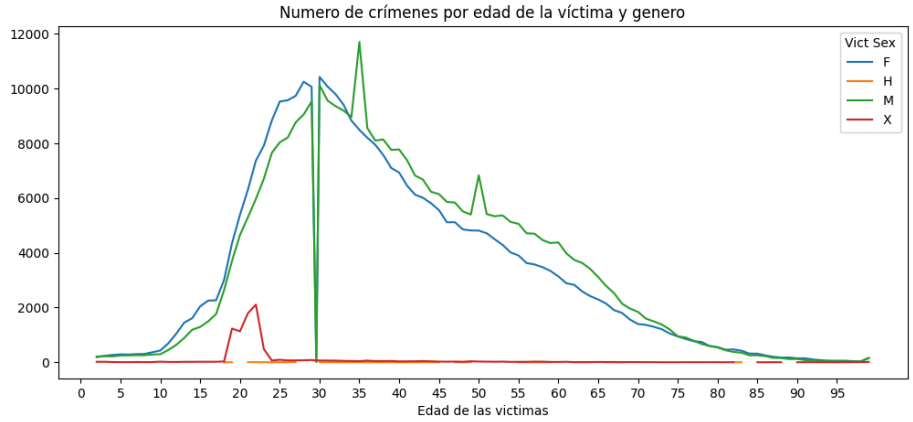

```
top5descents = df.groupby(['Vict Descent'])['Day'].count().sort_values(ascending=False).index[:5]
pltdata = df[(df['Vict Age']>0) & (df['Vict Descent'].isin(top5descents))].copy()
vddict = {'B':'Black', 'H':'Hispanic/Latin/Mexican', 'O':'Other', 'W':'White', 'X':'Unknown'}
pltdata['Vict Descent'] = pltdata['Vict Descent'].apply(lambda x: vddict[x])

plt_data = pltdata.groupby(['Vict Age', 'Vict Descent'])['Day'].count().unstack()
plt_data.plot(kind='line', figsize=(12, 5), xticks=range(0, 100, 5))
plt.title('Numero de crímenes por edad de la víctima y origen étnico')
plt.xlabel('Edad de las Victimas')
```

```
Text(0.5, 0, 'Edad de las Victimas')
```



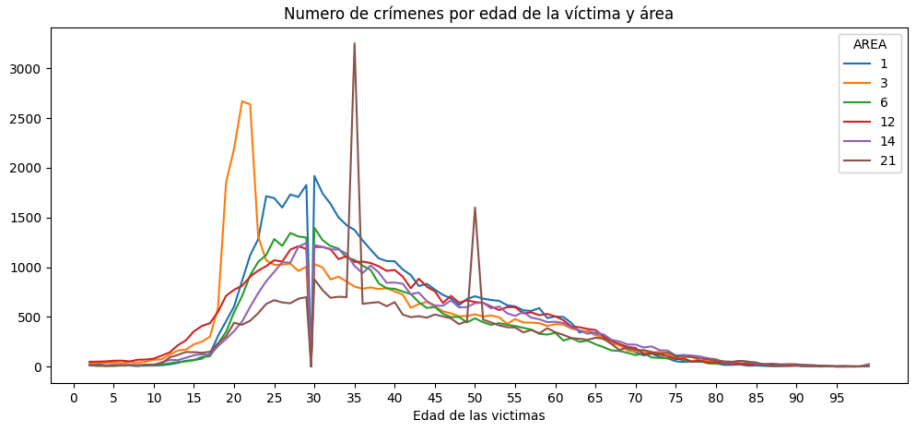

```
plt_data = df[(df['Vict Age']>0)]
plt_data = plt_data.groupby(['Vict Age', 'Vict Sex'])['Day'].count().unstack()
plt_data.plot(kind='line', figsize=(12, 5), xticks=range(0, 100, 5))
plt.title('Numero de crímenes por edad de la víctima y genero')
plt.xlabel('Edad de las victimas')
```

```
Text(0.5, 0, 'Edad de las victimas')
```



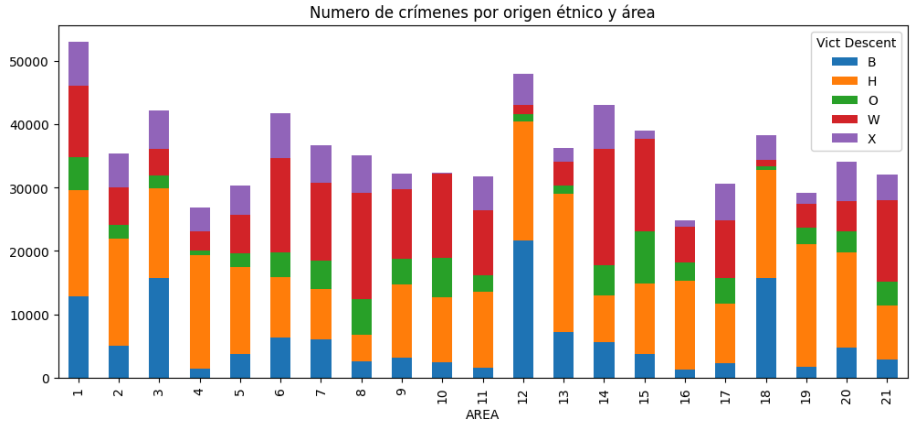

```
top5areas = list(df.groupby(['AREA'])['Day'].count().sort_values(ascending=False).index[:5])
plt_data = df[(df['Vict Age']>0) & (df['AREA'].isin(top5areas+[21]))]

plt_data = plt_data.groupby(['Vict Age', 'AREA'])['Day'].count().unstack()
plt_data.plot(kind='line', figsize=(12, 5), xticks=range(0, 100, 5))
plt.title('Numero de crímenes por edad de la víctima y área')
plt.xlabel('Edad de las victimas')
```

```
Text(0.5, 0, 'Edad de las victimas')
```



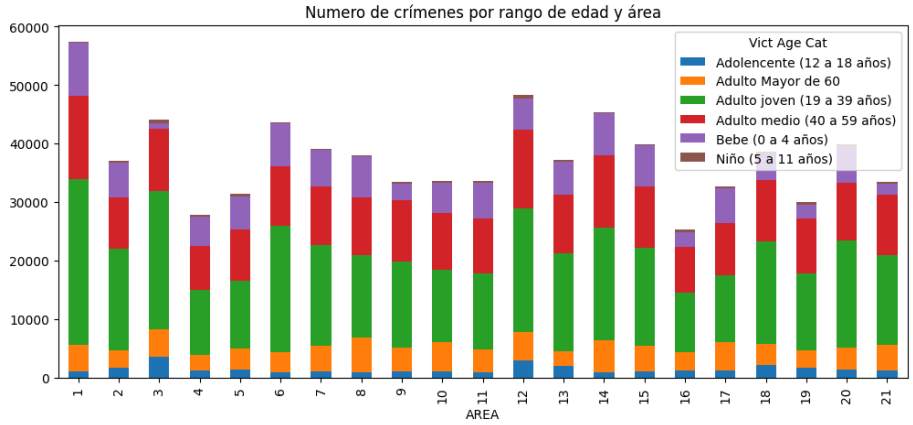
```
top5descents = df.groupby(['Vict Descent'])['Day'].count().sort_values(ascending=False).index[:5]
data = df[df['Vict Descent'].isin(top5descents)]
plt_data = data.groupby(['AREA', 'Vict Descent'])['Day'].count().unstack()
plt_data.plot(kind='bar', stacked=True, figsize=(12, 5))
#plt_data.plot(kind='line', figsize=(12, 5), xticks=range(1, 22))
plt.title('Numero de crímenes por origen étnico y área')
```

```
Text(0.5, 1.0, 'Numero de crímenes por origen étnico y área')
```




```
plt_data = df[df['Vict Descent'].notnull()].groupby(['AREA', 'Vict Age Cat'])['Day'].count().unstack()
plt_data.plot(kind='bar', stacked=True, figsize=(12, 5))
#plt_data.plot(kind='line', figsize=(12, 5), xticks=range(1, 22))
plt.title('Numero de crímenes por rango de edad y área')
```

```
Text(0.5, 1.0, 'Numero de crímenes por rango de edad y área')
```

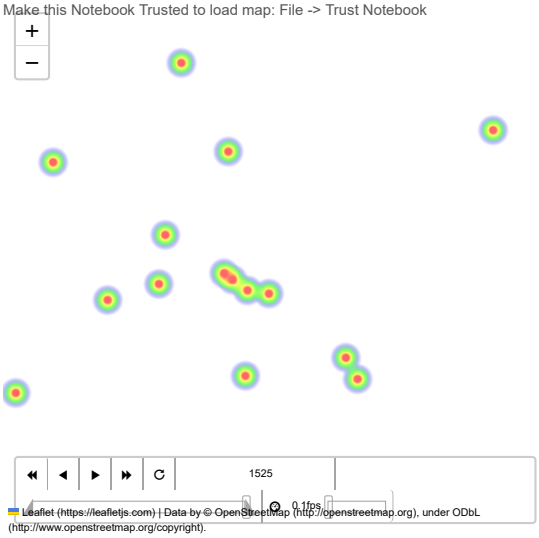


Análisis Espacial

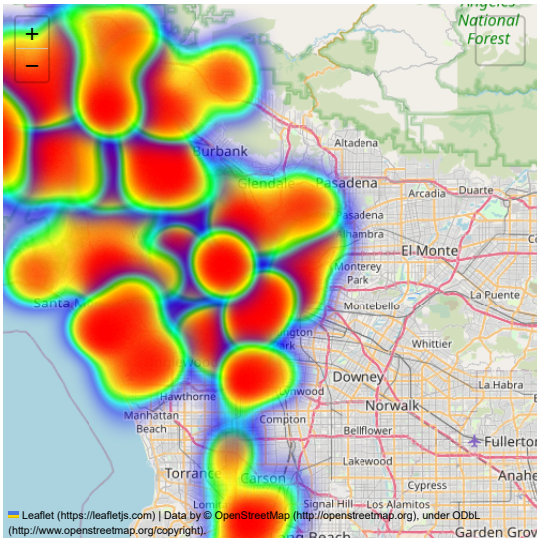
```
filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
filtered_df.shape
```

```
#Creación de mapa de calor con Serie de Tiempo
from folium import plugins

filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
filtered_df = filtered_df[['LAT','LON','DATE OCC']]
filtered_df['time'] = pd.to_datetime(filtered_df['DATE OCC'])
filtered_df = filtered_df.drop('DATE OCC', axis=1)

# Agrupamos los datos por tiempo
time_indexed = filtered_df.groupby(filtered_df['time'].dt.to_pydatetime()).apply(lambda x: x[['LAT', 'LON']].values.tolist()).tolist()

# Creamos el mapa
crime_map = folium.Map(width=500, height=500, location=[34.0522, -118.2437], zoom_start=12)
hm = plugins.HeatMapWithTime(time_indexed, radius=15, auto_play=True, min_speed=0.1, max_speed=10, speed_step=5.0)
hm.add_to(crime_map)
crime_map.save('mapa_serie_de_tiempo.html')
crime_map
```

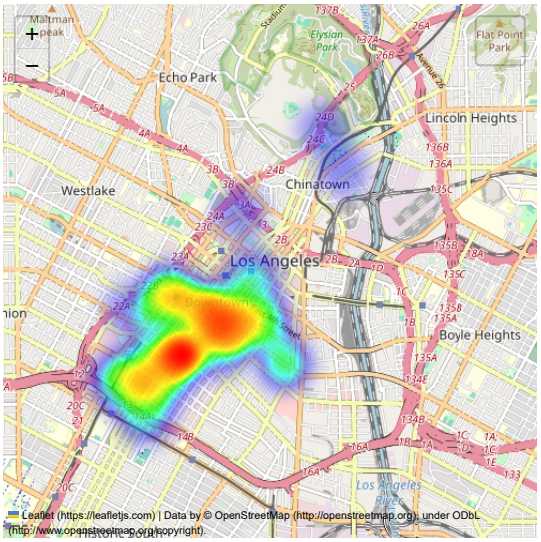
Make this Notebook Trusted to load map: File -> Trust Notebook



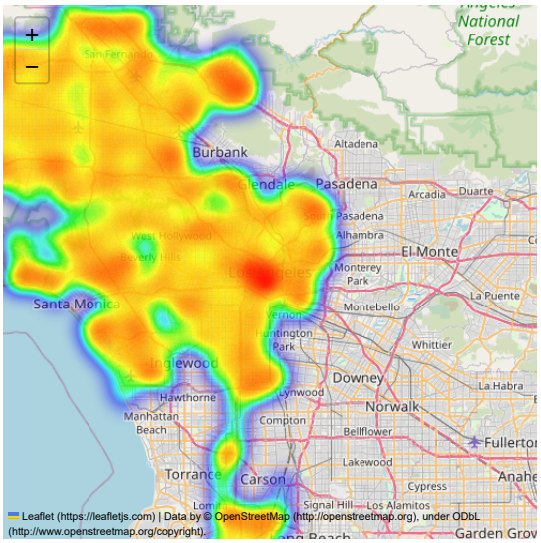

```
#Mapa de calor por capas
crime_map = folium.Map(width=500,height=500, location=[34.0522, -118.2437], zoom_start=10)

for area in range(1, 22):
  filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
  filtered_df = filtered_df[(filtered_df['AREA']== area) ]
  heat_data = [[row['LAT'], row['LON']] for index, row in filtered_df.iterrows()]
  HeatMap(heat_data, min_opacity=0.4,blur = 18).add_to(folium.FeatureGroup(name='Area ' +str(area)).add_to(crime_map))

folium.LayerControl().add_to(crime_map)
crime_map.save('Mapa por capas.html')
crime_map
```




```
#Mapa de calor de Crimen: VEHICLE - STOLEN -510
crime_map = folium.Map(width=500, height=500, location=[34.0522, -118.2437], zoom_start=13)
filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
filtered_df = filtered_df[(filtered_df['AREA']==1) &  (filtered_df['Crm Cd']==510)]
heat_data = [[row['LAT'], row['LON']] for index, row in filtered_df.iterrows()]
HeatMap(heat_data, min_opacity=0.4,blur = 18).add_to(folium.FeatureGroup(name='Heat Map').add_to(crime_map))
folium.LayerControl().add_to(crime_map)
crime_map.save('Area1 Robo Vehiculos.html')
crime_map
```




```
# Crear un mapa centra en Los Angeles

crime_map = folium.Map(width=700, height=700, location=[34.0522, -118.2437], zoom_start=10)
filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
heat_data = [[row['LAT'], row['LON']] for index, row in filtered_df.iterrows()]
HeatMap(heat_data, radius=15).add_to(crime_map)
crime_map.save('hotspots.html')
crime_map
```



Limpieza de Datos

Elimininaremos los registros que no cuenten con valores en Latitud y Longitud

```
filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
points = [[row['LAT'], row['LON']] for index, row in filtered_df.iterrows()]
filtered_df.shape
```

```
(908443, 28)
```

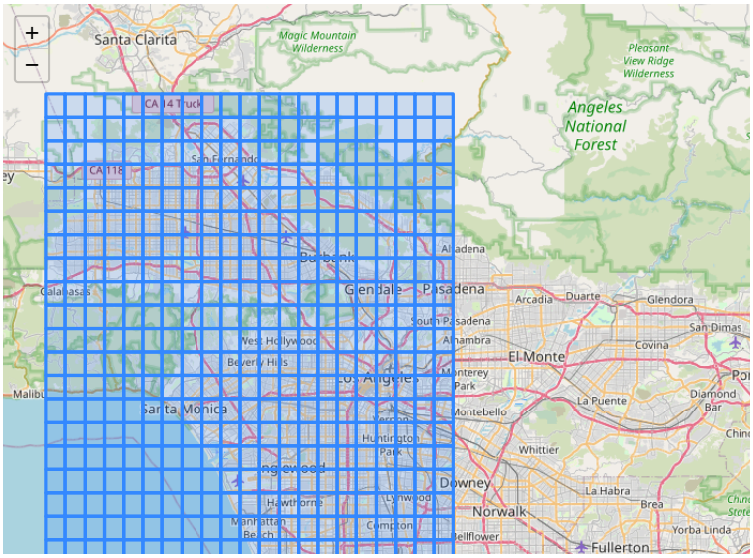
Generación de nuevo conjunto de datos

Generación de nuevas caracteristicas (features engineering)

```
import geopandas as gpd
from shapely.geometry import box



# Definir los límites de la ciudad de Los Ángeles
delta = 0.001  # Margen de error para los límites
min_lat = filtered_df['LAT'].min() - delta
max_lat = filtered_df['LAT'].max() + delta
min_lon = filtered_df['LON'].min() - delta
max_lon = filtered_df['LON'].max() + delta

print(f"Latitud Minima {min_lat}")
print(f"Latitud Maxima {max_lat}")
print(f"Longitud Minima {min_lon}")
print(f"Longitud Maxima {max_lon}")


# Definir el tamaño de cada celda de la cuadrícula en grados
# (0.01 grados es aproximadamente 1.1 km; ajusta según sea necesario)
```

```
cell_size = 0.025

# Crear listas para almacenar las celdas de la cuadrícula
cells = []
cell_counts = {}
cell_names = {}

# Crear la cuadrícula
lat_steps = int((max_lat - min_lat) / cell_size) + 1
lon_steps = int((max_lon - min_lon) / cell_size) + 1

catidad_total_de_celdas = lat_steps * lon_steps

print(f"Tamaño de la matriz {lat_steps}x{lon_steps} =",str(lat_steps * lon_steps))

for i in range(lat_steps):
    for j in range(lon_steps):
        minx = min_lon + (j * cell_size)
        maxx = min_lon + ((j + 1) * cell_size)
        miny = min_lat + (i * cell_size)
        maxy = min_lat + ((i + 1) * cell_size)
        cell = box(minx, miny, maxx, maxy)
        cells.append(cell)
        cell_counts[cell] = 0
        cell_names[cell] = (i + 1) * ( j+ 1)


# Crear un GeoDataFrame con las celdas de la cuadrícula
gdf = gpd.GeoDataFrame({'geometry': cells})
# Establecer el índice para que inicie en 1
gdf.index = range(1, len(gdf) + 1)

# Guardar el GeoDataFrame en un archivo shapefile
gdf.to_file('la_grid.shp')

# Mostrar el GeoDataFrame
import folium

# Crear un mapa centrado en Los Ángeles
m = folium.Map(width=700, height=700, location=[34.0522, -118.2437], zoom_start=10)

# Agregar cada celda de la cuadrícula al mapa
for _, row in gdf.iterrows():
    geo_json = folium.GeoJson(row['geometry'])
    geo_json.add_to(m)

# Guardar el mapa en un archivo HTML
m.save('Mapa_Cuadricula.html')
m
```

```
Latitud Minima 33.7049
Latitud Maxima 34.3353
Longitud Minima -118.6686
Longitud Maxima -118.1544
Tamaño de la matriz 26x21 = 546
```



Calculamos la Geocelda para cada punto

```
from shapely.geometry import Point

#Función para encontrar la celda correspondiente a un punto
import geopandas as gpd
from shapely.geometry import Point
import concurrent.futures

def find_cell(lat, lon):
    point = Point(lon, lat)
    for idx, cell in enumerate(cells, start=1):
        if cell.contains(point):
            return cell, idx
        if cell.intersects(point):
            # print(f"Interseccion ({lat}, {lon})")
            return cell, idx
        if cell.touches(point):
            # print(f"Lo toca ({lat}, {lon})")
            return cell, idx
    return None, None

delta = 0.01
min_lat = 33.7049 - delta
max_lat = 34.3353 + delta
min_lon = -118.6686 - delta
max_lon = -118.1544 + delta

cell_size = 0.025

import math

num_fil = int((max_lat - min_lat) / cell_size) +1
num_col = int((max_lon - min_lon) / cell_size) +1
print(f"Tamaño de la matriz {num_fil}x{num_col} =",str(num_fil * num_col))
```

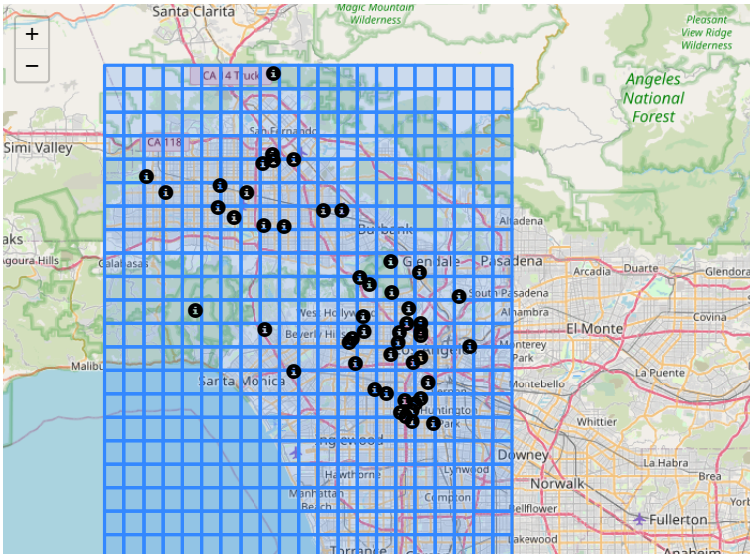
```
celdas = {i: 0 for i in range(1, num_fil * num_col)}

filtered_df = data[(data['LAT'].notnull()) & (data['LON'].notnull()) & (data['LAT'] != 0) & (data['LON'] != 0)]
count = filtered_df.shape[0]
print(f"Cantidad de crimenes : {count}")

for index, row in filtered_df.iterrows():
    x = row['LAT']
    y = row['LON']
    _, filtered_df.loc[index,'GeoCelda']   = find_cell(x,y)
    filtered_df.loc[index,'Index']= index

    if index % (90844)  == 0:
      print(f"Procesando {index} de {count} - {str(index/count *100)}%")
    if index % (90844)*10  == 0:
      filtered_df.to_csv('delitos_con_celda.csv', index=False)
```

```
Tamaño de la matriz 27x22 = 594
Cantidad de crimenes : 908443
<ipython-input-59-aa9cab6af61e>:35: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  _, filtered_df.loc[index,'GeoCelda']   = find_cell(x,y)
<ipython-input-59-aa9cab6af61e>:36: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df.loc[index,'Index']= index
Procesando 0 de 908443 - 0.0%
Procesando 90844 de 908443 - 9.999966976464126%
Procesando 181688 de 908443 - 19.999933952928252%
Procesando 272532 de 908443 - 29.999900929392375%
Procesando 363376 de 908443 - 39.999867905856505%
Procesando 454220 de 908443 - 49.999834882320634%
Procesando 545064 de 908443 - 59.99980185878475%
Procesando 635908 de 908443 - 69.99976883524887%
Procesando 726752 de 908443 - 79.99973581171301%
Procesando 817596 de 908443 - 89.99970278817713%
Procesando 908440 de 908443 - 99.99966976464127%
```

Empieza a programar o a crear código con IA.

Probamos la visualización y ubicación en celdas

```
import geopandas as gpd
from shapely.geometry import box
import folium

#Leemos el archivo shape y la cantidad de delitos por celda
gdf = gpd.read_file('la_grid.shp')
import pandas as pd
data = pd.read_csv('delitos_con_celda.csv')

m = folium.Map(width=700, height=700, location=[34.0522, -118.2437], zoom_start=10)

for index, row in data.iterrows():
  x = row['LAT']
  y = row['LON']
  cell_index = row['GeoCelda']

  folium.Marker(location=[x, y], popup='Zona: '+str(cell_index)  ,icon=folium.Icon(color='red', icon='info-sign')).add_to(m)
  if index == 50:
    break;
for _, row in gdf.iterrows():
    geo_json = folium.GeoJson(row['geometry'])
    geo_json.add_to(m)

# Guardar el mapa en un archivo HTML
m.save('Mapa_Cuadricula_con_marcadores.html')
m
```



Creación de nuevo conjunto de datos de serie de tiempo

```
from datetime import datetime, timedelta
import pandas as pd
import warnings

from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/My Drive/Precrimen/DataSets

# Suprimir todas las advertencias
warnings.filterwarnings('ignore')

delitos_celda = pd.read_csv('delitos_con_celda.csv')
delitos_celda['DATE OCC'] = pd.to_datetime(delitos_celda['DATE OCC'])
delitos_celda = delitos_celda[(delitos_celda['GeoCelda'].notnull()) & (delitos_celda['GeoCelda']!=0)]

import numpy as np
import geopandas as gpd


delitos_celda = delitos_celda.rename(columns={'DATE OCC': 'Fecha'})
delitos_celda['GeoCelda'] = delitos_celda['GeoCelda'].astype(int)
delitos_celda = delitos_celda.rename(columns={'GeoCelda': 'Zona'})

maxima_celda = catidad_total_de_celdas
fecha_inicio = delitos_celda['Fecha'].min()
fecha_fin = delitos_celda['Fecha'].max()

# Generar una lista de fechas y zonas
fechas = pd.date_range(start=fecha_inicio, end=fecha_fin)

# Generar todas las zonas de 1 a la máxima fecha
zonas = np.arange(1, maxima_celda)

# Crear un DataFrame con todas las combinaciones de fechas y zonas
fechas_zonas = pd.MultiIndex.from_product([fechas, zonas], names=['Fecha', 'Zona'])

df_completo = pd.DataFrame(index=fechas_zonas).reset_index()
```

```
df_completo['Cantidad'] = 0
df_completo.head()

df_grouped = delitos_celda.groupby(['Fecha', 'Zona']).size().reset_index(name='Cantidad')
df_grouped.head()

# Hacer un merge con el DataFrame original para rellenar los datos faltantes
serie_de_tiempo = pd.merge(df_completo, df_grouped, on=['Fecha', 'Zona'], how='left')
serie_de_tiempo['Cantidad de Crimenes'] = serie_de_tiempo.apply(lambda row: row['Cantidad_x'] if pd.isnull(row['Cantidad_y']) else int(max(row['Cantidad_x'], row['Cantidad_y'])))
serie_de_tiempo.drop(['Cantidad_x', 'Cantidad_y'], axis=1, inplace=True)
serie_de_tiempo.to_csv('serie_de_tiempo_area.csv', index=False)

print(f'El tamaño del DataFrame original es: {df_completo.shape}')
print(f'El tamaño del DataFrame con los datos agrupados es: {df_grouped.shape}')
print(f'El tamaño del DataFrame final es: {serie_de_tiempo.shape}')

serie_de_tiempo.head()
```

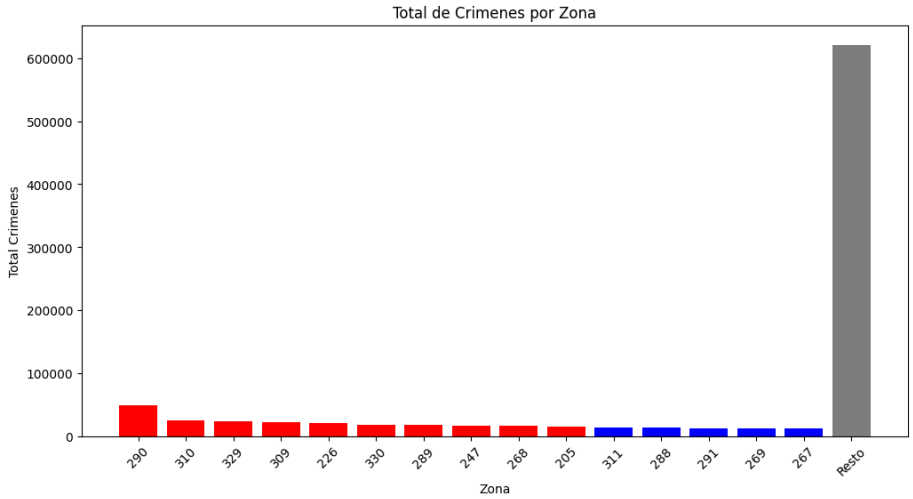
Mostrar salida oculta

Empieza a programar o a crear código con IA.

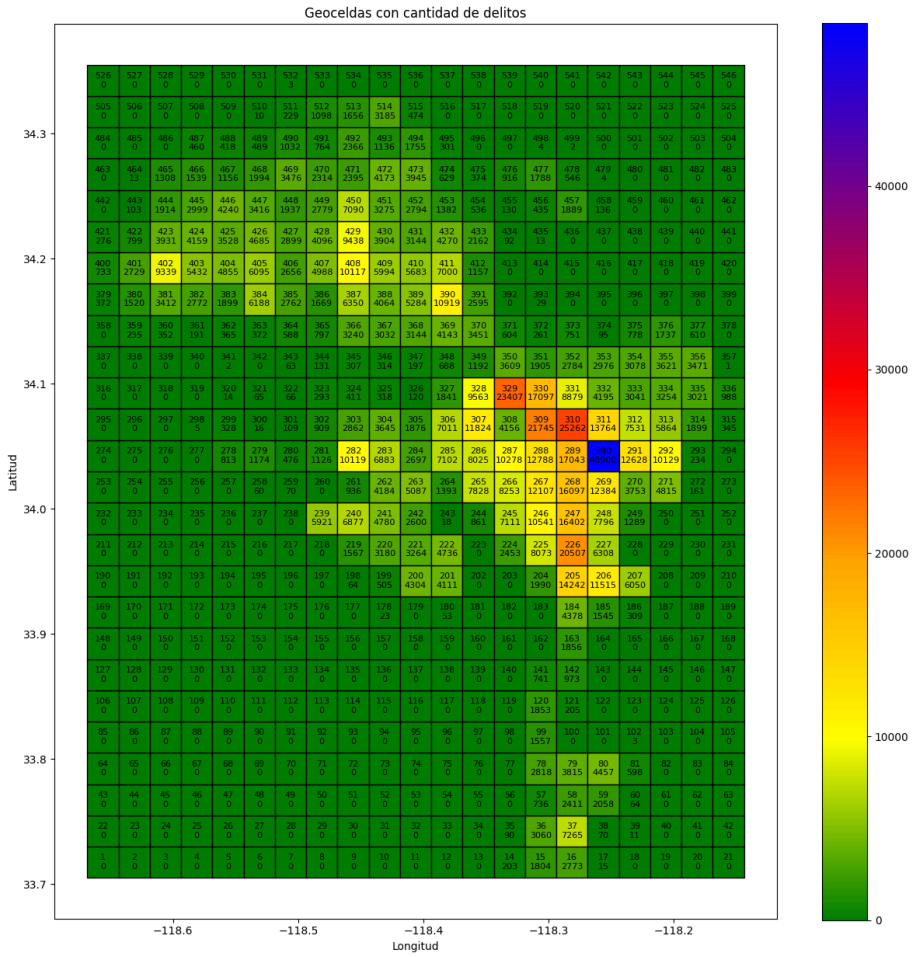
```
serie_de_tiempo['Cantidad de Crimenes'].sum()
```

```
906177
```

```
#Leemos el archivo shape y la cantidad de delitos por celda
gdf = gpd.read_file('la_grid.shp')
import pandas as pd
data = pd.read_csv('serie_de_tiempo.csv')


crimenes_por_zona = data.groupby('Zona')['Cantidad de Crimenes'].sum().reset_index()
crimenes_por_zona.columns = ['Zona', 'Total Crimenes']
zonas_completas = pd.DataFrame({'Zona': range(1, 547)})
# Combinar con el DataFrame original para asegurarnos de tener todas las zonas
total_crimenes_zona = zonas_completas.merge(crimenes_por_zona, on='Zona', how='left')

# Rellenar los valores faltantes en 'Total Crimenes' con 0
total_crimenes_zona['Total Crimenes'].fillna(0, inplace=True)


total_crimenes_zona['Total Crimenes'] = total_crimenes_zona['Total Crimenes'].astype(int)
total_crimenes_zona = total_crimenes_zona.sort_values(by='Zona')
total_crimenes_zona.reset_index(drop=True, inplace=True)
# Suponiendo que el DataFrame 'crimenes_completos' ya está cargado

# Ordenar el DataFrame por 'Total Crimenes' de mayor a menor
crimenes_completos = total_crimenes_zona.sort_values(by='Total Crimenes', ascending=False).reset_index(drop=True)

# Crear una nueva columna para categorizar las barras
crimenes_completos['Categoria'] = 'Resto'
crimenes_completos.loc[:9, 'Categoria'] = 'Top 10'
crimenes_completos.loc[10:14, 'Categoria'] = 'Siguientes 5'

# Crear un DataFrame acumulado para el resto de las zonas
acumulado = pd.DataFrame({
    'Zona': ['Resto'],
    'Total Crimenes': [crimenes_completos.loc[15:, 'Total Crimenes'].sum()],
    'Categoria': ['Resto']
})

# Filtrar los primeros 15 y añadir el acumulado
crimenes_grafico = pd.concat([crimenes_completos.loc[:14], acumulado], ignore_index=True)

# Definir colores para cada categoría
colores = ['red' if cat == 'Top 10' else 'blue' if cat == 'Siguientes 5' else 'gray' for cat in crimenes_grafico['Categoria']]

# Graficar
plt.figure(figsize=(12, 6))
plt.bar(crimenes_grafico['Zona'].astype(str), crimenes_grafico['Total Crimenes'], color=colores)
plt.xlabel('Zona')
plt.ylabel('Total Crimenes')
plt.title('Total de Crimenes por Zona')
plt.xticks(rotation=45)
plt.show()
```

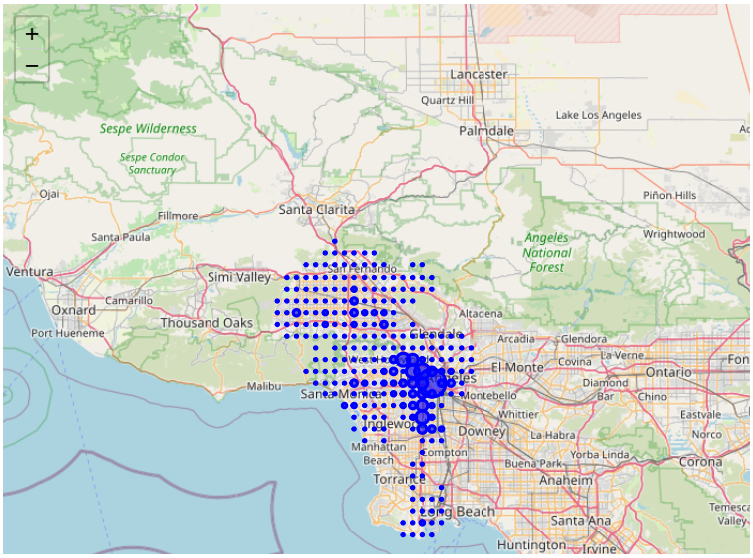



```
import matplotlib.colors as colors
# Crear una figura y un eje
fig, ax = plt.subplots(figsize=(15, 15))
# Normalizar los valores del array para mapearlos a colores
norm = colors.Normalize(vmin=0, vmax=total_crimenes_zona['Total Crimenes'].max())
cmap = colors.LinearSegmentedColormap.from_list('custom_cmap', ['green', 'yellow', 'orange', 'red','purple', 'blue'])

# Crear una lista de colores basada en los valores del array
colors = [cmap(norm(value)) for value in total_crimenes_zona['Total Crimenes']]
# Añadir una columna de colores al GeoDataFrame
gdf['color'] = colors
gdf.plot(ax=ax, color=gdf['color'], edgecolor='black')
for idx, row in gdf.iterrows():
    geom = row['geometry']
    x, y = geom.centroid.x, geom.centroid.y
    zona = idx + 1
    total_crimenes = total_crimenes_zona[total_crimenes_zona['Zona'] == zona]['Total Crimenes'].values[0]
    ax.text(x, y, str(zona) +'\n' +str(int(total_crimenes)), horizontalalignment='center', verticalalignment='center', color='black', fontsize=8)

# Añadir una barra de color
sm = plt.cm.ScalarMappable(cmap=cmap, norm=norm)
sm.set_array([])  # Se necesita solo para la barra de color
fig.colorbar(sm, ax=ax)

# Configurar el título y las etiquetas
ax.set_title('Geoceldas con cantidad de delitos')
ax.set_xlabel('Longitud')
ax.set_ylabel('Latitud')

# Mostrar la gráfica
plt.show()

plt.savefig('Imagenes\cuadricula_de_Geoceldas.png')
```

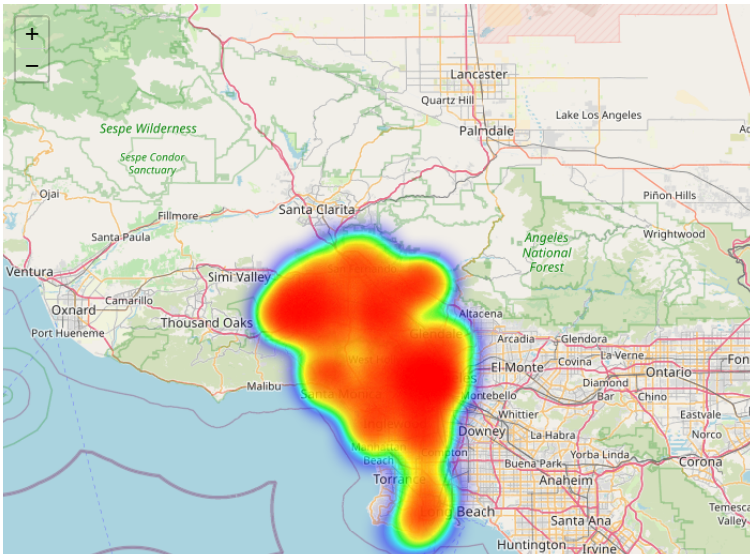


<Figure size 640x480 with 0 Axes>

```
import folium
from folium import CircleMarker
m = folium.Map(width=700, height=700, location=[34.0522, -118.40], zoom_start=9)

acum_crimenes = total_crimenes_zona['Total Crimenes'].sum()
# Iterar sobre el GeoDataFrame y añadir puntos al mapa
for idx, row in gdf.iterrows():
    geom = row['geometry']
    x, y = geom.centroid.y, geom.centroid.x  # Folium utiliza latitud y longitud en este orden
    zona = idx + 1
    total_crimenes = int(total_crimenes_zona[total_crimenes_zona['Zona'] == zona]['Total Crimenes'].values[0])
    # Ajustar el tamaño del punto basado en el valor del array
    if total_crimenes > 0:
      radius = (total_crimenes / acum_crimenes)*250
      # Añadir el punto al mapa
      folium.CircleMarker(
          location=[x, y],
          radius=radius,
          color='blue',
          fill=True,
          fill_color='blue',
          fill_opacity=0.6,
          popup=f'Valor: {total_crimenes}'
      ).add_to(m)

# Guardar el mapa en un archivo HTML
m.save('mapa_con_centroides.html')
m
```




```
import folium
from folium.plugins import HeatMap

# Crear el mapa centrado en la ubicación aproximada de Los Ángeles
m = folium.Map(width=700, height=700, location=[34.0522, -118.40], zoom_start=9)

# Crear una lista de coordenadas y valores para el HeatMap
heat_data = []
for idx, row in gdf.iterrows():
    geom = row['geometry']
    x, y = geom.centroid.y, geom.centroid.x  # Folium utiliza latitud y longitud en este orden
    zona = idx + 1
    total_crimenes = int(total_crimenes_zona[total_crimenes_zona['Zona'] == zona]['Total Crimenes'].values[0])

    if total_crimenes > 0:
      heat_data.append([x, y, total_crimenes])  # Añadir la latitud, longitud y el valor a la lista

# Añadir el HeatMap al mapa
HeatMap(heat_data).add_to(m)

# Guardar el mapa en un archivo HTML
m.save('mapa_interactivo_heatmap.html')
m
```



Analizaremos la serie de tiempo

Cargamos los datos de la serie de tiempo por Zona

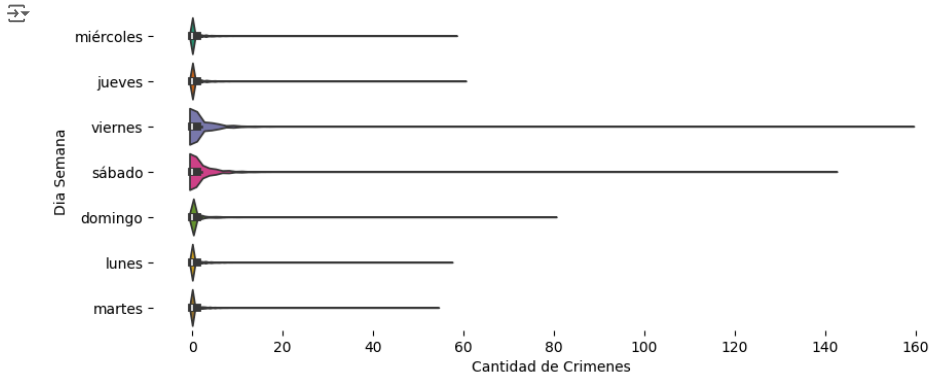
```
# @title Cargamos los datos de la serie de tiempo por Zona
from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/My Drive/Precrimen/DataSets

import warnings
warnings.filterwarnings('ignore')

# 1. Cargar y preprocesar los datos
import pandas as pd
data = pd.read_csv('serie_de_tiempo.csv')
data['Fecha'] = pd.to_datetime(data['Fecha'])
data.info()
df = data.copy()
df.info()

def Recargar_datos():
  from google.colab import drive
  drive.mount('/content/drive')
  %cd /content/drive/My Drive/Precrimen/DataSets
  data = pd.read_csv('serie_de_tiempo.csv')
  data['Fecha'] = pd.to_datetime(data['Fecha'])
  df = data.copy()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
/content/drive/My Drive/Precrimen/DataSets
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 831125 entries, 0 to 831124
Data columns (total 3 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   Fecha                831125 non-null  datetime64[ns]
 1   Zona                 831125 non-null  int64
 2   Cantidad de Crimenes  831125 non-null  int64
dtypes: datetime64[ns](1), int64(2)
```

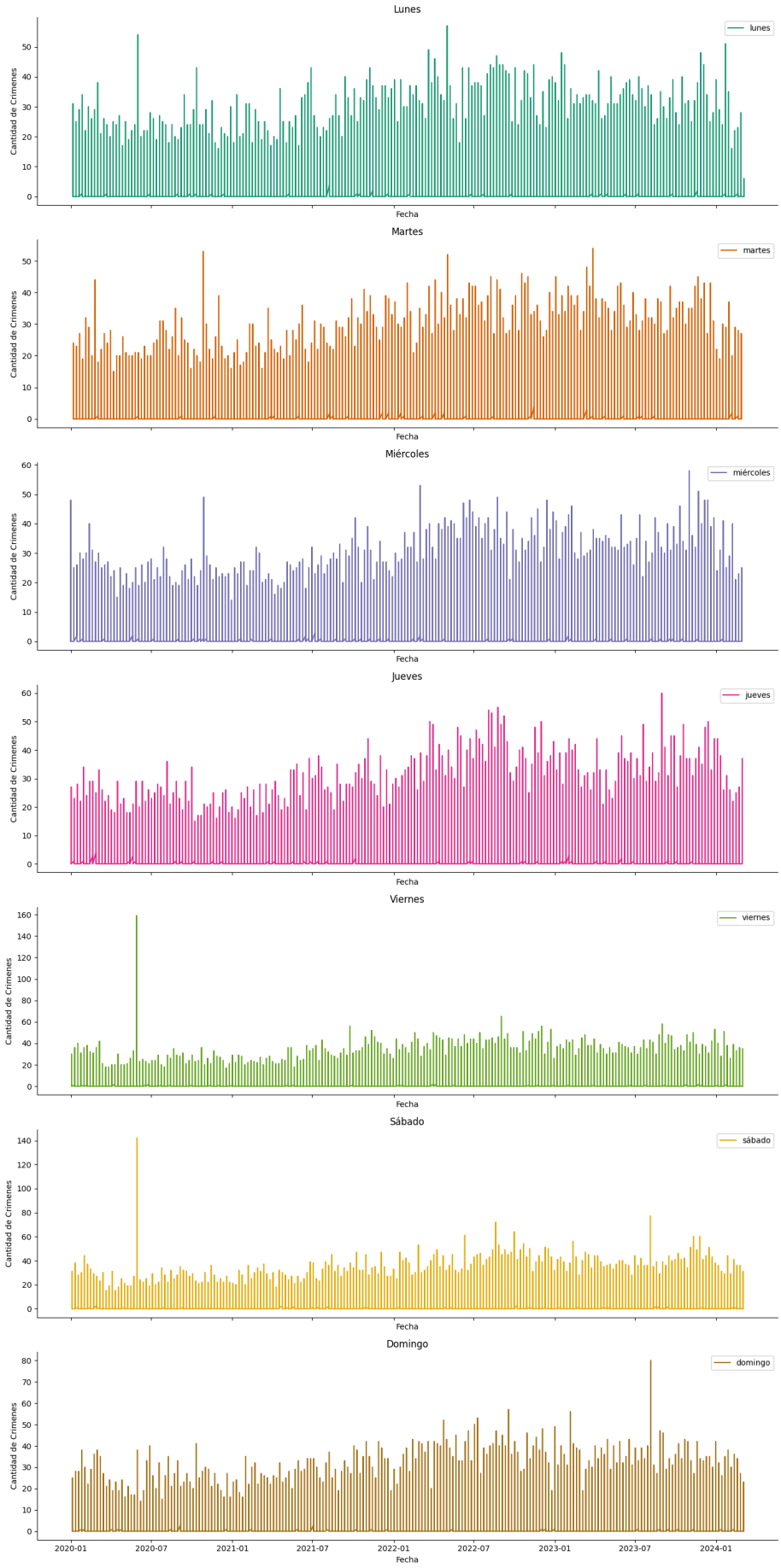
```
memory usage: 19.0 MB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 831125 entries, 0 to 831124
Data columns (total 3 columns):
 #   Column                Non-Null Count   Dtype         
---  ------                --------------   -----         
 0   Fecha                 831125 non-null  datetime64[ns]
 1   Zona                  831125 non-null  int64         
 2   Cantidad de Crimenes  831125 non-null  int64         
dtypes: datetime64[ns](1), int64(2)
memory usage: 19.0 MB
```

Gráfica de Día de la semana vs Cantidad de Crimenes

```
# @title Gráfica de Día de la semana vs Cantidad de Crimenes
dias_semana = ['lunes', 'martes', 'miércoles', 'jueves', 'viernes', 'sábado', 'domingo']
# Agregar una columna con el nombre del día de la semana en español
serie_de_tiempo['Dia Semana'] = serie_de_tiempo['Fecha'].dt.dayofweek.apply(lambda x: dias_semana[x])


from matplotlib import pyplot as plt
import seaborn as sns
figsize = (10, 4)
plt.figure(figsize=figsize)
sns.violinplot(serie_de_tiempo, x='Cantidad de Crimenes', y='Dia Semana', inner='box', palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

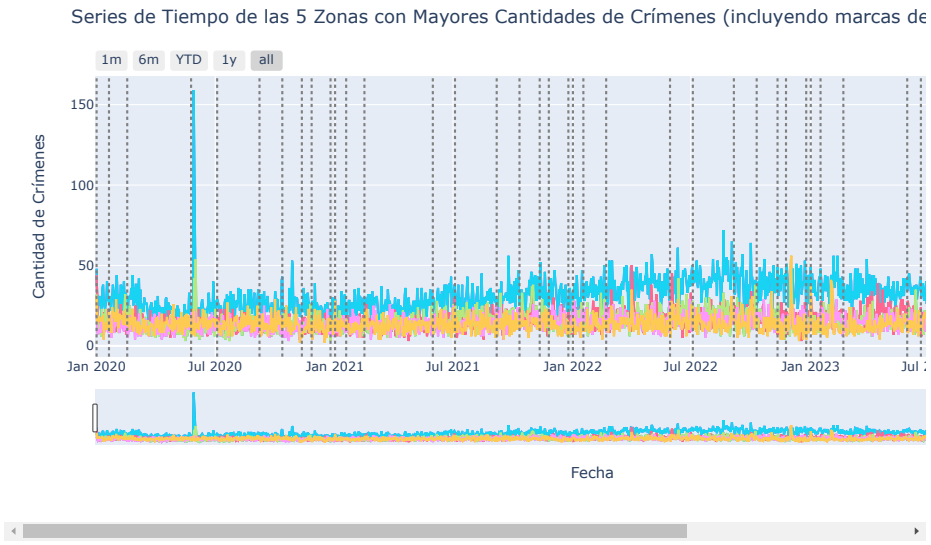



```
def _plot_series(ax, series, series_name, series_index=0):
    palette = list(sns.color_palette('Dark2'))
    xs = series['Fecha']
    ys = series['Cantidad de Crimenes']
    ax.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])
    ax.set_title(series_name.capitalize())
    ax.set_xlabel('Fecha')
    ax.set_ylabel('Cantidad de Crimenes')
    ax.legend()
    sns.despine(ax=ax)

# Crear la figura y los ejes para los siete días de la semana
fig, axes = plt.subplots(7, 1, figsize=(14, 28), sharex=True)

df_sorted = serie_de_tiempo.sort_values('Fecha', ascending=True)

# Asegurar que los días de la semana estén en orden
dias_semana_ordenados = ['lunes', 'martes', 'miércoles', 'jueves', 'viernes', 'sábado', 'domingo']
for i, dia in enumerate(dias_semana_ordenados):
    series = df_sorted[df_sorted['Dia Semana'] == dia]
    _plot_series(axes[i], series, dia, i)

# Ajustar el diseño
plt.tight_layout()

# Mostrar la gráfica
plt.show()
```

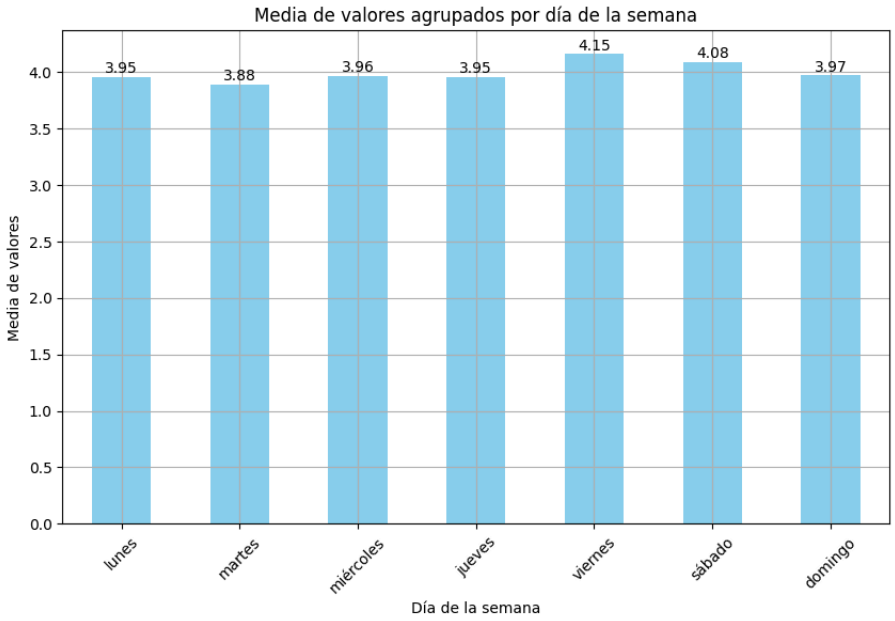

Lunes
lunes
Cantidad de Crimenes
Fecha
Martes
martes
Cantidad de Crimenes
Fecha
Miércoles
miércoles
Cantidad de Crimenes
Fecha
Jueves
jueves
Cantidad de Crimenes
Fecha
Viernes
viernes
Cantidad de Crimenes
Fecha
Sábado
sábado
Cantidad de Crimenes
Fecha
Domingo
domingo
Cantidad de Crimenes
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01
Fecha

Gráfica de las series de tiempo incluyendo festivos

```
# @title Gráfica de las series de tiempo incluyendo festivos
#!pip install -U kaleido
#!pip install plotly --upgrade
import pandas as pd
import plotly.graph_objects as go


# Calcular el total acumulado de 'Cantidad de Crímenes' por Zona
zona_totales = df.groupby('Zona')['Cantidad de Crimenes'].sum().sort_values(ascending=False)

# Seleccionar las 5 Zonas con mayores valores acumulados
top_5_zonas = zona_totales.head(5).index
# Filtrar el DataFrame para incluir solo las Zonas seleccionadas
df_top_5 = df[df['Zona'].isin(top_5_zonas)]
# Crear el gráfico usando Plotly
fig = go.Figure()

# Añadir una serie de tiempo para cada una de las 5 Zonas principales
for zona in top_5_zonas:
    df_zona = df_top_5[df_top_5['Zona'] == zona]
    fig.add_trace(go.Scatter(x=df_zona['Fecha'], y=df_zona['Cantidad de Crimenes'], mode='lines', name=f'Zona {zona}'))

# Lista de fechas festivas en USA (para 2023, por ejemplo)
festivos_usa = [
    '2020-01-01', '2020-01-20', '2020-02-17', '2020-05-25', '2020-07-04', '2020-09-07',
    '2020-10-12', '2020-11-11', '2020-11-26', '2020-12-25', '2021-01-01', '2021-01-18',
    '2021-02-15', '2021-05-31', '2021-07-04', '2021-09-06', '2021-10-11', '2021-11-11',
    '2021-11-25', '2021-12-25', '2022-01-01', '2022-01-17', '2022-02-21', '2022-05-30',
    '2022-07-04', '2022-09-05', '2022-10-10', '2022-11-11', '2022-11-24', '2022-12-25',
    '2023-01-01', '2023-01-16', '2023-02-20', '2023-05-29', '2023-06-19', '2023-07-04',
    '2023-09-04', '2023-10-09', '2023-11-10', '2023-11-23', '2023-12-25'
]
festivos_usa = pd.to_datetime(festivos_usa)


# Añadir una serie de tiempo para cada una de las 5 Zonas principales
for zona in top_5_zonas:
    df_zona = df_top_5[df_top_5['Zona'] == zona]
    fig.add_trace(go.Scatter(x=df_zona['Fecha'], y=df_zona['Cantidad de Crimenes'], mode='lines', name=f'Zona {zona}'))

festivos_usa = pd.to_datetime(festivos_usa)

# Añadir líneas para los días festivos
for festivo in festivos_usa:
    fig.add_shape(
        dict(
            type="line",
            x0=festivo,
            y0=0,
            x1=festivo,
            y1=1,
            xref='x',
            yref='paper',
            line=dict(color="gray", width=2, dash="dot")
        )
    )


# Configurar el título y las etiquetas
fig.update_layout(
    title='Series de Tiempo de las 5 Zonas con Mayores Cantidades de Crímenes (incluyendo marcas de días festivos)',
    xaxis_title='Fecha',
    yaxis_title='Cantidad de Crímenes',
    legend_title='Zonas',
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1, label='1m', step='month', stepmode='backward'),
                dict(count=6, label='6m', step='month', stepmode='backward'),
                dict(count=1, label='YTD', step='year', stepmode='todate'),
                dict(count=1, label='1y', step='year', stepmode='backward'),
                dict(step='all')
            ])
        ),
        rangeslider=dict(visible=True),
        type='date'
    )
)

# Guardar el gráfico como imagen
#fig.write_image('Top_5_Zonas_Crimenes.png', engine='kaleido')

# Mostrar el gráfico
fig.show()
```




```
df_agrupado = serie_de_tiempo[serie_de_tiempo['Cantidad de Crimenes']!=0].groupby('Dia Semana')['Cantidad de Crimenes'].mean().reindex(dias_semana)

# Visualizar la serie de tiempo agrupada por día de la semana
plt.figure(figsize=(10, 6))
df_agrupado.plot(kind='bar', color='skyblue')
plt.title('Media de valores agrupados por día de la semana')
plt.xlabel('Día de la semana')
plt.ylabel('Media de valores')
plt.xticks(rotation=45)
plt.grid(True)
# Agregar los valores encima de las barras
for i, valor in enumerate(df_agrupado):
    plt.text(i, valor + 0.01, f'{valor:.2f}', ha='center', va='bottom')
plt.show()
```

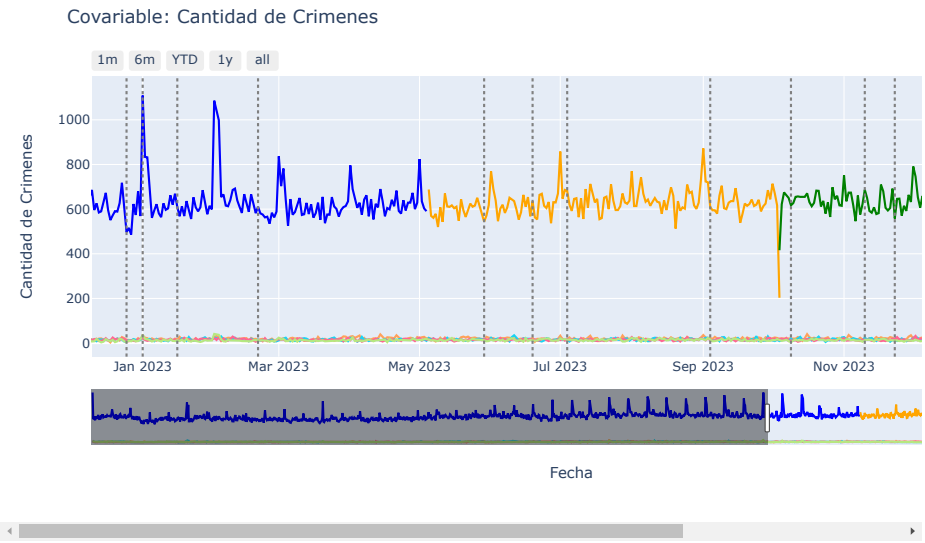


Preprocesamiemto

```
N = df.shape[0]
Ntrain = int(0.8*N)  # Número de datos de entrenamiento
Nval = int(0.1*N)    # Número de datos de validación
Ntst = N - Ntrain - Nval # Número de datos de prueba
train = data[0:Ntrain]
val = data[Ntrain:Ntrain+Nval]
test = data[Ntrain+Nval:]
print(f'Tamaño set completo: {df.shape}')
print(f'Tamaño set completo: {data.shape}')
print(f'Tamaño set de entrenamiento: {train.shape}')
print(f'Tamaño set de validación: {val.shape}')
print(f'Tamaño set de prueba: {test.shape}')
```

```
Tamaño set completo: (831125, 3)
Tamaño set completo: (831125, 3)
Tamaño set de entrenamiento: (664900, 3)
Tamaño set de validación: (83112, 3)
Tamaño set de prueba: (83113, 3)
```

```
import plotly.graph_objects as go
import plotly.express as px

col = 'Cantidad de Crimenes'
# Crear el gráfico usando Plotly
fig = go.Figure()

train_grouped = train.groupby('Fecha')['Cantidad de Crimenes'].sum().reset_index()
val_grouped = val.groupby('Fecha')['Cantidad de Crimenes'].sum().reset_index()
test_grouped = test.groupby('Fecha')['Cantidad de Crimenes'].sum().reset_index()

fig.add_trace(go.Scatter(x=train_grouped['Fecha'], y=train_grouped['Cantidad de Crimenes'], mode='lines', name='Entrenamiento', line=dict(color='blue')))
fig.add_trace(go.Scatter(x=val_grouped['Fecha'], y=val_grouped['Cantidad de Crimenes'], mode='lines', name='Validación', line=dict(color='orange')))
fig.add_trace(go.Scatter(x=test_grouped['Fecha'], y=test_grouped['Cantidad de Crimenes'], mode='lines', name='Pruebas', line=dict(color='green')))

# Añadir una serie de tiempo para cada una de las 5 Zonas principales
for zona in top_5_zonas:
    df_zona = df_top_5[df_top_5['Zona'] == zona]
    fig.add_trace(go.Scatter(x=df_zona['Fecha'], y=df_zona['Cantidad de Crimenes'], mode='lines', name=f'Zona {zona}'))

# Añadir líneas para los días festivos
for festivo in festivos_usa:
    fig.add_shape(
        dict(
            type="line",
            x0=festivo,
            y0=0,
            x1=festivo,
            y1=1,
            xref='x',
            yref='paper',
            line=dict(color="gray", width=2, dash="dot")
        )
    )


# Configurar el título y las etiquetas
fig.update_layout(
    title=f'Covariable: {col} ',
    xaxis_title='Fecha',
    yaxis_title=col,
    legend_title='Conjunto de Datos',
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1, label='1m', step='month', stepmode='backward'),
                dict(count=6, label='6m', step='month', stepmode='backward'),
                dict(count=1, label='YTD', step='year', stepmode='todate'),
                dict(count=1, label='1y', step='year', stepmode='backward'),
                dict(step='all')
            ])
        ),
        rangeslider=dict(visible=True),
        type='date'
    )
)
# Guardar el gráfico como imagen
#fig.write_image('Datos'+str(zona)+'.png')

# Mostrar el gráfico
fig.show()
```



Feature engeniering

```
import holidays
data['Fecha'] = pd.to_datetime(data['Fecha'])
df = data.copy()
zonas = df['Zona'].unique()
zonas = np.array(zonas)
# Crear características adicionales
us_holidays = holidays.US()
df['Es Festivo'] = df['Fecha'].isin(us_holidays).astype(int)
df['Día de la Semana'] = df['Fecha'].dt.dayofweek
df['Mes'] = df['Fecha'].dt.month
# Convertir 'Día de la Semana' en variables dummy
df = pd.get_dummies(df, columns=['Día de la Semana'], prefix='Día')
df.head()
```

	Fecha	Zona	Cantidad de Crimenes	Es Festivo	Mes	Día_0	Día_1	Día_2	Día_3	Día_4	Día_5	Día_6
0	2020-01-01	1	0	0	1	False	False	True	False	False	False	False
1	2020-01-01	2	0	0	1	False	False	True	False	False	False	False
2	2020-01-01	3	0	0	1	False	False	True	False	False	False	False
3	2020-01-01	4	0	0	1	False	False	True	False	False	False	False
4	2020-01-01	5	0	0	1	False	False	True	False	False	False	False

Modelos de Predicción

Modelo de predicción 1: RandomForestRegressor

```
!pip install lightgbm
!pip install skforecast

from lightgbm import LGBMRegressor
import pandas as pd
import holidays
from skforecast.ForecasterAutoreg import ForecasterAutoreg
from sklearn.ensemble import RandomForestRegressor
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from skforecast.model_selection import backtesting_forecaster
from sklearn.metrics import mean_absolute_error
from joblib import dump, load
import warnings
warnings.filterwarnings('ignore')

from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/My Drive/Precrimen/DataSets
data = pd.read_csv('serie_de_tiempo.csv')
data['Fecha'] = pd.to_datetime(data['Fecha'])

# Cargar el conjunto de datos
try:
    data.shape
    df = data.copy()
except NameError:
    from google.colab import drive
    drive.mount('/content/drive')
    %cd /content/drive/My Drive/Precrimen/DataSets
    data = pd.read_csv('serie_de_tiempo.csv')
    data['Fecha'] = pd.to_datetime(data['Fecha'])

# Función para generar las características adicionales
def generar_arreglo_a_predecir(dataframe):
    us_holidays = holidays.US()
    dataframe['Es Festivo'] = dataframe['Fecha'].isin(us_holidays).astype(int)
    dataframe['Día de la Semana'] = dataframe['Fecha'].dt.dayofweek
    dataframe['Mes'] = dataframe['Fecha'].dt.month
    dataframe = pd.get_dummies(dataframe, columns=['Día de la Semana'], prefix='Día')

    # Convertir las columnas Día_0 a Día_6 a tipo entero
    for i in range(7):
        columna = f'Día_{i}'
        if columna in dataframe.columns:
            dataframe[columna] = dataframe[columna].astype(int)
    return dataframe

# Generar el DataFrame con las características adicionales
df = generar_arreglo_a_predecir(data.copy())

# Definir las variables exógenas
variables_exogenas = ['Zona', 'Es Festivo', 'Mes'] + [f'Día_{i}' for i in range(7)]

data_to_train = generar_arreglo_a_predecir(data.copy())

# Calcular el total acumulado de 'Cantidad de Crímenes' por Zona
zona_totales = df.groupby('Zona')['Cantidad de Crimenes'].sum().sort_values(ascending=False)

# Seleccionar las n Zonas con mayores valores acumulados
n = 150
top_n_zonas = zona_totales.head(n).index
df_top_n = data[data['Zona'].isin(top_n_zonas)]
zonas = df_top_n['Zona'].unique()

# Inicializar un diccionario para los forecasters
forecasters = {}
predicciones_array = {}
entrenamiento_array = {}
metricas_array = []

# Crear la figura para las visualizaciones
#fig, ax = plt.subplots(len(zonas), 1, figsize=(10, 6 * len(zonas)))

# Ajustar un forecaster para cada zona
for i, zona in enumerate(zonas):
    print(f"Entrenando para la zona: {zona}")
    zona_ds = data_to_train[data_to_train['Zona'] == zona].set_index('Fecha')

    N = zona_ds.shape[0]
    Ntrain = int(0.8 * N)
    Nval = int(0.1 * N)
    Ntst = N - Ntrain - Nval


    # Verificar que la longitud de trainds sea suficiente
    if len(zona_ds) < Ntrain + 4:
        print(f"No hay suficientes datos para la zona {zona} el tamaño es {len(zona_ds)}")
        continue

    train_ds = zona_ds.iloc[:Ntrain]
    test_ds = zona_ds.iloc[Ntrain:]

    train_ds.index = pd.to_datetime(train_ds.index)
    test_ds.index = pd.to_datetime(test_ds.index)
```

```
    zona_ds = zona_ds.resample("D").sum()
    train_ds = train_ds.resample("D").sum()
    test_ds = test_ds.resample("D").sum()

    # Crear el forecaster
    forecaster = ForecasterAutoreg(regressor=RandomForestRegressor(n_estimators=10), lags=7)

    if train_ds[variables_exogenas].isnull().any().any():
        print(f"Hay valores nulos en las variables exógenas para la zona {zona}")
        continue

    forecaster.fit(y=train_ds['Cantidad de Crimenes'], exog=train_ds[variables_exogenas])
    print(f"Forecaster para la zona {zona} entrenado")

    # Guardar el forecaster en el diccionario
    forecasters[zona] = forecaster
    forecaster = forecasters[zona]


    # Realizar el backtesting
    metricas, predicciones = backtesting_forecaster(
        forecaster=forecaster,
        y=zona_ds['Cantidad de Crimenes'],
        exog=zona_ds[variables_exogenas],
        initial_train_size=Ntrain,
        steps=7,
        metric='mean_absolute_error',
        refit=True,
        verbose=False
    )
    zona_ds['pred'] = predicciones
    predicciones_array[zona] = predicciones
    entrenamiento_array[zona] = zona_ds
    metricas_array.append(metricas)
    # Calcular la métrica MAE
    # Mostrar las métricas
    print(f"Zona {zona} - Mean Absolute Error: {metricas}")
'''
    # Visualización de las predicciones
    ax[i].plot(zona_ds.index, zona_ds['Cantidad de Crimenes'], label='Datos reales', color='black')
    ax[i].plot(predicciones.index, predicciones['pred'], label='Predicciones', color='red')
    ax[i].axvline(zona_ds.index[Ntrain], color='gray', linestyle='--', linewidth=1, label='Fin del entrenamiento')
    ax[i].set_title(f'Predicciones vs Datos Reales para la Zona {zona} - Mean Absolute Error: {metricas}')
    ax[i].set_xlabel('Fecha')
    ax[i].set_ylabel('Cantidad de Crimenes')
    ax[i].legend()
'''

# Guardar los modelos en disco
dump(forecasters, 'forecasters.joblib')

# Convertir las listas a DataFrames
predicciones_df = pd.concat(predicciones_array)
entrenamiento_df = pd.concat(entrenamiento_array)
metricas_df = pd.DataFrame(metricas_array)

# Guardar los DataFrames en archivos CSV
entrenamiento_df.to_csv('entrenamiento.csv', index=True)
metricas_df.to_csv('metricas.csv', index=False)

#plt.tight_layout()
#plt.savefig('Backtesting_zonas.png')
#plt.show()
```

```
Requirement already satisfied: lightgbm in /usr/local/lib/python3.10/dist-packages (4.1.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from lightgbm) (1.25.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from lightgbm) (1.11.4)
Requirement already satisfied: skforecast in /usr/local/lib/python3.10/dist-packages (0.12.1)
Requirement already satisfied: numpy<1.27,>=1.20 in /usr/local/lib/python3.10/dist-packages (from skforecast) (1
Requirement already satisfied: pandas<2.3,>=1.2 in /usr/local/lib/python3.10/dist-packages (from skforecast) (2.
Requirement already satisfied: tqdm<4.67,>=4.57 in /usr/local/lib/python3.10/dist-packages (from skforecast) (4.
Requirement already satisfied: scikit-learn<1.5,>=1.2 in /usr/local/lib/python3.10/dist-packages (from skforecas
Requirement already satisfied: optuna<3.7,>=2.10 in /usr/local/lib/python3.10/dist-packages (from skforecast) (3
Requirement already satisfied: joblib<1.5,>=1.1 in /usr/local/lib/python3.10/dist-packages (from skforecast) (1.
Requirement already satisfied: alembic>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from optuna<3.7,>=2.10
Requirement already satisfied: colorlog in /usr/local/lib/python3.10/dist-packages (from optuna<3.7,>=2.10->skfo
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from optuna<3.7,>=2.1
Requirement already satisfied: sqlalchemy>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from optuna<3.7,>=2
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from optuna<3.7,>=2.10->skfore
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas<2.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<2.3,>=1.2->s
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas<2.3,>=1.2-
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn<1.5,>=
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-lear
Requirement already satisfied: Mako in /usr/local/lib/python3.10/dist-packages (from alembic>=1.5.0->optuna<3.7,
Requirement already satisfied: typing-extensions>=4 in /usr/local/lib/python3.10/dist-packages (from alembic>=1.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from sqlalchemy>=1.3
Requirement already satisfied: MarkupSafe>=0.9.2 in /usr/local/lib/python3.10/dist-packages (from Mako->alembic>
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", forc
/content/drive/My Drive/Precrimen/DataSets
Entrenando para la zona: 15
Forecaster para la zona 15 entrenado
100%                                         44/44 [00:02<00:00, 21.10it/s]
Zona 15 - Mean Absolute Error: 1.0337704918032786
Entrenando para la zona: 16
Forecaster para la zona 16 entrenado
100%                                         44/44 [00:00<00:00, 118.48it/s]
Zona 16 - Mean Absolute Error: 1.3440437158469944
Entrenando para la zona: 36
Forecaster para la zona 36 entrenado
100%                                         44/44 [00:00<00:00, 103.20it/s]
Zona 36 - Mean Absolute Error: 1.359344262295082
Entrenando para la zona: 37
Forecaster para la zona 37 entrenado
100%                                         44/44 [00:00<00:00, 112.16it/s]
Zona 37 - Mean Absolute Error: 2.2226229508196718
Entrenando para la zona: 58
Forecaster para la zona 58 entrenado
100%                                         44/44 [00:00<00:00, 104.05it/s]
Zona 58 - Mean Absolute Error: 1.2265573770491804
Entrenando para la zona: 59
Forecaster para la zona 59 entrenado
100%                                         44/44 [00:00<00:00, 144.55it/s]
Zona 59 - Mean Absolute Error: 1.0849180327868853
Entrenando para la zona: 78
Forecaster para la zona 78 entrenado
100%                                         44/44 [00:00<00:00, 140.93it/s]
Zona 78 - Mean Absolute Error: 1.2918032786885245
Entrenando para la zona: 79
Forecaster para la zona 79 entrenado
100%                                         44/44 [00:00<00:00, 144.36it/s]
Zona 79 - Mean Absolute Error: 1.4396721311475411
Entrenando para la zona: 80
Forecaster para la zona 80 entrenado
100%                                         44/44 [00:00<00:00, 145.17it/s]
Zona 80 - Mean Absolute Error: 1.7475409836065574
Entrenando para la zona: 99
Forecaster para la zona 99 entrenado
100%                                         44/44 [00:00<00:00, 156.34it/s]
Zona 99 - Mean Absolute Error: 0.908743169398907
Entrenando para la zona: 120
Forecaster para la zona 120 entrenado
100%                                         44/44 [00:00<00:00, 164.14it/s]
Zona 120 - Mean Absolute Error: 1.0777049180327871
Entrenando para la zona: 163
Forecaster para la zona 163 entrenado
100%                                         44/44 [00:00<00:00, 157.91it/s]
Zona 163 - Mean Absolute Error: 1.019945355191257
Entrenando para la zona: 184
Forecaster para la zona 184 entrenado
100%                                         44/44 [00:00<00:00, 143.42it/s]
Zona 184 - Mean Absolute Error: 1.598688524590164
Entrenando para la zona: 185
Forecaster para la zona 185 entrenado
100%                                         44/44 [00:00<00:00, 169.13it/s]
Zona 185 - Mean Absolute Error: 0.9886885245901642
Entrenando para la zona: 200
Forecaster para la zona 200 entrenado
100%                                         44/44 [00:00<00:00, 140.06it/s]
Zona 200 - Mean Absolute Error: 1.7822950819672128
Entrenando para la zona: 201
Forecaster para la zona 201 entrenado
100%                                         44/44 [00:00<00:00, 134.16it/s]
Zona 201 - Mean Absolute Error: 1.5839344262295083
Entrenando para la zona: 204
```

```
Forecaster para la zona 204 entrenado
```
100% 44/44 [00:00<00:00, 138.77it/s]
```
Zona 204 - Mean Absolute Error: 1.1357377049180326
Entrenando para la zona: 205
Forecaster para la zona 205 entrenado
```
100% 44/44 [00:00<00:00, 124.67it/s]
```
Zona 205 - Mean Absolute Error: 2.6744262295081964
Entrenando para la zona: 206
Forecaster para la zona 206 entrenado
```
100% 44/44 [00:00<00:00, 110.98it/s]
```
Zona 206 - Mean Absolute Error: 2.83672131147541
Entrenando para la zona: 207
Forecaster para la zona 207 entrenado
```
100% 44/44 [00:00<00:00, 109.55it/s]
```
Zona 207 - Mean Absolute Error: 1.8688524590163935
Entrenando para la zona: 219
Forecaster para la zona 219 entrenado
```
100% 44/44 [00:00<00:00, 125.57it/s]
```
Zona 219 - Mean Absolute Error: 0.9530874316939889
Entrenando para la zona: 220
Forecaster para la zona 220 entrenado
```
100% 44/44 [00:00<00:00, 125.32it/s]
```
Zona 220 - Mean Absolute Error: 1.3308196721311476
Entrenando para la zona: 221
Forecaster para la zona 221 entrenado
```
100% 44/44 [00:00<00:00, 115.68it/s]
```
Zona 221 - Mean Absolute Error: 1.4049180327868853
Entrenando para la zona: 222
Forecaster para la zona 222 entrenado
```
100% 44/44 [00:00<00:00, 116.75it/s]
```
Zona 222 - Mean Absolute Error: 2.0550819672131144
Entrenando para la zona: 224
Forecaster para la zona 224 entrenado
```
100% 44/44 [00:00<00:00, 144.17it/s]
```
Zona 224 - Mean Absolute Error: 1.113770491803279
Entrenando para la zona: 225
Forecaster para la zona 225 entrenado
```
100% 44/44 [00:00<00:00, 131.78it/s]
```
Zona 225 - Mean Absolute Error: 2.3137704918032784
Entrenando para la zona: 226
Forecaster para la zona 226 entrenado
```
100% 44/44 [00:00<00:00, 133.77it/s]
```
Zona 226 - Mean Absolute Error: 3.5160655737704922
Entrenando para la zona: 227
Forecaster para la zona 227 entrenado
```
100% 44/44 [00:00<00:00, 145.14it/s]
```
Zona 227 - Mean Absolute Error: 1.9065573770491804
Entrenando para la zona: 239
Forecaster para la zona 239 entrenado
```
100% 44/44 [00:00<00:00, 141.85it/s]
```
Zona 239 - Mean Absolute Error: 1.8055737704918033
Entrenando para la zona: 240
Forecaster para la zona 240 entrenado
```
100% 44/44 [00:00<00:00, 135.23it/s]
```
Zona 240 - Mean Absolute Error: 2.120983606557377
Entrenando para la zona: 241
Forecaster para la zona 241 entrenado
```
100% 44/44 [00:00<00:00, 141.61it/s]
```
Zona 241 - Mean Absolute Error: 1.679016393442623
Entrenando para la zona: 242
Forecaster para la zona 242 entrenado
```
100% 44/44 [00:00<00:00, 161.90it/s]
```
Zona 242 - Mean Absolute Error: 1.2134426229508197
Entrenando para la zona: 245
Forecaster para la zona 245 entrenado
```
100% 44/44 [00:00<00:00, 145.96it/s]
```
Zona 245 - Mean Absolute Error: 1.965901639344262
Entrenando para la zona: 246
Forecaster para la zona 246 entrenado
```
100% 44/44 [00:00<00:00, 134.04it/s]
```
Zona 246 - Mean Absolute Error: 2.5947540983606556
Entrenando para la zona: 247
Forecaster para la zona 247 entrenado
```
100% 44/44 [00:00<00:00, 133.48it/s]
```
Zona 247 - Mean Absolute Error: 3.14
Entrenando para la zona: 248
Forecaster para la zona 248 entrenado
```
100% 44/44 [00:00<00:00, 132.91it/s]
```
Zona 248 - Mean Absolute Error: 2.3918032786885246
Entrenando para la zona: 249
Forecaster para la zona 249 entrenado
```
100% 44/44 [00:00<00:00, 155.20it/s]
```
Zona 249 - Mean Absolute Error: 0.9137540983606557
Entrenando para la zona: 262
Forecaster para la zona 262 entrenado
```
100% 44/44 [00:00<00:00, 125.63it/s]
```
Zona 262 - Mean Absolute Error: 1.560655737704918
Entrenando para la zona: 263
```

```
Entrenando para la zona: 263
Forecaster para la zona 263 entrenado
100%                                          44/44 [00:00<00:00, 126.31it/s]
Zona 263 - Mean Absolute Error: 1.7800000000000002
Entrenando para la zona: 264
Forecaster para la zona 264 entrenado
100%                                          44/44 [00:00<00:00, 142.49it/s]
Zona 264 - Mean Absolute Error: 0.9593442622950821
Entrenando para la zona: 265
Forecaster para la zona 265 entrenado
100%                                          44/44 [00:00<00:00, 117.24it/s]
Zona 265 - Mean Absolute Error: 2.2849180327868854
Entrenando para la zona: 266
Forecaster para la zona 266 entrenado
100%                                          44/44 [00:00<00:00, 142.48it/s]
Zona 266 - Mean Absolute Error: 2.425901639344262
Entrenando para la zona: 267
Forecaster para la zona 267 entrenado
100%                                          44/44 [00:00<00:00, 141.31it/s]
Zona 267 - Mean Absolute Error: 2.72327868852459
Entrenando para la zona: 268
Forecaster para la zona 268 entrenado
100%                                          44/44 [00:00<00:00, 144.98it/s]
Zona 268 - Mean Absolute Error: 4.3022950819672126
Entrenando para la zona: 269
Forecaster para la zona 269 entrenado
100%                                          44/44 [00:00<00:00, 129.51it/s]
Zona 269 - Mean Absolute Error: 3.0085245901639346
Entrenando para la zona: 270
Forecaster para la zona 270 entrenado
100%                                          44/44 [00:00<00:00, 146.10it/s]
Zona 270 - Mean Absolute Error: 1.6078688524590166
Entrenando para la zona: 271
Forecaster para la zona 271 entrenado
100%                                          44/44 [00:00<00:00, 133.12it/s]
Zona 271 - Mean Absolute Error: 1.7101639344262292
Entrenando para la zona: 282
Forecaster para la zona 282 entrenado
100%                                          44/44 [00:00<00:00, 44.72it/s]
Zona 282 - Mean Absolute Error: 2.728524590163935
Entrenando para la zona: 283
Forecaster para la zona 283 entrenado
100%                                          44/44 [00:00<00:00, 140.83it/s]
Zona 283 - Mean Absolute Error: 1.9485245901639343
Entrenando para la zona: 284
Forecaster para la zona 284 entrenado
100%                                          44/44 [00:00<00:00, 150.45it/s]
Zona 284 - Mean Absolute Error: 1.160327868852459
Entrenando para la zona: 285
Forecaster para la zona 285 entrenado
100%                                          44/44 [00:00<00:00, 127.43it/s]
Zona 285 - Mean Absolute Error: 2.3242622950819674
Entrenando para la zona: 286
Forecaster para la zona 286 entrenado
100%                                          44/44 [00:00<00:00, 123.39it/s]
Zona 286 - Mean Absolute Error: 2.3022950819672134
Entrenando para la zona: 287
Forecaster para la zona 287 entrenado
100%                                          44/44 [00:00<00:00, 112.56it/s]
Zona 287 - Mean Absolute Error: 2.734426229508197
Entrenando para la zona: 288
Forecaster para la zona 288 entrenado
100%                                          44/44 [00:00<00:00, 107.30it/s]
Zona 288 - Mean Absolute Error: 2.6219672131147544
Entrenando para la zona: 289
Forecaster para la zona 289 entrenado
100%                                          44/44 [00:00<00:00, 106.96it/s]
Zona 289 - Mean Absolute Error: 3.2347540983606557
Entrenando para la zona: 290
Forecaster para la zona 290 entrenado
100%                                          44/44 [00:00<00:00, 100.18it/s]
Zona 290 - Mean Absolute Error: 5.883934426229508
Entrenando para la zona: 291
Forecaster para la zona 291 entrenado
100%                                          44/44 [00:00<00:00, 107.11it/s]
Zona 291 - Mean Absolute Error: 2.9875409836065576
Entrenando para la zona: 292
Forecaster para la zona 292 entrenado
100%                                          44/44 [00:00<00:00, 117.65it/s]
Zona 292 - Mean Absolute Error: 2.5626229508196716
Entrenando para la zona: 303
Forecaster para la zona 303 entrenado
100%                                          44/44 [00:00<00:00, 147.41it/s]
Zona 303 - Mean Absolute Error: 1.1763934426229508
Entrenando para la zona: 304
Forecaster para la zona 304 entrenado
100%                                          44/44 [00:00<00:00, 158.24it/s]
Zona 304 - Mean Absolute Error: 1.478360655737705
```

```
Entrenando para la zona: 305
Forecaster para la zona 305 entrenado
100%                                          44/44 [00:00<00:00, 165.83it/s]
Zona 305 - Mean Absolute Error: 1.2124590163934426
Entrenando para la zona: 306
Forecaster para la zona 306 entrenado
100%                                          44/44 [00:00<00:00, 144.46it/s]
Zona 306 - Mean Absolute Error: 2.034754098360656
Entrenando para la zona: 307
Forecaster para la zona 307 entrenado
100%                                          44/44 [00:00<00:00, 136.90it/s]
Zona 307 - Mean Absolute Error: 3.103606557377049
Entrenando para la zona: 308
Forecaster para la zona 308 entrenado
100%                                          44/44 [00:00<00:00, 155.85it/s]
Zona 308 - Mean Absolute Error: 1.5177049180327868
Entrenando para la zona: 309
Forecaster para la zona 309 entrenado
100%                                          44/44 [00:00<00:00, 135.31it/s]
Zona 309 - Mean Absolute Error: 3.77016393442623
Entrenando para la zona: 310
Forecaster para la zona 310 entrenado
100%                                          44/44 [00:00<00:00, 136.04it/s]
Zona 310 - Mean Absolute Error: 4.748196721311475
Entrenando para la zona: 311
Forecaster para la zona 311 entrenado
100%                                          44/44 [00:00<00:00, 139.38it/s]
Zona 311 - Mean Absolute Error: 3.0721311475409836
Entrenando para la zona: 312
Forecaster para la zona 312 entrenado
100%                                          44/44 [00:00<00:00, 146.97it/s]
Zona 312 - Mean Absolute Error: 2.2678688524590163
Entrenando para la zona: 313
Forecaster para la zona 313 entrenado
100%                                          44/44 [00:00<00:00, 122.32it/s]
Zona 313 - Mean Absolute Error: 1.958688524590164
Entrenando para la zona: 314
Forecaster para la zona 314 entrenado
100%                                          44/44 [00:00<00:00, 145.42it/s]
Zona 314 - Mean Absolute Error: 1.0432786885245902
Entrenando para la zona: 327
Forecaster para la zona 327 entrenado
100%                                          44/44 [00:00<00:00, 139.54it/s]
Zona 327 - Mean Absolute Error: 0.9960655737704919
Entrenando para la zona: 328
Forecaster para la zona 328 entrenado
100%                                          44/44 [00:00<00:00, 125.65it/s]
Zona 328 - Mean Absolute Error: 2.482950819672131
Entrenando para la zona: 329
Forecaster para la zona 329 entrenado
100%                                          44/44 [00:00<00:00, 120.63it/s]
Zona 329 - Mean Absolute Error: 4.541639344262294
Entrenando para la zona: 330
Forecaster para la zona 330 entrenado
100%                                          44/44 [00:00<00:00, 125.11it/s]
Zona 330 - Mean Absolute Error: 3.359016393442622
Entrenando para la zona: 331
Forecaster para la zona 331 entrenado
100%                                          44/44 [00:00<00:00, 145.34it/s]
Zona 331 - Mean Absolute Error: 2.480655737704918
Entrenando para la zona: 332
Forecaster para la zona 332 entrenado
100%                                          44/44 [00:00<00:00, 152.59it/s]
Zona 332 - Mean Absolute Error: 1.4331147540983606
Entrenando para la zona: 333
Forecaster para la zona 333 entrenado
100%                                          44/44 [00:00<00:00, 155.79it/s]
Zona 333 - Mean Absolute Error: 1.3091803278688525
Entrenando para la zona: 334
Forecaster para la zona 334 entrenado
100%                                          44/44 [00:00<00:00, 151.46it/s]
Zona 334 - Mean Absolute Error: 1.363606557377049
Entrenando para la zona: 335
Forecaster para la zona 335 entrenado
100%                                          44/44 [00:00<00:00, 150.36it/s]
Zona 335 - Mean Absolute Error: 1.2744262295081967
Entrenando para la zona: 350
Forecaster para la zona 350 entrenado
100%                                          44/44 [00:00<00:00, 139.53it/s]
Zona 350 - Mean Absolute Error: 1.4163934426229507
Entrenando para la zona: 351
Forecaster para la zona 351 entrenado
100%                                          44/44 [00:00<00:00, 151.41it/s]
Zona 351 - Mean Absolute Error: 1.1039344262295083
Entrenando para la zona: 352
Forecaster para la zona 352 entrenado
100%                                          44/44 [00:00<00:00, 161.98it/s]
```

```
Zona 352 - Mean Absolute Error: 1.2347540983606557
Entrenando para la zona: 353
Forecaster para la zona 353 entrenado
100%                                        44/44 [00:00<00:00, 156.46it/s]
Zona 353 - Mean Absolute Error: 1.3540983606557377
Entrenando para la zona: 354
Forecaster para la zona 354 entrenado
100%                                        44/44 [00:00<00:00, 147.66it/s]
Zona 354 - Mean Absolute Error: 1.3252459016393443
Entrenando para la zona: 355
Forecaster para la zona 355 entrenado
100%                                        44/44 [00:00<00:00, 147.20it/s]
Zona 355 - Mean Absolute Error: 1.4431693989071037
Entrenando para la zona: 356
Forecaster para la zona 356 entrenado
100%                                        44/44 [00:00<00:00, 132.19it/s]
Zona 356 - Mean Absolute Error: 1.3095081967213114
Entrenando para la zona: 366
Forecaster para la zona 366 entrenado
100%                                        44/44 [00:00<00:00, 135.63it/s]
Zona 366 - Mean Absolute Error: 1.3124590163934424
Entrenando para la zona: 367
Forecaster para la zona 367 entrenado
100%                                        44/44 [00:00<00:00, 122.47it/s]
Zona 367 - Mean Absolute Error: 1.3262295081967213
Entrenando para la zona: 368
Forecaster para la zona 368 entrenado
100%                                        44/44 [00:00<00:00, 137.88it/s]
Zona 368 - Mean Absolute Error: 1.2318032786885245
Entrenando para la zona: 369
Forecaster para la zona 369 entrenado
100%                                        44/44 [00:00<00:00, 120.65it/s]
Zona 369 - Mean Absolute Error: 1.5252459016393445
Entrenando para la zona: 370
Forecaster para la zona 370 entrenado
100%                                        44/44 [00:00<00:00, 123.32it/s]
Zona 370 - Mean Absolute Error: 1.3163934426229509
Entrenando para la zona: 376
Forecaster para la zona 376 entrenado
100%                                        44/44 [00:00<00:00, 141.30it/s]
Zona 376 - Mean Absolute Error: 0.9740983606557377
Entrenando para la zona: 380
Forecaster para la zona 380 entrenado
100%                                        44/44 [00:00<00:00, 163.11it/s]
Zona 380 - Mean Absolute Error: 0.8804371584699453
Entrenando para la zona: 381
Forecaster para la zona 381 entrenado
100%                                        44/44 [00:00<00:00, 158.34it/s]
Zona 381 - Mean Absolute Error: 1.389726775956284
Entrenando para la zona: 382
Forecaster para la zona 382 entrenado
100%                                        44/44 [00:00<00:00, 145.51it/s]
Zona 382 - Mean Absolute Error: 1.2026229508196724
Entrenando para la zona: 383
Forecaster para la zona 383 entrenado
100%                                        44/44 [00:00<00:00, 154.66it/s]
Zona 383 - Mean Absolute Error: 1.0524590163934426
Entrenando para la zona: 384
Forecaster para la zona 384 entrenado
100%                                        44/44 [00:00<00:00, 141.63it/s]
Zona 384 - Mean Absolute Error: 1.880327868852459
Entrenando para la zona: 385
Forecaster para la zona 385 entrenado
100%                                        44/44 [00:00<00:00, 156.81it/s]
Zona 385 - Mean Absolute Error: 1.3613114754098363
Entrenando para la zona: 386
Forecaster para la zona 386 entrenado
100%                                        44/44 [00:00<00:00, 162.51it/s]
Zona 386 - Mean Absolute Error: 0.9373770491803278
Entrenando para la zona: 387
Forecaster para la zona 387 entrenado
100%                                        44/44 [00:00<00:00, 133.22it/s]
Zona 387 - Mean Absolute Error: 1.9921311475409833
Entrenando para la zona: 388
Forecaster para la zona 388 entrenado
100%                                        44/44 [00:00<00:00, 143.45it/s]
Zona 388 - Mean Absolute Error: 1.7800000000000002
Entrenando para la zona: 389
Forecaster para la zona 389 entrenado
100%                                        44/44 [00:00<00:00, 137.70it/s]
Zona 389 - Mean Absolute Error: 1.7403278688524588
Entrenando para la zona: 390
Forecaster para la zona 390 entrenado
100%                                        44/44 [00:00<00:00, 109.04it/s]
Zona 390 - Mean Absolute Error: 2.8659016393442625
Entrenando para la zona: 391
Forecaster para la zona 391 entrenado
100%                                        44/44 [00:00<00:00, 157.70it/s]
```

```
Zona 391 - Mean Absolute Error: 1.1772131147540983
Entrenando para la zona: 401
Forecaster para la zona 401 entrenado
100%                                        44/44 [00:00<00:00, 144.77it/s]
Zona 401 - Mean Absolute Error: 1.364918032786885
Entrenando para la zona: 402
Forecaster para la zona 402 entrenado
100%                                        44/44 [00:00<00:00, 137.49it/s]
Zona 402 - Mean Absolute Error: 2.605901639344262
Entrenando para la zona: 403
Forecaster para la zona 403 entrenado
100%                                        44/44 [00:00<00:00, 129.41it/s]
Zona 403 - Mean Absolute Error: 1.8737704918032787
Entrenando para la zona: 404
Forecaster para la zona 404 entrenado
100%                                        44/44 [00:00<00:00, 133.10it/s]
Zona 404 - Mean Absolute Error: 1.7380327868852457
Entrenando para la zona: 405
Forecaster para la zona 405 entrenado
100%                                        44/44 [00:00<00:00, 131.16it/s]
Zona 405 - Mean Absolute Error: 1.9298360655737703
Entrenando para la zona: 406
Forecaster para la zona 406 entrenado
100%                                        44/44 [00:00<00:00, 138.47it/s]
Zona 406 - Mean Absolute Error: 1.3272131147540984
Entrenando para la zona: 407
Forecaster para la zona 407 entrenado
100%                                        44/44 [00:00<00:00, 144.05it/s]
Zona 407 - Mean Absolute Error: 1.7088524590163936
Entrenando para la zona: 408
Forecaster para la zona 408 entrenado
100%                                        44/44 [00:00<00:00, 143.02it/s]
Zona 408 - Mean Absolute Error: 2.4662295081967214
Entrenando para la zona: 409
Forecaster para la zona 409 entrenado
100%                                        44/44 [00:00<00:00, 142.81it/s]
Zona 409 - Mean Absolute Error: 1.8245901639344262
Entrenando para la zona: 410
Forecaster para la zona 410 entrenado
100%                                        44/44 [00:00<00:00, 147.48it/s]
Zona 410 - Mean Absolute Error: 1.9295081967213115
Entrenando para la zona: 411
Forecaster para la zona 411 entrenado
100%                                        44/44 [00:00<00:00, 140.32it/s]
Zona 411 - Mean Absolute Error: 1.9006557377049182
Entrenando para la zona: 423
Forecaster para la zona 423 entrenado
100%                                        44/44 [00:00<00:00, 141.32it/s]
Zona 423 - Mean Absolute Error: 1.52
Entrenando para la zona: 424
Forecaster para la zona 424 entrenado
100%                                        44/44 [00:00<00:00, 139.12it/s]
Zona 424 - Mean Absolute Error: 1.5081967213114753
Entrenando para la zona: 425
Forecaster para la zona 425 entrenado
100%                                        44/44 [00:00<00:00, 140.50it/s]
Zona 425 - Mean Absolute Error: 1.400655737704918
Entrenando para la zona: 426
Forecaster para la zona 426 entrenado
100%                                        44/44 [00:00<00:00, 144.35it/s]
Zona 426 - Mean Absolute Error: 1.8337704918032784
Entrenando para la zona: 427
Forecaster para la zona 427 entrenado
100%                                        44/44 [00:00<00:00, 149.00it/s]
Zona 427 - Mean Absolute Error: 1.2108196721311475
Entrenando para la zona: 428
Forecaster para la zona 428 entrenado
100%                                        44/44 [00:00<00:00, 103.09it/s]
Zona 428 - Mean Absolute Error: 1.5849180327868853
Entrenando para la zona: 429
Forecaster para la zona 429 entrenado
100%                                        44/44 [00:00<00:00, 123.92it/s]
Zona 429 - Mean Absolute Error: 2.6055737704918034
Entrenando para la zona: 430
Forecaster para la zona 430 entrenado
100%                                        44/44 [00:00<00:00, 137.57it/s]
Zona 430 - Mean Absolute Error: 1.5475409836065572
Entrenando para la zona: 431
Forecaster para la zona 431 entrenado
100%                                        44/44 [00:00<00:00, 137.54it/s]
Zona 431 - Mean Absolute Error: 1.382950819672131
Entrenando para la zona: 432
Forecaster para la zona 432 entrenado
100%                                        44/44 [00:00<00:00, 130.48it/s]
Zona 432 - Mean Absolute Error: 1.5445901639344264
Entrenando para la zona: 433
Forecaster para la zona 433 entrenado
100%                                        44/44 [00:00<00:00, 132.87it/s]
```

100% 44/44 [00:00<00:00, 135.27it/s]

Zona 433 - Mean Absolute Error: 0.9724590163934427
Entrenando para la zona: 444
Forecaster para la zona 444 entrenado

100% 44/44 [00:00<00:00, 151.66it/s]

Zona 444 - Mean Absolute Error: 0.9642076502732242
Entrenando para la zona: 445
Forecaster para la zona 445 entrenado

100% 44/44 [00:00<00:00, 149.83it/s]

Zona 445 - Mean Absolute Error: 1.3232786885245902
Entrenando para la zona: 446
Forecaster para la zona 446 entrenado

100% 44/44 [00:00<00:00, 134.37it/s]

Zona 446 - Mean Absolute Error: 1.7419672131147539
Entrenando para la zona: 447
Forecaster para la zona 447 entrenado

100% 44/44 [00:00<00:00, 139.05it/s]

Zona 447 - Mean Absolute Error: 1.359672131147541
Entrenando para la zona: 448
Forecaster para la zona 448 entrenado

100% 44/44 [00:00<00:00, 156.86it/s]

Zona 448 - Mean Absolute Error: 1.1337704918032785
Entrenando para la zona: 449
Forecaster para la zona 449 entrenado

100% 44/44 [00:00<00:00, 151.29it/s]

Zona 449 - Mean Absolute Error: 1.1501639344262293
Entrenando para la zona: 450
Forecaster para la zona 450 entrenado

100% 44/44 [00:00<00:00, 145.17it/s]

Zona 450 - Mean Absolute Error: 2.2081967213114755
Entrenando para la zona: 451
Forecaster para la zona 451 entrenado

100% 44/44 [00:00<00:00, 151.35it/s]

Zona 451 - Mean Absolute Error: 1.3934426229508197
Entrenando para la zona: 452
Forecaster para la zona 452 entrenado

100% 44/44 [00:00<00:00, 138.58it/s]

Zona 452 - Mean Absolute Error: 1.2475409836065574
Entrenando para la zona: 453
Forecaster para la zona 453 entrenado

100% 44/44 [00:00<00:00, 153.64it/s]

Zona 453 - Mean Absolute Error: 0.8120765027322403
Entrenando para la zona: 457
Forecaster para la zona 457 entrenado

100% 44/44 [00:00<00:00, 159.81it/s]

Zona 457 - Mean Absolute Error: 1.0111475409836066
Entrenando para la zona: 465
Forecaster para la zona 465 entrenado

100% 44/44 [00:00<00:00, 179.41it/s]

Zona 465 - Mean Absolute Error: 0.7773989071038251
Entrenando para la zona: 466
Forecaster para la zona 466 entrenado

100% 44/44 [00:00<00:00, 165.61it/s]

Zona 466 - Mean Absolute Error: 0.9491803278688524
Entrenando para la zona: 468
Forecaster para la zona 468 entrenado

100% 44/44 [00:00<00:00, 140.24it/s]

Zona 468 - Mean Absolute Error: 1.1314754098360655
Entrenando para la zona: 469
Forecaster para la zona 469 entrenado

100% 44/44 [00:00<00:00, 129.03it/s]

Zona 469 - Mean Absolute Error: 1.4495081967213115
Entrenando para la zona: 470
Forecaster para la zona 470 entrenado

100% 44/44 [00:00<00:00, 134.44it/s]

Zona 470 - Mean Absolute Error: 1.1918032786885246
Entrenando para la zona: 471
Forecaster para la zona 471 entrenado

100% 44/44 [00:00<00:00, 143.95it/s]

Zona 471 - Mean Absolute Error: 1.0996721311475408
Entrenando para la zona: 472
Forecaster para la zona 472 entrenado

100% 44/44 [00:00<00:00, 130.48it/s]

Zona 472 - Mean Absolute Error: 1.618032786885246
Entrenando para la zona: 473
Forecaster para la zona 473 entrenado

100% 44/44 [00:00<00:00, 121.54it/s]

Zona 473 - Mean Absolute Error: 1.521311475409836
Entrenando para la zona: 477
Forecaster para la zona 477 entrenado

100% 44/44 [00:00<00:00, 159.75it/s]

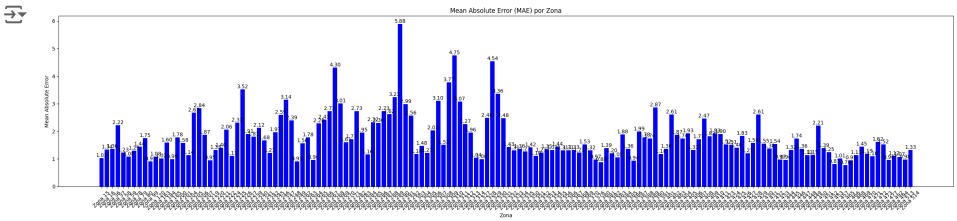
Zona 477 - Mean Absolute Error: 0.9583060109289616
Entrenando para la zona: 492
Forecaster para la zona 492 entrenado

100% 44/44 [00:00<00:00, 151.65it/s]

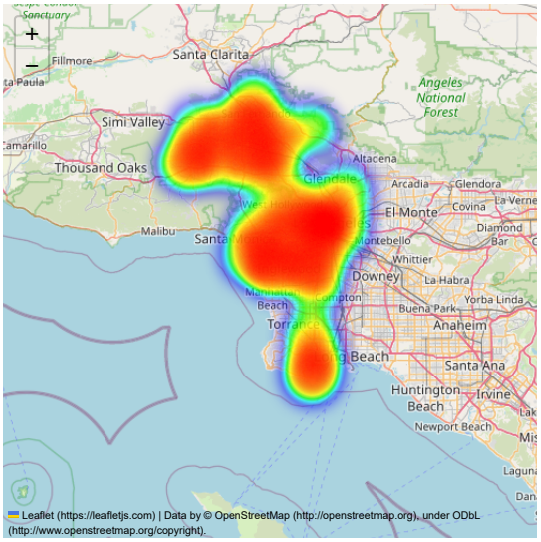
Zona 492 - Mean Absolute Error: 1.1167213114754098
Entrenando para la zona: 494
Forecaster para la zona 494 entrenado

```
100%                                                44/44 [00:00<00:00, 174.55it/s]
Zona 494 - Mean Absolute Error: 1.0678688524590163
Entrenando para la zona: 513
Forecaster para la zona 513 entrenado
100%                                                44/44 [00:00<00:00, 162.63it/s]
Zona 513 - Mean Absolute Error: 0.9781147540983608
Entrenando para la zona: 514
Forecaster para la zona 514 entrenado
100%                                                44/44 [00:00<00:00, 141.20it/s]
```

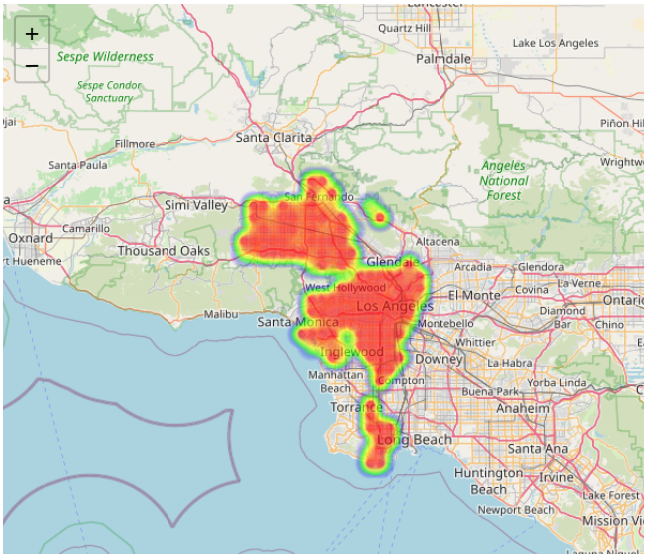
```
# Gráfica de las métricas (Mean Absolute Error) para cada zona
zonas_labels = [f'Zona {zona}' for zona in zonas]
metricas_values = [mae for mae in metricas_array]

fig, ax = plt.subplots(figsize=(25, 6))

barras = ax.bar(zonas_labels, metricas_values, color='blue')

ax.bar(zonas_labels, metricas_values, color='blue')
ax.set_title('Mean Absolute Error (MAE) por Zona')
ax.set_xlabel('Zona')
ax.set_ylabel('Mean Absolute Error')
ax.tick_params(axis='x', rotation=45)

# Añadir el valor del MAE encima de cada barra
for barra in barras:
    altura = barra.get_height()
    ax.text(barra.get_x() + barra.get_width() / 2, altura, f'{altura:.2f}', ha='center', va='bottom')


plt.tight_layout()

plt.show()
```




```
#Generación de mapa

import folium
from folium.plugins import HeatMap
import geopandas as gpd


#Leemos el archivo shape y la cantidad de delitos por celda
gdf = gpd.read_file('la_grid.shp')

# Crear el mapa centrado en la ubicación aproximada de Los Ángeles
m = folium.Map(width=500,height=500, location=[34.0522, -118.2437], zoom_start=10)


heat_data = []
for idx, zona in enumerate(zonas):
    geom = gdf.loc[zona-1,'geometry']
    x, y = geom.centroid.y, geom.centroid.x  # Folium utiliza latitud y longitud en este orden
    predicciones = predicciones_array[zona]
    entrenamiento = entrenamiento_array[zona]
    total_delitos = sum(entrenamiento['Cantidad de Crimenes']) + sum(predicciones['pred'])  # Obtener el total de delitos en la celda
    if total_delitos > 0:
      heat_data.append([x, y, total_delitos])  # Añadir la latitud, longitud y el valor a la lista

# Añadir el HeatMap al mapa
HeatMap(heat_data).add_to(m)

# Guardar el mapa en un archivo HTML
m.save('mapa_predictivo_50_geoceldas.html')

m
```

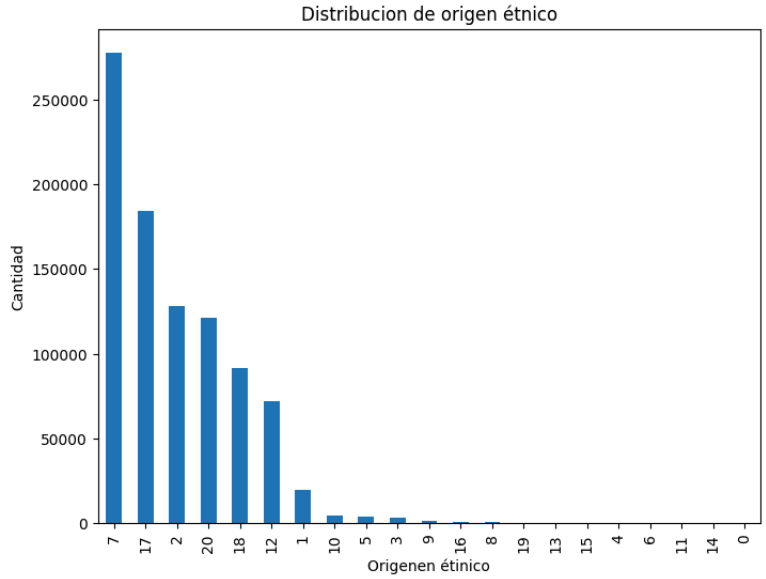

Santa Clarita
Fillmore
Simi Valley
Thousand Oaks
Malibu
Angeles National Forest
Altacena
Glendale
Arcadia
Glendora
El Monte
Covina
Diamond Bar
Montebello
Whittier
Downey
La Habra
Buena Park
Yorba Linda
Anaheim
Santa Ana
Irvine
Huntington Beach
Newport Beach
Torrance
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

```
import folium
from folium.plugins import HeatMapWithTime
import geopandas as gpd
import numpy as np

# Leer el archivo shape y la cantidad de delitos por celda
gdf = gpd.read_file('la_grid.shp')


zona_totales = df.groupby('Zona')['Cantidad de Crimenes'].sum().sort_values(ascending=False)


n = 150
top_n_zonas = zona_totales.head(n).index
df_top_n = data[data['Zona'].isin(top_n_zonas)]
zonas = df_top_n['Zona'].unique()

# Crear el mapa centrado en la ubicación aproximada de Los Ángeles
crime_map = folium.Map(width=600, height=600, location=[34.0522, -118.2437], zoom_start=12)

heat_data = []

# Crear una lista de tiempos (fechas únicas) en formato de cadena para HeatMapWithTime
time_index = sorted(data['Fecha'].dt.strftime('%Y-%m-%d').unique())


for time in time_index:
    heat_data_time = []
    for idx, row in gdf.iterrows():
        zona = row['FID'] + 1  # Asegurarse de que estamos usando la columna correcta para la zona
        if zona in zonas:  # Usar el nombre de la zona en lugar del índice
            geom = row['geometry']
            x, y = geom.centroid.y, geom.centroid.x  # Folium utiliza latitud y longitud en este orden

            predicciones = predicciones_array[zona]
            entrenamiento = entrenamiento_array[zona]

            # Actualizar el conjunto de entrenamiento con las predicciones
            N = entrenamiento.shape[0]
            Ntrain = int(0.8 * N)

            entrenamiento.iloc[Ntrain:, entrenamiento.columns.get_loc('Cantidad de Crimenes')] = predicciones['pred'].values

            # Filtrar los datos de entrenamiento y predicción para el tiempo actual
            entrenamiento_time = entrenamiento[entrenamiento.index.strftime('%Y-%m-%d') == time]

            if not entrenamiento_time.empty:
                total_delitos = int(entrenamiento_time['Cantidad de Crimenes'].iloc[0])  # Obtener el total de delitos en la celda

                # Calcular la intensidad basada en total_delitos con escala logarítmica
                if total_delitos > 0:
                    intensidad = np.log(total_delitos)  # Aplicar la escala logarítmica
                else:
                    intensidad = 0
                heat_data_time.append([x, y, intensidad])  # Añadir la latitud, longitud, radio a la lista de tiempo actual

    if heat_data_time:  # Solo añadir si hay datos para el tiempo actual
        heat_data.append(heat_data_time)

# Crear una lista para los radios en HeatMapWithTime
#heat_data_with_radius = [[[x, y, value] for x, y, value, radius in time] for time in heat_data]

# Añadir el HeatMapWithTime al mapa con el radio especificado
hm = HeatMapWithTime(heat_data, index=time_index, auto_play=True, min_speed=1,max_speed=20, speed_step=1.0) # Paso de velocidad)
hm.add_to(crime_map)
crime_map.save('mapa_predictivo_serie_de_tiempo.html')
crime_map
```



Modelo de predicción 2: RandomForestRegressor

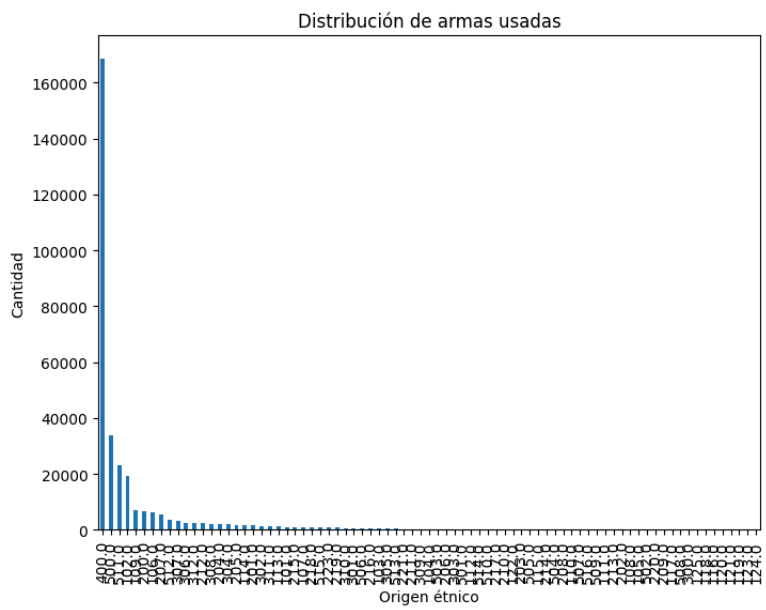
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt


url_los_angeles='Crime_Data_from_2020_to_Present.csv'
heatmap_df = pd.read_csv(url_los_angeles)
heatmap_df['DATE OCC'] = pd.to_datetime(heatmap_df['DATE OCC'])


heatmap_columns = ['TIME OCC', 'AREA', 'Part 1-2', 'Crm Cd', 'Vict Age', 'Vict Sex', 'Vict Descent', 'Premis Cd','Weapon Used Cd', 'LAT', 'LON']
heatmap_df = heatmap_df[heatmap_columns]
label_encoder = LabelEncoder()
heatmap_df['Vict Sex'] = label_encoder.fit_transform(heatmap_df['Vict Sex'])
heatmap_df['Vict Descent'] = label_encoder.fit_transform(heatmap_df['Vict Descent'])


# Comprender la distribución del origen etnico
class_distribution = heatmap_df['Vict Descent'].value_counts()
plt.figure(figsize=(8, 6))
class_distribution.plot(kind='bar')
plt.xlabel('Origenen étinico')
plt.ylabel('Cantidad')
plt.title('Distribucion de origen étnico')
plt.show()
```



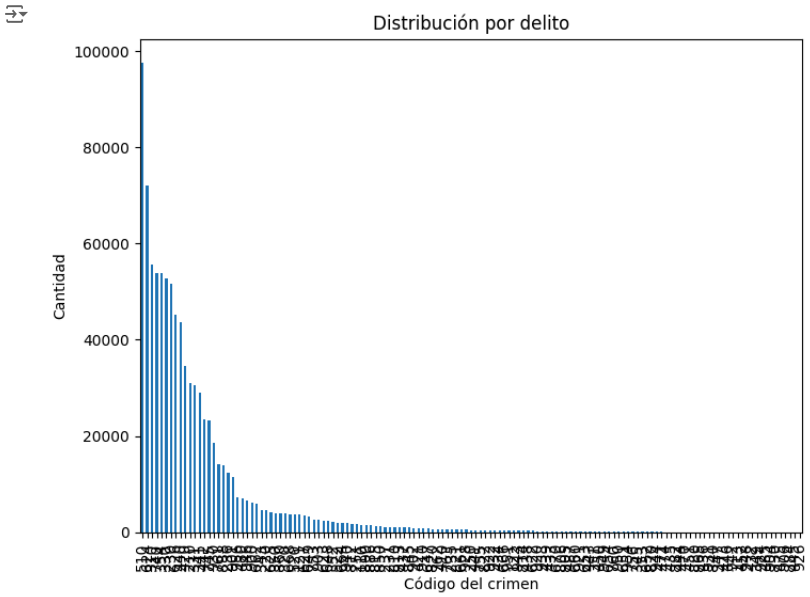

```
# Seleccionamos los seis tipos de origen étnicos
origen_etnico = [7, 17, 2, 20, 18, 12, 1]
# Filter the DataFrame to keep rows with desired values
heatmap_df = heatmap_df[heatmap_df['Vict Descent'].isin(origen_etnico)]
heatmap_df.head()
```

	TIME OCC	AREA	Part 1-2	Crm Cd	Vict Age	Vict Sex	Vict Descent	Premis Cd	Weapon Used Cd	LAT	LON
0	2130	7	1	510	0	3	12	101.0	NaN	34.0375	-118.3506
1	1800	1	1	330	47	3	12	128.0	NaN	34.0444	-118.2628
2	1700	3	1	480	19	4	18	502.0	NaN	34.0210	-118.3002
3	2037	9	1	343	19	3	12	405.0	NaN	34.1576	-118.4387
4	1200	6	2	354	28	3	7	102.0	NaN	34.0944	-118.3277

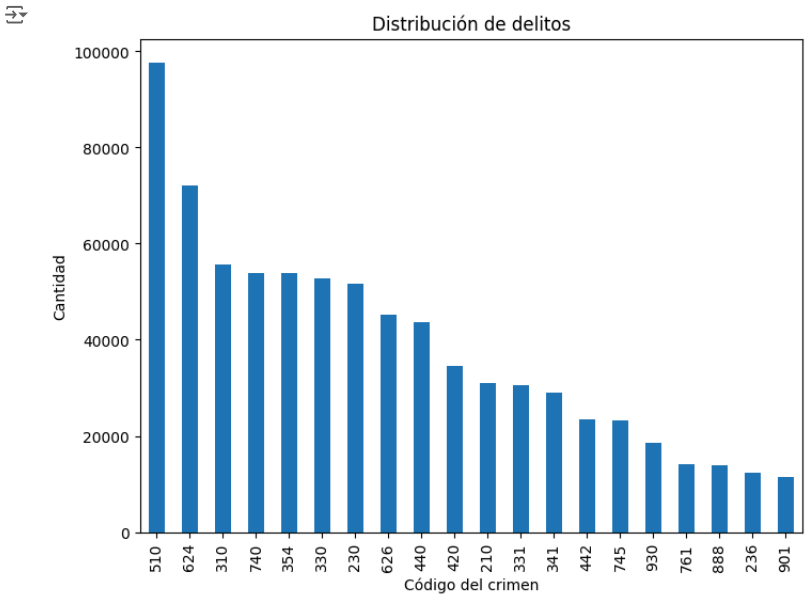
```
# Distribución de armas usadas
class_distribution = heatmap_df['Weapon Used Cd'].value_counts()
plt.figure(figsize=(8, 6))
class_distribution.plot(kind='bar')
plt.xlabel('Origen étnico')
plt.ylabel('Cantidad')
plt.title('Distribución de armas usadas')
plt.show()
```



```
class_distribution = heatmap_df['Crm Cd'].value_counts()
# Plot the class distribution
plt.figure(figsize=(8, 6))
class_distribution.plot(kind='bar')
plt.xlabel('Código del crimen')
plt.ylabel('Cantidad')
plt.title('Distribución por delito')
plt.show()
```




```
# Filtraremos los delitos con código mayor a 1000
class_distribution = heatmap_df['Crm Cd'].value_counts()
class_distribution = class_distribution[class_distribution > 10000]
class_codes = class_distribution.index.tolist()
heatmap_df = heatmap_df[heatmap_df['Crm Cd'].isin(class_codes)]
plt.figure(figsize=(8, 6))
class_distribution.plot(kind='bar')
plt.xlabel('Código del crimen')
plt.ylabel('Cantidad')
plt.title('Distribución de delitos')
plt.show()
```

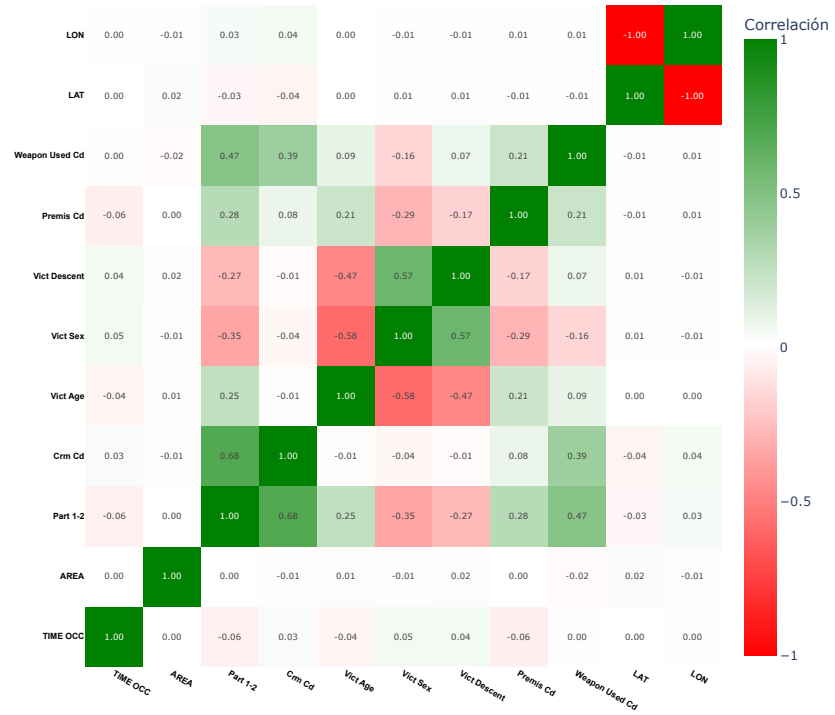



```
heatmap_df.info()
```

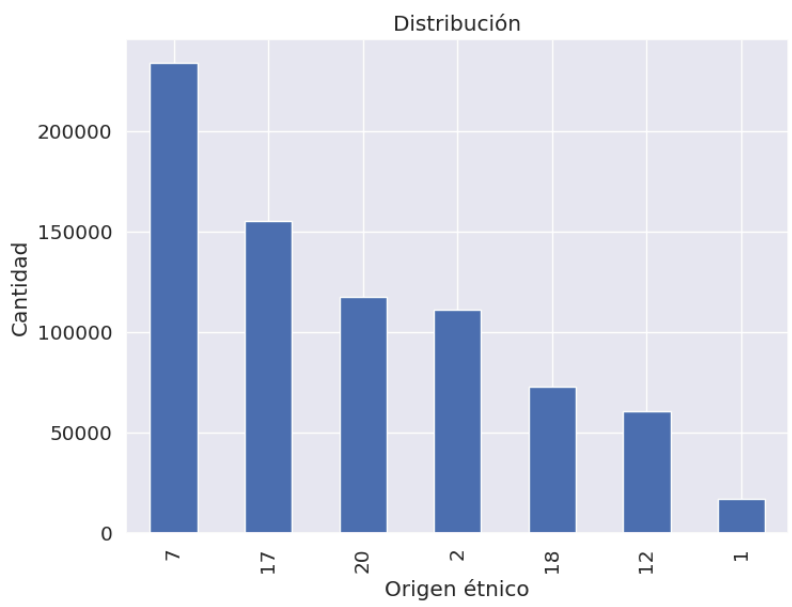
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 910707 entries, 0 to 910706
Data columns (total 11 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   TIME OCC        910707 non-null  int64
 1   AREA            910707 non-null  int64
 2   Part 1-2        910707 non-null  int64
 3   Crm Cd          910707 non-null  int64
 4   Vict Age        910707 non-null  int64
 5   Vict Sex        910707 non-null  int64
 6   Vict Descent    910707 non-null  int64
```

```
 7   Premis Cd       910697 non-null  float64
 8   Weapon Used Cd  315247 non-null  float64
 9   LAT             910707 non-null  float64
 10  LON             910707 non-null  float64
dtypes: float64(4), int64(7)
memory usage: 76.4 MB
```

```
import plotly.graph_objects as go
import pandas as pd

# Supongamos que heatmap_df y heatmap_columns ya están definidos
# heatmap_df = ...
# heatmap_columns = ...

# Calcular la matriz de correlación
correlation_matrix = heatmap_df[heatmap_columns].corr()

# Definir la escala de colores personalizada
colorscale = [
    [0, 'red'],    # Valor mínimo (correlación -1) en rojo
    [0.5, 'white'], # Valor medio (correlación 0) en blanco
    [1, 'green']   # Valor máximo (correlación 1) en verde
]

# Crear el heatmap usando Plotly
fig = go.Figure(data=go.Heatmap(
    z=correlation_matrix.values,
    x=correlation_matrix.columns,
    y=correlation_matrix.columns,
    colorscale=colorscale,  # Usar la escala de colores personalizada
    colorbar=dict(title='Correlación'),
    zmin=-1, zmax=1,  # Establecer el rango de la escala de color
    text=correlation_matrix.values,  # Añadir los valores de la matriz como texto
    texttemplate="%{text:.2f}",  # Formatear el texto con dos decimales
    hoverinfo='z'  # Mostrar solo el valor de correlación en el hover
))

# Añadir el título y ajustar el tamaño de la fuente
fig.update_layout(
    title='Mapa de Correlación',
    xaxis=dict(
        tickmode='array',
        tickvals=list(range(len(correlation_matrix.columns))),
        ticktext=[f'<b>{col}</b>' for col in correlation_matrix.columns],  # Etiquetas en negrita
        tickfont=dict(size=8, family='Arial', color='black')  # Ajustar el tamaño de la fuente
    ),
    yaxis=dict(
        tickmode='array',
        tickvals=list(range(len(correlation_matrix.columns))),
        ticktext=[f'<b>{col}</b>' for col in correlation_matrix.columns],  # Etiquetas en negrita
        tickfont=dict(size=8, family='Arial', color='black')  # Ajustar el tamaño de la fuente
    ),
    width=800,
    height=800
)

# Ajustar el tamaño de la fuente de los valores en el heatmap
fig.update_traces(
    textfont=dict(size=8)
)

# Mostrar la figura
fig.show()
```




```
class_distribution = heatmap_df['Vict Descent'].value_counts()
plt.figure(figsize=(8, 6))
class_distribution.plot(kind='bar')
plt.xlabel('Origen étnico')
plt.ylabel('Cantidad')
plt.title('Distribución')
plt.show()
```




```
columns_to_use = [ 'AREA','Crm Cd', 'Vict Age', 'Vict Sex', 'Vict Descent']
data_selected = heatmap_df[columns_to_use]
data_selected = data_selected.dropna()
data_selected.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 910707 entries, 0 to 910706
Data columns (total 5 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   AREA          910707 non-null  int64
 1   Crm Cd        910707 non-null  int64
 2   Vict Age      910707 non-null  int64
```

```
 3   Vict Sex      910707 non-null  int64
 4   Vict Descent  910707 non-null  int64
dtypes: int64(5)
memory usage: 34.7 MB
```

```
data_selected.shape
```

```
(910707, 5)
```

Estandarización

```
from sklearn.preprocessing import StandardScaler
scaler_ss = StandardScaler()
data_ss = scaler_ss.fit_transform(data_selected)
data_ss = pd.DataFrame(data_ss, columns=data_selected.columns)
data_ss
```

	AREA	Crm Cd	Vict Age	Vict Sex	Vict Descent
0	-0.606123	0.044267	-1.354066	0.267904	0.087784
1	-1.589374	-0.822760	0.796771	0.267904	0.087784
2	-1.261623	-0.100237	-0.484579	0.984871	1.015642
3	-0.278372	-0.760141	-0.484579	0.267904	0.087784
4	-0.769998	-0.707156	-0.072716	0.267904	-0.685431
...	...	...	...	...	...
910702	1.688129	-0.389246	0.659484	-1.166031	0.087784
910703	0.868754	0.598202	0.567959	-1.166031	-0.685431
910704	1.524254	0.593385	2.306934	-1.166031	0.087784
910705	-0.606123	0.044267	-1.354066	1.701838	1.324928
910706	-1.589374	1.176220	-1.354066	0.984871	1.015642

910707 rows × 5 columns

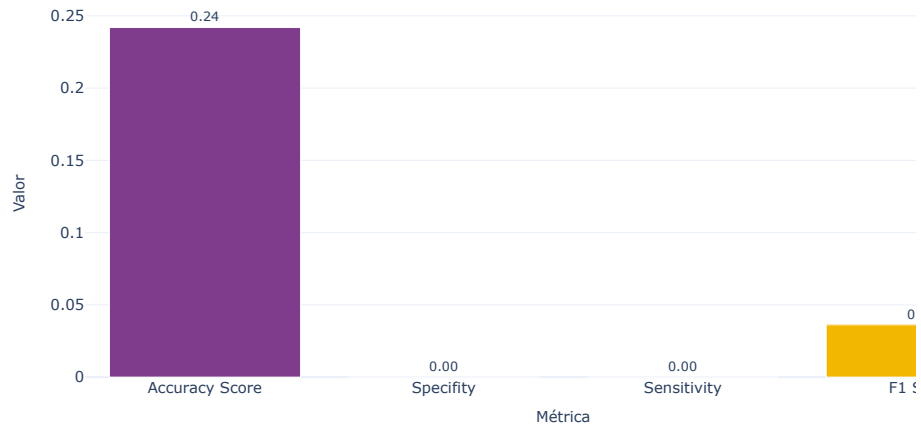
```
# La variable a predecir es el código del crímen
X = data_ss.drop('Crm Cd', axis=1)
y = data_selected['Crm Cd']
```

Predicción con algortimo Random Forest

```
from sklearn.ensemble import RandomForestClassifier

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
modelo1 = RandomForestClassifier()
modelo1.fit(X_train, y_train)
y_pred = modelo1.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.24192113845241625
```

```
import plotly.express as px
import pandas as pd
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score

# Calcular las métricas
a_s = accuracy_score(y_test, y_pred)
r_s = recall_score(y_test, y_pred, average=None)
sp_s = recall_score(y_test, y_pred, average=None)[0]
se_s = recall_score(y_test, y_pred, average=None)[1]
p_s = precision_score(y_test, y_pred, average=None)
f1 = f1_score(y_test, y_pred, average='macro')

# Crear el DataFrame con las métricas
d = {'Score':  ['Accuracy Score', 'Specifity', 'Sensitivity', 'F1 Score'],
     'Valor': [a_s, sp_s, se_s, f1]}
scores1 = pd.DataFrame(data=d)

# Crear la gráfica con Plotly
fig = px.bar(scores1, x='Score', y='Valor',
             text='Valor',
             title='Métricas de Evaluación del Modelo',
             labels={'Valor': 'Valor', 'Score': 'Métrica'},
             template='plotly_white',
             color='Score',
             color_discrete_sequence=px.colors.qualitative.Bold)

# Ajustar el tamaño del texto en las barras y el título
fig.update_traces(texttemplate='%{text:.2f}', textposition='outside', textfont_size=12)
fig.update_layout(title_font_size=24, xaxis_tickfont_size=14, yaxis_tickfont_size=14)

# Mostrar la figura
fig.show()
```



X_test

	AREA	Vict Age	Vict Sex	Vict Descent
800582	-1.425498	0.888296	-1.166031	-0.685431
388347	0.213253	-1.354066	0.984871	1.015642
688805	0.377128	0.613721	-1.166031	0.860999
682859	-1.425498	-0.164241	0.267904	-0.685431
69716	-0.769998	-0.026954	0.267904	-0.685431
...	...	...	...	...
23743	-0.442248	0.476434	-1.166031	0.860999
901919	-1.589374	1.208634	-1.166031	-1.458645
275288	0.213253	-1.354066	1.701838	1.324928
100939	0.704878	-0.164241	0.267904	-1.458645
733571	1.688129	0.339146	-1.166031	-0.685431

182142 rows × 4 columns

```
columns_to_use = [ 'AREA','Crm Cd', 'Vict Age', 'Vict Sex', 'Vict Descent']
data_selected = heatmap_df[columns_to_use]
data_selected = data_selected.dropna()
data_selected.info()

# Escalar los datos
scaler_X = StandardScaler()
scaler_y = StandardScaler()

# La variable a predecir es el código del crímen
X = data_selected.drop('Crm Cd', axis=1)
y = data_selected['Crm Cd']

X_scaled = scaler_X.fit_transform(X)
y_scaled = scaler_y.fit_transform(y.values.reshape(-1, 1))

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_scaled, test_size=0.2, random_state=42)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 910707 entries, 0 to 910706
Data columns (total 5 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
```