UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

# Predicción del crimen y patrullaje predictivo: una aplicación en la lucha contra la delincuencia

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: El Aouni, Fatima Zahra

Tutor/a: Vázquez Barrachina, Elena

Cotutor/a: Chirivella González, Vicente

Cotutor/a: Alcover Arandiga, Rosa María

CURSO ACADÉMICO: 2022/2023

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

# Predicción del crimen y patrullaje predictivo: un nuevo enfoque en la lucha contra la delincuencia

Trabajo Fin de Grado

**Grado en Ciencia de datos**

**Autor**: Fatima Zahra El Aouni

**Tutor**: Elena Barrachina, Rosa Alcover, Vicente Chirivella

Curso 2022-2023

Resumen

En este estudio de investigación, se exploró la predicción del crimen utilizando enfoques de modelado basados en redes neuronales. Se realizaron análisis comparativos entre dos tipos de redes neuronales: MLP (Perceptrón Multicapa) y Redes Neuronales Keras Regressor con Atención. El objetivo principal fue evaluar si la incorporación de atención mejoraba la precisión en la predicción del crimen utilizando datos de la ciudad de San Francisco.

Inicialmente, se realizó un análisis de clustering con el propósito de identificar clusters o grupos de características similares en los datos de crimen. Los resultados de aplicar un análisis de clustering a los datos, no revelaron la existencia de clusters o grupos de características similares en los datos de crimen, por lo que se utilizó la estructura geográfica de los distritos policiales como grupos de referencia para desarrollar modelos específicos. Esta aproximación permitió tener en cuenta las particularidades y patrones individuales de cada distrito. Se pretendió crear modelos de predicción del crimen para cada distrito policial, con el objetivo de reducir la variabilidad en las predicciones.

Los resultados obtenidos mostraron que esta estrategia de modelos por distrito policial fue efectiva para abordar la variabilidad en la comisión de delitos. Al considerar las características específicas de cada distrito, los modelos lograron capturar mejor los patrones y las tendencias del crimen en San Francisco. Se observó una mejora significativa en la precisión de la predicción del crimen en comparación con el enfoque tradicional de MLP.

**Palabras clave:** *Transformer*, MLP, *Machine Learning*, crimen, delito, incidente, *OpenDataSF*, San Francisco, atención, *Keras Regressor*, *Grid Search,* distrito policial

# Abstract

In this research study, crime prediction was explored using neural network-based modeling approaches. Comparative analyses were conducted between two types of neural networks: MLP (Multilayer Perceptron) and Neural Networks with Attention using Keras Regressor. The main objective was to evaluate if the incorporation of attention improved the accuracy in crime prediction using data from the city of San Francisco.

Initially, an attempt was made to perform clustering analysis with the purpose of identifying clusters or groups with similar characteristics in the crime data. However, the results did not reveal the existence of clusters, and the geographical structure of police districts was used as reference groups to develop specific models. This approach allowed for the consideration of individual district peculiarities and patterns. The intention was to create crime prediction models for each police district, aiming to reduce variability in the predictions.

The results obtained showed that this strategy of district-specific models was effective in addressing variability in the predictions. By considering the specific characteristics of each district, the models better captured crime patterns and trends in San Francisco. A significant improvement in the accuracy of crime prediction was observed compared to the traditional MLP approach.

**Keywords:** Transformer, MLP, Machine Learning, crime, offense, incident, OpenDataSF, San Francisco, attention, Keras Regressor, GridSearch, police district

# Agradecimientos

Me gustaría aprovechar esta oportunidad para expresar mi más sincero agradecimiento a todas las personas, familiares y amigos, que han sido parte fundamental de mi trayectoria durante la realización de mis estudios en la universidad y durante el desarrollo de este trabajo.

A la familia Ramírez en especial por haberme enseñado a luchar y trabajar duro por mis sueños independientemente de los millones de obstáculos que se me presenten, por haber estado ahí para mí y por mí, también por haberme inculcado valores que sin haberme cruzado con ellos en el camino nunca hubiera adquirido y por haber sido el refugio en el que sentirme segura a pesar de toda la inestabilidad que me rodeara.

Quiero agradecer también a mis profesores, quienes no solo me han impartido conocimientos valiosos, sino que también han sido un ejemplo a seguir y una fuente constante de inspiración, su dedicación, pasión por la enseñanza y disponibilidad para resolver mis dudas, que han sido fundamentales para mi crecimiento académico y personal. En especial a Elena Barrachina, Rosa Maria Alcover y Vicente Chirivella por haber velado por el desarrollo de este proyecto, por haber dedicado tiempo fuera de horario laboral y por los ánimos y apoyo que me han brindado.

Y en último lugar quisiera aprovechar este momento para expresar mi más profundo agradecimiento a la persona más importante en mi vida: mi hermana pequeña. Tú has sido mi mayor motivación para levantarme cada mañana y dedicar tiempo a mi carrera estudiantil y profesional, con el objetivo de convertirme en tu ejemplo e inspiración a seguir.

A ti, mi querida hermana, quiero agradecerte las risas compartidas en momentos difíciles. Tu alegría y positividad han llenado mi corazón de fuerza y energía para seguir adelante, incluso cuando los desafíos parecían abrumadores.

Tus sonrisas han sido un recordatorio constante de por qué he luchado tan arduamente para lograr mis metas. Cada paso que he dado en mi camino académico y profesional ha estado guiado por el deseo de brindarte un futuro lleno de oportunidades y de ser un modelo a seguir para ti.Gracias, querida hermana, por ser mi apoyo incondicional.

¡Gracias de todo corazón!

# Tabla de contenidos

# 1. Introducción

¿Podrían el Big Data y la Inteligencia Artificial ayudar en la prevención del crimen? Esta pregunta surge a raíz de series de televisión de ciencia ficción como “Person of Interest", creada por Jonathan Nolan. A través de su trama de ciencia ficción, la serie examina el potencial de estas tecnologías para predecir y evitar actos delictivos antes de que ocurran. Sin embargo, también se abordan las implicaciones éticas de utilizar estas tecnologías en la seguridad y la protección de los derechos individuales. La serie nos invita a reflexionar sobre el equilibrio entre la seguridad y la privacidad, así como sobre los límites y responsabilidades de la aplicación de la Inteligencia Artificial en la prevención del crimen.

En la actualidad, los avances en el campo del Big Data y la Inteligencia Artificial ofrecen nuevas oportunidades para explorar enfoques predictivos en la seguridad pública, con el objetivo de mejorar la eficacia en la prevención del crimen.

La delincuencia y la violencia son problemas que han existido a lo largo de la historia y que continúan siendo un desafío importante en muchas sociedades, afectando significativamente la seguridad y el bienestar de las personas. A pesar de los esfuerzos de las autoridades de seguridad, la prevención del crimen sigue siendo un problema complejo y multifacético.

En este contexto, el presente trabajo académico tiene como objetivo analizar en profundidad las técnicas de patrullaje predictivo en el ámbito policial, que se refiere al uso de algoritmos de *Machine Learning* y al análisis de datos para predecir y prevenir la ocurrencia de delitos en áreas específicas. Esta estrategia se basa en la recopilación y análisis de datos históricos y en tiempo real, con el fin de identificar patrones, tendencias y áreas de mayor riesgo delictivo. El estudio examinará cómo estas técnicas pueden ser aplicadas de manera efectiva en la prevención del crimen y la delincuencia. Además, se incluye un glosario en el apartado Anexo 1. Glosario, que contiene la definición de los términos técnicos y conceptos clave utilizados en el estudio. Este glosario proporciona una referencia útil para aquellos lectores que deseen aclarar el significado de los términos específicos utilizados a lo largo del informe.

Inicialmente, se intentó obtener información sobre incidentes, crimen y delincuencia en España. Sin embargo, nos encontramos con obstáculos debido a los trámites burocráticos prolongados y las restricciones impuestas por el Ministerio del Interior para acceder a estos datos. También se intentó limitar la búsqueda a las ciudades de Valencia y Barcelona, pero nos enfrentamos a la misma situación, ya que en España estos datos no están disponibles al público.

Ante esta dificultad, se optó por explorar una alternativa y centrar el estudio en el análisis de los datos suministrados por el Departamento de Policía de San Francisco. Estos datos comprenden información sobre los crímenes y delitos registrados en la ciudad, desglosados por distrito.

A través del uso de modelos de *Machine Learning*, se buscó predecir la delincuencia en la ciudad y comparar los resultados obtenidos a partir de diferentes técnicas estadísticas.

Además, estas predicciones podrían servir de base, por ejemplo, para la aplicación de teoría de grafos y conseguir optimizar las rutas de patrullaje de la policía, lo que permitiría maximizar la eficiencia en la asignación de recursos y reducir la delincuencia en las zonas de mayor riesgo de la ciudad.

En resumen, el presente trabajo de investigación busca contribuir al desarrollo de soluciones innovadoras para la prevención del crimen y la delincuencia, aplicando herramientas de *Machine Learning* en el ámbito policial. Se espera que los resultados obtenidos puedan tener un impacto positivo en la seguridad proponiendo medidas paliativas en el campo policial para aumentar la eficiencia de los agentes y minimizar los incidentes en la ciudad que puedan ser útiles y extrapolables a otras ciudades.

# 2. Estructura del trabajo

Con el propósito de obtener una documentación exhaustiva sobre el tema tratado en este proyecto se incluirán en esta memoria, los siguientes apartados:

- **Introducción**: en este apartado se presenta el tema que se va a tratar en la memoria, se introduce el contexto en el que se desarrolla el trabajo y se explica brevemente el objetivo del mismo.
- **Estructura del trabajo**: en este apartado se presenta la estructura de la memoria, se describen los diferentes apartados que la componen y se explica brevemente lo que se va a tratar en cada uno de ellos.
- **Motivación**: se explica por qué se ha decidido llevar a cabo este trabajo, cuáles son las razones que han llevado a su realización y qué se espera conseguir con él.
- **Objetivos**: se describen de forma clara y concisa los objetivos que se pretenden alcanzar con el trabajo, qué se espera conseguir y por qué son importantes.
- **Estado del arte**: se realiza una revisión bibliográfica sobre el tema tratado en la memoria, se presentan los trabajos y estudios previos que se han llevado a cabo sobre el tema y se explican sus principales conclusiones.
- **Obtención e Integración de los datos**: se describe el proceso de obtención de los datos necesarios para el desarrollo del trabajo y su integración en el mismo.
- **Metodología**: se describe detalladamente la metodología utilizada para llevar a cabo el trabajo, desde el proceso de selección de las herramientas y técnicas hasta el análisis de los resultados obtenidos.
- **Desarrollo del proyecto**: se explica la metodología CRISP-DM utilizada para la resolución del problema planteado, sus diferentes fases y cómo se ha aplicado en el trabajo.
- **Resultados y conclusiones**: se presentan los resultados obtenidos tras la aplicación de la metodología y se elaboran las conclusiones correspondientes, en las que se resumen los hallazgos más relevantes y se hace una valoración crítica de los mismos.
- **Trabajos futuros**: se sugieren posibles líneas de trabajo futuro a partir de los resultados obtenidos y las conclusiones extraídas.
- **Glosario**: se incluyen las definiciones de los términos técnicos que se van a utilizar a lo largo de la memoria para que el lector pueda entender mejor el contenido.
- **Referencias bibliográficas**: se incluyen todas las referencias bibliográficas utilizadas a lo largo de la memoria.
- **Anexos**: se incluyen los anexos necesarios para completar la información presentada en la memoria, como por ejemplo, gráficos, tablas, códigos fuente, etc.

# 3. Motivación

La delincuencia es un fenómeno complejo que afecta a las comunidades en diferentes niveles, una de ellas, la ciudad de San Francisco. En ese sentido, resulta fundamental contar con estrategias avanzadas de patrullaje predictivo para hacer frente al y a actos delictivos. El objetivo principal de estas estrategias es reducir la violencia en áreas de alto riesgo, mejorar la asignación de recursos policiales y contribuir a la disminución general de la delincuencia.

El patrullaje predictivo se basa en el aprovechamiento de tecnologías como la inteligencia artificial y los algoritmos de aprendizaje automático. Estas herramientas permiten analizar grandes volúmenes de datos, identificar patrones complejos y generar pronósticos sobre los lugares donde es más probable que ocurran delitos. Además, el uso de algoritmos para optimizar las rutas de patrullaje tiene el potencial de aumentar la eficiencia y la cobertura en áreas críticas.

El presente trabajo académico se enfoca en realizar un análisis detallado de las técnicas de patrullaje predictivo y su aplicabilidad en una población específica. Inicialmente, se hizo un esfuerzo por acceder a los datos relacionados con crímenes, delitos e incidentes en España. Se estableció contacto con el Ministerio del Interior y se presentó una solicitud formal para obtener dicha información. No obstante, debido a la burocracia y los procesos lentos, no fue posible obtener acceso a los datos solicitados en esta instancia.

Adicionalmente, se exploraron otras alternativas, como intentar acceder a los datos proporcionados por la policía local de Valencia y la guardia urbana de Barcelona. Lamentablemente, también se encontraron restricciones que impidieron obtener los datos deseados de estas fuentes. Ante este panorama, se decidió adoptar una alternativa viable, la cual consiste en utilizar los datos proporcionados por el Departamento de Policía de la ciudad. Estos datos ofrecen información detallada sobre los crímenes y delitos registrados en diferentes distritos, lo cual permitirá llevar a cabo el análisis necesario para el presente estudio. Cabe destacar que, para facilitar la comprensión de los términos y conceptos utilizados en este trabajo, se incluirá un glosario al final del documento.

En cuanto a la metodología, se emplearán redes neuronales recurrentes y modelos de tipo *Transformer* para la predicción del crimen. Estas tecnologías avanzadas permiten analizar los patrones delictivos y realizar predicciones precisas sobre posibles incidentes en áreas específicas. Mediante la aplicación de redes neuronales recurrentes y modelos de redes neuronales de tipo *Transformer*, se busca determinar cuál de estas metodologías es más efectiva para mejorar la capacidad de predicción del crimen y proporcionar información valiosa para la seguridad pública.

Al finalizar el estudio, se realizará una comparación de los resultados obtenidos por cada técnica, con el fin de seleccionar los modelos más efectivos para su implementación en San Francisco. Se espera que la aplicación de estas estrategias contribuya a mejorar la asignación de recursos y a reducir la delincuencia en las zonas de mayor riesgo de la ciudad.

# 4. Objetivos

El objetivo general de este proyecto es analizar las técnicas de patrullaje predictivo y examinar cómo estas técnicas pueden ser aplicadas de manera efectiva en la prevención del crimen y la delincuencia. Se buscará contribuir al desarrollo de soluciones innovadoras para la prevención del crimen y la delincuencia, aplicando herramientas de *Big Data* y *AI* en el ámbito policial.

Para alcanzar este objetivo general, se plantearon los siguientes objetivos específicos:

**Objetivo 1.** Analizar la información proporcionada por el Departamento de Policía de San Francisco acerca de los crímenes y delitos registrados en la ciudad por distrito. Este análisis tiene como finalidad detectar patrones y tendencias en la distribución de los incidentes.

**Objetivo 2.** Implementar modelos de *Machine Learning* para predecir la delincuencia en la ciudad. La implementación de estos modelos permitirá utilizar los datos históricos de los crímenes y delitos registrados para predecir las áreas y momentos más propensos a la ocurrencia de delitos. Comparando los resultados obtenidos de la aplicación de diferentes técnicas de *Machine Learning*, se podrá seleccionar el método más adecuado para alcanzar el objetivo general del proyecto. Esto proporcionará una herramienta efectiva para la anticipación y prevención de la delincuencia en la ciudad.

# 5. Estado del arte

Los primeros estudios realizados sobre el tema que nos ocupan examinan las relaciones existentes entre el crimen y los aspectos socioeconómicos. En "*Self-exciting point process modeling of crime*", (J. Amer. Stat. Assoc., 2011), uno de los estudios más destacados, proporciona una visión integral de los factores socioeconómicos que influyen en la criminalidad, lo cual es fundamental para comprender el contexto en el que se desarrollan las predicciones del crimen. [1]

Otro estudio remarcable, "*Crime prediction based on crime types and using spatial and temporal criminal hotspots*", (Almanie T, Mirza R, Lor E, 2015), en el que Almanie, Mirza y otros, incluyen un enfoque novedoso en el que se propone un modelo de proceso de punto autoexcitables que considera la dependencia temporal de los delitos, lo que mejora la precisión de las predicciones al capturar la dinámica inherente de los actos delictivos. En este enfoque, la ocurrencia de un delito puede aumentar la probabilidad de que ocurran más delitos en un período de tiempo cercano. Esto se debe a que los delitos anteriores pueden tener un efecto estimulante en la ocurrencia de delitos posteriores en la misma área o contexto. Al considerar esta dependencia temporal, los modelos de punto autoexcitables buscan capturar y utilizar estos patrones de excitación mutua para mejorar la precisión de las predicciones del crimen. [2]

Por otro lado, con el creciente uso de redes sociales por parte de la población se incluyeron datos de Twitter para mejorar las predicciones. En este contexto se emplearon técnicas de estimación de densidad de núcleos para predecir el crimen y en "*Predicting crime using Twitter and kernel density estimation*", (M.S.Gerber, 2014), Gerber demostró que la información generada en las redes sociales puede ser aprovechada para mejorar la precisión de las predicciones del crimen. [3]

Con los avances en métodos ensamble surgió un enfoque basado en patrones espaciotemporales que empleaba aprendizaje en conjunto para predecir el crimen. A partir de este enfoque, en "*Crime Forecasting Using Spatio-temporal Pattern with Ensemble Learning*", (Yu, CH., Ding.W, Chen, P. , Morabito, M, 2014), los autores del estudio detectaron la importancia de considerar la información geográfica y temporal para lograr una mayor precisión en las predicciones. [4]

Posteriormente, Murad y Pyun en "*Deep Recurrent Neural Networks for Human Activity Recognition*", (Murad, A. Pyun, J., -Y, 2017), desarrollaron un modelo de predicción del crimen basado en los tipos de delitos y los puntos calientes espaciales y temporales en su investigación que destaca la importancia de considerar la información específica del tipo de delito y los patrones espaciotemporales para mejorar la precisión de las predicciones. [5]

En otro enfoque, "*Crime Prediction for Patrol Routes Generation Using Machine Learning*", (Guevara, C., Santos, M., 2021), Guevara y Santos investigan la generación de rutas de patrullaje basadas en la predicción del crimen en su estudio. Utilizando técnicas de aprendizaje automático, como árboles de decisión y máquinas de soporte vectorial, su objetivo es optimizar la asignación de recursos policiales y mejorar la eficiencia de las estrategias de patrullaje. Al predecir las áreas de mayor actividad

delictiva, pueden diseñar rutas de patrullaje más efectivas y proactivas, lo que contribuye a la prevención y disuasión del crimen.[6]

A medida que comenzó el crecimiento del uso de redes neuronales, en los estudios "*Crime Hot Spot Forecasting: A Recurrent Model with Spatial and Temporal Information*", (Y. Zhuang, M. Almeida, M. Morabito, W. Ding, 2017) y "*Crime Prediction Model using Deep Neural Networks*", (Soon Ae Chun, Venkata Avinash Paturu, Shengcheng Yuan, Rohit Pathak, Vijayalakshmi Atluru, Nabil R. Adam, 1019), se presenta una perspectiva que utiliza una arquitectura de red neuronal recurrente para integrar la información espacial y temporal en la predicción de puntos calientes de criminalidad. Al considerar ambos aspectos, el modelo logra capturar las tendencias y patrones complejos asociados con los delitos, lo que permite una predicción más precisa y efectiva de los puntos calientes de criminalidad. [7][8]

En los últimos años se han realizado avances significativos en la aplicación de técnicas de aprendizaje profundo en la predicción del crimen. Estudios recientes, (L. Lochner, 2020), (S. Zhou, X. Wang y Z. Yang, 2020), (N. Esquivel, O. Nicolis, B. Peralta y J. Mate, 2022), (S. Wang, J. Cao y P. Yu, 2020), (X. Zhou, X. Wang, G. Brown, C. Wang y P. Chin, 2021), han explorado el uso de redes neuronales convolucionales, redes neuronales recurrentes y modelos de lenguaje pre-entrenados para mejorar la precisión de las predicciones del crimen. Estas técnicas permiten considerar la información espacio temporal y otros factores relevantes, como patrones de actividad humana y características delictivas. Como comparativa, el uso de las redes neuronales convolucionales permite capturar características visuales y espaciales en conjuntos de datos de vigilancia, mientras que las redes neuronales recurrentes son capaces de modelar la dependencia temporal de los eventos delictivos. En la aplicación de redes neuronales destacan los modelos de lenguaje pre-entrenados, como CrimeBERT en "*Monitoring and early warning of new cyber-telecom crime platform based on BERT migratio learning*", (S. Zhou, X. Wang, Z.Yang, 2020), han demostrado su eficacia al capturar y comprender el lenguaje relacionado con el crimen. Estos avances amplían las posibilidades y la precisión en la predicción del crimen, brindando nuevas herramientas para apoyar la toma de decisiones en la prevención y combate del delito. [9][10][11][12][13]

Entre las técnicas destacadas relacionadas con redes neuronales ha destacado la aplicación de redes neuronales *Transformer* demostrando ser prometedoras en predicciones en entornos espacio temporales complejos. En "*Mixed Spatio-Temporal Neural Networks on Real-time Prediction of Crimes*", X. Zhou, X. Wang, G. Brown, C. Wang and P. Chin, 2021), puso de manifiesto una mejora significativa en la precisión de la predicción del crimen en comparación con enfoques tradicionales. Al aprovechar la capacidad de las redes neuronales transformers para capturar dependencias espaciales y temporales de manera simultánea, se logró una mayor comprensión de los patrones delictivos y, por lo tanto, una mejor capacidad para predecir futuros incidentes. Además este estudio emplea optimización bayesiana para encontrar los mejores hiper parámetros. [13]

# 6. Obtención e Integración de los datos

Con el propósito de alcanzar los objetivos detallados anteriormente en el apartado 5.Objetivos, se requirió en primer lugar un conjunto de datos que suministró información acerca de los incidentes y su correspondiente ubicación, junto con la fecha en la que se registraron.

Para el desarrollo de este proyecto se utilizaron datos de la ciudad de San Francisco debido a la disponibilidad de información tan detallada.

La fuente de datos seleccionada para este estudio fue *DataSF Open Data*, una plataforma de datos abiertos de la ciudad de San Francisco en California. Esta plataforma ofrece una amplia gama de conjuntos de datos y recursos para que los usuarios puedan acceder, descargar y visualizar los datos de manera efectiva. La elección de esta fuente de datos permitió al estudio contar con información relevante y actualizada para la aplicación de las técnicas de patrullaje predictivo en el ámbito policial.

Para la descarga de los datos necesarios para este proyecto, se empleó el lenguaje de programación Python haciendo uso de la librería *Socrata*. La librería *Socrata* está diseñada para extraer datos alojados por organizaciones gubernamentales en la nube. Para utilizar esta librería, se requirió la creación de una cuenta developer en la página web *OpenDataSF*.

Además, se hizo uso de la librería *Requests* para descargar conjuntos de datos que no requerían de una clave de API, permitiendo la realización de solicitudes para obtener dicha información. El código desarrollado para la obtención de estos datos se encuentra publicado en el archivo *Extract-data* del proyecto creado para el desarrollo de este trabajo en GitHub.

Los conjuntos de datos que fueron seleccionados para el desarrollo de este estudio fueron los siguientes:

1. ***Police Department Incident Reports: 2018 to Present*** . Este dataset contiene registros de los incidentes (crímenes y delitos) registrados en San Francisco desde 2018 hasta la actualidad. Cada registro incluye información tal como el distrito o vecindario en el que ocurrió el incidente, la fecha y hora, el tipo de incidente, la descripción del delito, la resolución y el número de identificación del caso. Además, se proporciona información sobre la ubicación geográfica del incidente en términos de latitud y longitud. La descripción detallada de las variables presentes se encuentra en la sección *Dataset Explainers* de la plataforma.

   Este fue el conjunto de datos principales para el desarrollo del proyecto.

   A continuación, se incluye una breve descripción de las columnas presentes en este conjunto de datos :

| Columna | Descripción |
|---|---|
| *incent_datetime* | Fecha del incidente en formato year-mm-dd hh:mm:ss |
| *incident_date* | Fecha del incidente en formato year-mm-dd |
| *incident_time* | Hora del incidente en formato hh:mm:ss |
| *incident_year* | Año del incidente |
| *incident_day_of_week* | Día de la semana |
| *report_datetime* | Fecha y hora en la que el incidente fue reportado |
| *row_id* | Identificador de la fila en la que se encuentra el registro |
| *incident_id* | Identificador del incidente |
| *incident_number* | Número del incidente |
| *report_type_code* | Código del tipo de informe en el que se ha almacenado la información del incidente |
| *report_type_description* | Descripción del tipo de informe en el que se ha almacenado la información del incidente |
| *incident_code* | Código del incidente |
| *incident_category* | Categoría del incidente |
| *incident_subcategory* | Subcategoría del incidente |
| *incident_subdescription* | Descripción del incidente |
| *resolution* | Resolución del incidente |
| *police_district* | Distrito policial en el que se produjo el incidente |
| *field_online* | Indica si el incidente ha sido reportado de forma online |

| | |
|---|---|
| *cad_number* | Número de sistema que identifica a la llamada de emergencia o al incidente |
| *intersection* | Intersección de calles más cercana al incidente |
| *cnn* | Número de coordenadas |
| *analysis_nighborhood* | Vecindario en el que se produjo el incidente |
| *supervisor_district* | Distrito supervisor al que pertenece la localización del incidente |
| *supervisor_district_2012* | Distrito supervisor al que pertenece la localización del incidente según la delimitación establecida en el año 2012 |
| *latitude* | Latitud de la ubicación en la que se produjo el incidente |
| *longitude* | Longitud de la ubicación en la que se produjo el incidente |
| *point* | Tupla que representa el par (latitud,longitud) que permite identificar la ubicación del incidente |

**Figura 1. Tabla resumen datos *Police Department Incident Reports: 2018 to Present***

2. ***Bay Area Countries*.** Este dataset publicado por el Departamento de Planificación de San Francisco en la plataforma *OpenData* proporciona información sobre las coordenadas geográficas de las ciudades que pertenecen al Área de la Bahía de San Francisco. Este conjunto permitió identificar los puntos geográficos que pertenecen a San Francisco y además fue de utilidad en las visualizaciones gráficas.

| **Columna** | **Descripción** |
|---|---|
| *the_geom* | Geometría espacial de las áreas |
| *OBJECT_ID* | Identificador único para cada área |
| *FIPSSTCO* | Hora del incidente en formato hh:mm:ss |
| *COUNTY* | Nombre del condado |

**Figura 2. Tabla resumen *Bay Area Countries***

3. ***Current Police Districts*.** En este dataset se encuentra información sobre los polígonos que definen los distritos policiales de San Francisco publicados en la plataforma *OpenData*.

| Columna | Descripción |
|---|---|
| *COMPANY* | Nombre de la compañía policial responsable del distrito |
| *the_geom* | Geometría espacial que representa la forma y límites de cada distrito |
| *Shape_Leng* | Longitud del contorno del distrito en unidades de longitud |
| *DISTRICT* | Nombre del distrito |
| *Shape_Le_1* | Longitud adicional del contorno del distrito |
| *Shape_Area* | Área del distrito en unidades de área |

**Figura 3. Tabla resumen *Current Police Districts***

4. ***Reference Incident Code Crosswalk*.** En este conjunto de datos tenemos información relacionada con los códigos de incidente y su correspondencia con las categorías y subcategorías de incidentes del primer dataset descargado.

   Estos datos han sido publicados por el departamento de Policía de San Francisco en la misma plataforma que los datos anteriores. Esta información nos permitió detectar correspondencias atípicas entre las tres variable incident_code, incident_category e incident_subcategory presentes en el conjunto de datos principal.

| Columna | Descripción |
|---|---|
| *INC_CODE* | Código del incidente |
| *CATEGORY* | Categoría del incidente |
| *SUBCATEGORY* | Subcategoría del incidente |

**Figura 4. Tabla resumen *Reference Incident Code Crosswalk***

5. ***Analysis neighborhoods.*** En este archivo tenemos los polígonos que determinan los distintos vecindarios del condado de San Francisco. También se ha descargado de la plataforma *OpenData*.

Este archivo nos permitió calcular el vecindario a partir de nuestros registros iniciales.

| Columna | Descripción |
|---|---|
| *nhood* | Nombre del vecindario |
| *the_geom* | Geometría espacial que representa la forma y límites de cada distrito |

**Figura 5. Tabla resumen datos *Analysis neighborhoods***

6. ***Street Names.*** En este conjunto de datos tenemos las distintas calles que hay en San Francisco de la plataforma *OpenData* para poder estudiar los valores de calle que tengamos en el dataset principal.

| Columna | Descripción |
|---|---|
| *FullStreetName* | Nombre completo de la calle |
| *StreetName* | Nombre de la calle |
| *StreetType* | Abreviatura del tipo de calle |
| *PostDirection* | Dirección postal |

**Figura 6. Tabla resumen *Street Names***

En cuanto a la integración de nuestros datos seguimos los siguientes pasos:

- **Limpieza y análisis exploratorio. Parte 1:**

Se realizó un análisis inicial y exploración de los datos recopilados. Durante este proceso, se identificaron las columnas que eran relevantes para el estudio y se descartaron aquellas que no contenían información relevante. Además, se examinaron los datos en busca de valores anómalos, atípicos y ausentes, los cuales fueron tratados adecuadamente para garantizar la calidad de los datos utilizados en nuestro análisis.

- **Preprocesado de los datos. Parte 1:**

En esta etapa, se llevó a cabo el cálculo de nuevas variables que se consideraron necesarias para el estudio. Estas variables incluían información como la calle donde ocurrieron los incidentes, el vecindario correspondiente, el distrito policial al que pertenecen y la identificación de si el día del incidente fue festivo o no. El preprocesado de los datos permitió tener un conjunto de variables más completo y preparado para su posterior análisis.

- **Limpieza y análisis exploratorio. Parte 2:**

En esta fase adicional de limpieza y análisis exploratorio, se examinaron con mayor detalle los resultados obtenidos de las variables calculadas en la etapa anterior. Se realizó un estudio de las relaciones existentes entre nuestras variables de interés y se llevaron a cabo técnicas de clustering para intentar identificar patrones o grupos de incidentes similares. Esta exploración nos proporcionó una visión más profunda de los datos y nos ayudó a descubrir posibles correlaciones o tendencias relevantes.

- **Preprocesado de los datos. Parte 2 (codificación de variables categóricas):**

Esta última etapa de preprocesado, se centró en la codificación de variables categóricas. Estas variables, como el tipo de incidente o la descripción del mismo, fueron codificadas de manera adecuada para poder utilizarlas en análisis posteriores, como modelos de aprendizaje automático o técnicas estadísticas. La codificación de variables categóricas nos permitió trabajar de manera más eficiente con estos datos en la aplicación de modelos estadísticos. También se aplicó un clustering con el objetivo de analizar los patrones y tendencias existentes en el conjunto de datos, así como para detectar clusters y crear los modelos estadísticos por cluster para mejorar la precisión del modelo.

Estos cuatro pasos de integración de datos fueron fundamentales para garantizar la calidad, la consistencia y la preparación adecuada de los datos utilizados en nuestro estudio. Estos pasos se encuentran detallados con exhaustividad en el apartado 9. Aplicación Crisp-DM, Fase 3. Preparación de los datos.

A través de la limpieza, el análisis exploratorio y el preprocesado de los datos, obtuvimos un conjunto de datos listo para su análisis en profundidad y la realización de nuestros objetivos de investigación.

Finalmente el dataset que se obtuvo sin incluir las variables codificadas contenía las variables que se detallan a continuación. Estas variables proporcionan información detallada sobre los incidentes reportados en San Francisco, incluyendo la fecha, categoría, ubicación, resolución y otras características relevantes. A continuación se detallan las variables finales sobre las cuales se realizó el estudio y una breve descripción.

| **Columna** | **Descripción** |
|---|---|
| *incident_datetime* | Fecha del incidente en formato year-mm-dd hh:mm:ss |
| *incidente_day_of_week* | Día de la semana en el que ocurrió el incidente |
| *incident_code* | Código del incidente |
| *incident_category* | Categoría del incidente |
| *incident_subcategory* | Subcategoría del incidente |

| | |
|---|---|
| *resolution* | Resolución del incidente |
| *latitude* | Latitud geográfica del incidente |
| *longitude* | Longitud geográfica del incidente |
| *neighborhood* | Vecindario en el que tuvo lugar el incidente |
| *police_district* | Distrito Policial en el que ocurrió el incidente |
| *holiday* | Variable binaria que indica si el día era festivo o no |
| *street* | Calle en la que ocurrió el incidente. |
| *day* | Número del día del mes del incidente |
| *month* | Mes del incidente |
| *year* | Año del incidente |
| *hour* | Hora del incidente |
| *minutes* | Minuto en el que ocurrió el incidente |
| *week* | Semana del mes en la que ocurrió el incidente |
| *quarter* | Cuatrimestre en el que ocurrió el incidente |
| *season* | Estación del año en la que ocurrió el incidente |
| *interval_hour* | Intervalo horario en horas en el que tuvo lugar el incidente |
| *interval_minutes* | Intervalo horario en minutos en el que ocurrió el incidente |

**Figura 7. Tabla resumen datos finales empleados en el proyecto**

En el siguiente apartado, se define la metodología seguida para emplear los datos anteriores como entrada a un modelo estadístico que permitió predecir incidentes en tiempo real.

# 7. Metodología

En este proyecto de ciencia de datos, se adoptó la metodología *CRISP-DM*, que es ampliamente utilizada en proyectos de ciencia de datos para abordar problemas de minería de datos y análisis de datos. *CRISP-DM* proporciona una estructura sistemática e iterativa para guiar el desarrollo de proyectos de ciencia de datos, desde la comprensión del problema hasta la implementación de soluciones. A continuación, proporcionaremos un resumen conciso de cómo se aplicará cada fase del modelo *CRISP-DM* en el presente proyecto. Posteriormente, explicaremos la implementación de cada una de las etapas del modelo *CRISP-DM* en el proyecto, ajustándose a la problemática particular abordada en este estudio y a los datos que estamos analizando.

**Metodología *CRISP-DM***

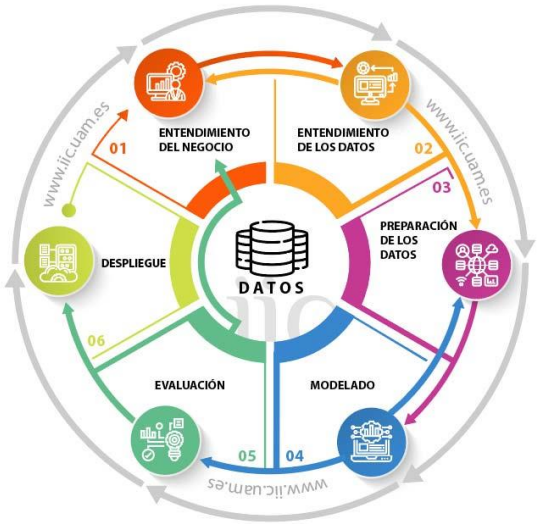


**Figura 8.** [Instituto de ingeniería del conocimiento (IIC), la metodología *CRISP-DM* en ciencia de datos.]

1. **Entendimiento del negocio.** Esta etapa, se enfocó en comprender a fondo el problema de predicción de incidentes en San Francisco en función del tiempo y el lugar. Se analizó el contexto del problema, se identificaron los objetivos y requisitos del proyecto, y se definieron las métricas de evaluación utilizadas para medir el rendimiento de los modelos de *Machine Learning*. También se examinaron las restricciones y consideraciones éticas relevantes para garantizar la validez de los resultados.
2. **Entendimiento de los datos.** Esta fase se centró en el análisis y la exploración de los datos relacionados con los incidentes en San Francisco. Se realizó un estudio detallado de las características y la calidad de los datos, identificando posibles problemas, como valores faltantes, inconsistencias o desequilibrios en la distribución de clases. También se investigaron las relaciones entre las variables y se buscaron patrones significativos que pudieran ser relevantes para el objetivo planteado de predicción.

3. **Preparación de los datos.** Una vez se comprenden completamente los datos, se procedió a realizar tareas de limpieza, transformación y selección de características. Se trataron los valores faltantes, se corrigieron inconsistencias, se realizó la codificación adecuada de variables categóricas y se escalaron las características según fuera necesario. Además, se crearon conjuntos de entrenamiento, validación y prueba para los modelos de *Machine Learning*, asegurando una partición adecuada de los datos.
4. **Modelado.** En esta etapa, a través de la aplicación de redes neuronales recurrentes y modelos de tipo *Transformer*, se buscó cumplir con el objetivo general de analizar las distintas técnicas de patrullaje predictivo en la prevención del crimen y la delincuencia. Estas tecnologías avanzadas permitieron analizar patrones delictivos y realizar predicciones precisas sobre posibles incidentes en áreas específicas. Con este enfoque, se buscó evaluar la efectividad de las redes neuronales recurrentes y los modelos *Transformer* en mejorar la capacidad de predicción del crimen, proporcionando información valiosa para fortalecer la seguridad pública.
5. **Evaluación.** En esta fase, se evaluó el rendimiento de los modelos desarrollados utilizando las métricas definidas previamente en la etapa de entendimiento del negocio. También se analizaron las predicciones realizadas en contraste con los valores reales y se estudiaron los errores cometidos por los modelos, para así comprender mejor las fortalezas y debilidades de los modelos.
6. **Despliegue.** Finalmente, en esta etapa se implementaron los modelos de *Machine Learning* en un entorno de producción. Además, se documentaron y comunicaron los resultados obtenidos, proporcionando información clara y comprensible para los *stakeholders* del proyecto.

Finalmente, esta metodología nos proporcionó un enfoque estructurado y riguroso para abordar este proyecto permitiendo comprender el problema. Esta metodología fue de utilidad para obtener resultados confiables y alcanzar los objetivos planteados.

# 8. Desarrollo del proyecto

Una vez definida la metodología a seguir este proyecto, se aplicaron cada una de las fases del *CRISP-DM*. A continuación se encuentran detalladas cada una de estas fases describiendo las acciones realizadas, las técnicas empleadas y los resultados obtenidos.

## Fase 1. Entendimiento del negocio.

En esta primera fase, se centró toda la atención en la búsqueda de estudios relacionados que tuvieron como objetivo predecir el crimen en función del tiempo y del lugar, ya que nuestro objetivo era predecir el lugar de los incidentes y el momento en el que estos vayan a ocurrir. A partir de estos estudios decidimos predecir nuestras variables espacio y tiempo a partir de la aplicación de redes neuronales de tipo *Transformer* ya que se ha demostrado que estos modelos de *Machine Learning* mejoraron la precisión en la predicción del crimen.

Las redes neuronales *Transformers* surgieron para superar las limitaciones de las redes neuronales recurrentes en cuanto a cálculo secuencial y memoria.

Las redes neuronales recurrentes (RNN) por su parte son redes neuronales feed forward que procesan la información de forma estática y no tienen memoria interna. La información fluye de la capa de entrada hacia las capas ocultas, y finalmente hacia la salida. Las RNN son una variante de las *feedforward* que se adaptan para procesar secuencias hacia adelante y capturar dependencias temporales. Adecuadas para tareas secuenciales. Kindle Direct Publishing. (2019). "Deep Learning – Introducción práctica con Keras (SEGUNDA PARTE)". ISBN 978-1-687-47399-8.

Estos algoritmos se emplean para problemas ordinarios o temporales como el procesamiento del lenguaje natural (NLP) o el reconocimiento de voz.

Utilizan datos de entrenamiento para aprender y destacan por hacer uso de memoria, esto quiere decir que obtienen información de entradas anteriores para influir en la entrada y salida actuales. Estas redes son un conjunto de neuronas que reciben una entrada y que generan una salida.

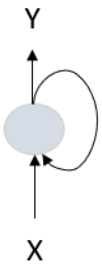


**Figura 9. Modelo simplista de una red neuronal**

El funcionamiento de estos algoritmos se basa en que la neurona recibe una entrada y genera una salida que se alimenta a sí misma.

En cada instante de tiempo, *timestep,* esta neurona recibe la entrada x de la capa anterior y su propia salida del instante de tiempo anterior para generar su salida tal y como se visualiza en la Figura 10.

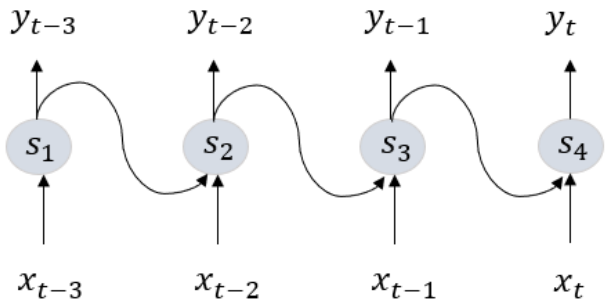


**Figura 10. Red neuronal recurrente**

La salida se obtiene aplicando una función de activación a la suma ponderada de la entrada. La función de activación de una red neuronal no es más que una función matemática que depende del tipo de red neuronal y el problema que se está abordando.

Las funciones de activación más comunes son:

- **Función Sigmoide,** genera salidas con valores 0 y 1. Se emplea para problemas de clasificación binaria.

$$f^{(x)} = \frac{1}{1 - e^{-x}} \qquad (1)$$

- **Función ReLU,** mapea únicamente los valores positivos truncando los valores negativos a cero. Destaca su uso popular por su simplicidad y buen rendimiento.

$$f(x) = \max(0, x) = \begin{cases} 0 & \forall < 0 \\ x & \forall x \geq 0 \end{cases} \qquad (2)$$

- **Función Tanh,** genera salidas en el intervalo -1 a 1. Se utiliza en problemas donde la salida debe estar centrada alrededor del 0.

$$f^{(x)} = \frac{2}{1 - e^{-2x}} - 1 \qquad (3)$$

- **Función Softmax,** es aplicada a la capa de salida calculando la probabilidad de cada una de las clases. Resalta su empleo en problemas multiclase en los que se debe asegurar que la suma de probabilidades de las diferentes clases debe ser 1.

$$f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad (4)$$

Además de tener en cuenta el *timestep* y emplear funciones de activación, las redes neuronales utilizan una matriz de pesos, W, para establecer conexiones ponderadas entre las neuronas de una capa y las neuronas de la capa siguiente. Esta matriz de pesos es fundamental en el funcionamiento de la red neuronal, ya que determina la influencia y la fuerza de las conexiones entre las neuronas.

Teniendo en cuenta el enfoque matemático, cada neurona recurrente tiene dos conjuntos de parámetros, el primero que hace referencia a la entrada de datos que recibe la capa anterior y otro conjunto que lo aplica a la entrada de datos respecto al vector salida del instante, t, anterior. Expresándose de forma matemática obtendremos la ecuación de la (5).

$$S_t = g_1(Wx_t + W_s S_{t-1}) + b \quad (5)$$

$$Y_t = g_2(W_Y + W_s S_t) + b \quad (6)$$

Siendo :

- S, estados de memoria ocultos asociados a un instante de tiempo
- x, entradas en los instantes de tiempo
- t, instante de tiempo actual
- W, matriz de pesos
- $g_1$ y $g_2$, funciones de activación
- b, bias o sesgo

Una vez que se ha introducido la matriz de pesos, denotada como W, la representación visual de una red neuronal se puede observar en la Figura 11.

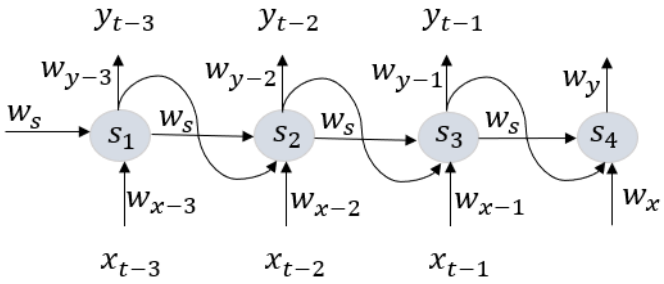


**Figura 11. Representación visual de la red neuronal introduciendo W**

Tras haber generado las salidas, este modelo hace uso del algoritmo *backpropagation* o *backward propagation*. Este algoritmo se emplea para ir reajustando el modelo después de haber obtenido el resultado, verifica si este es correcto, o no, para obtener la función de pérdida y posteriormente mediante el algoritmo *backward propagation* que calcula las derivadas parciales del error con respecto a los pesos de las neuronas definidos en el parámetro W de las ecuaciones (5) y (6).

Este algoritmo se basa en la función de pérdida, que cuantifica la discrepancia entre el valor predicho, yhat, y el valor real y. El objetivo durante el entrenamiento del modelo es minimizar esta función, lo que se logra ajustando parámetros de la red neuronal como los pesos y los sesgos.

La elección de esta función de pérdida dependerá del problema a resolver mediante el modelo predictivo.

Teniendo en cuenta el valor real, el valor predicho, el tamaño de la muestra y la función de pérdida genérica, independientemente de si el problema es de regresión o clasificación, el valor de la función se calculará aplicando la fórmula de la ecuación (6).

$$E(\theta) = \frac{1}{m}\sum_{i=1}^{m} L(y, yhat) \qquad (7)$$

**Siendo :**

**E, función de pérdida total**
**$\theta$, parámetros del modelo, pesos y sesgos**
**m, tamaño de la muestra**
**L, función de pérdida individual**
**y, valor real**
**yhat, valor predicho**

Una vez definida la pérdida, veamos cómo se aplica el algoritmo *backpropagation*. Su aplicación se basa en ir actualizando los valores realizando las derivadas parciales teniendo en cuenta el sesgo o bias, b, y los pesos, W. Las fórmulas de actualización de los parámetros se pueden observar en las ecuaciones (8) y (9).

$$w_{t+1} := w_t - \in \frac{\partial E(\theta)}{\partial w} \qquad (8)$$

$$b_{t+1} := b_t - \in \frac{\partial E(\theta)}{\partial b} \qquad (9)$$

**Siendo :**

**$\in$, hiperparámetro arbitrario, típicamente 0,5**

A partir de las ecuaciones (8) y (9), los valores de actualización de las predicciones, yhat, se obtienen por medio del uso de las siguientes ecuaciones.

$$\Delta w_t = -\in \frac{\partial E(\theta)}{\partial w} \qquad (10)$$

$$\Delta b_t = -\in \frac{\partial E(\theta)}{\partial b} \qquad (11)$$

**Siendo :**

**$\in$, hiperparámetro arbitrario, típicamente 0,5**

Donde E se calcula como se indica en la ecuación (12) :

$$E = L(y, yhat),\ yhat = g(z) \text{ y } z = wx + b \qquad \textbf{(12)}$$

Centrándonos ahora en las limitaciones de las redes neuronales recurrentes, la problemática de estas redes neuronales es el cálculo secuencial que emplea y la poca memoria que presentan. Esta poca memoria se debe a que para calcular la salida actual se emplea como entrada la salida anterior, pero no incluye las salidas del resto de neuronas. De esta forma, al calcular la última neurona no se tiene en cuenta las neuronas iniciales, por lo que se pierde precisión al no emplear la información de todas las salidas anteriores.

Para resolver esta problemática surgieron las redes neuronales *Transformer*, destacando por el empleo de la atención en una arquitectura *encoder-decoder* para capturar relaciones en la secuencia de entrada de manera más eficiente y procesarla de forma paralela.

Estas redes han demostrado mejorar la precisión de los modelos en muchas tareas de procesamiento del lenguaje natural en comparación con las arquitecturas tradicionales de las redes neuronales recurrentes. Sin embargo, la mejora en la precisión también depende de factores como la calidad y la cantidad de los datos de entrenamiento, la elección adecuada de hiperparámetros y la configuración del modelo.

Las redes neuronales Transformers ofrecen una arquitectura flexible y poderosa para el procesamiento de datos secuenciales y geoespaciales, lo que las convierte en una opción prometedora para la predicción de la fecha, la latitud y la longitud en diversos escenarios.

Este enfoque matemático, parte de la arquitectura de redes neuronales, ya definida en apartados anteriores, añadiendo el concepto de atención.

***Transformer* model architecture**

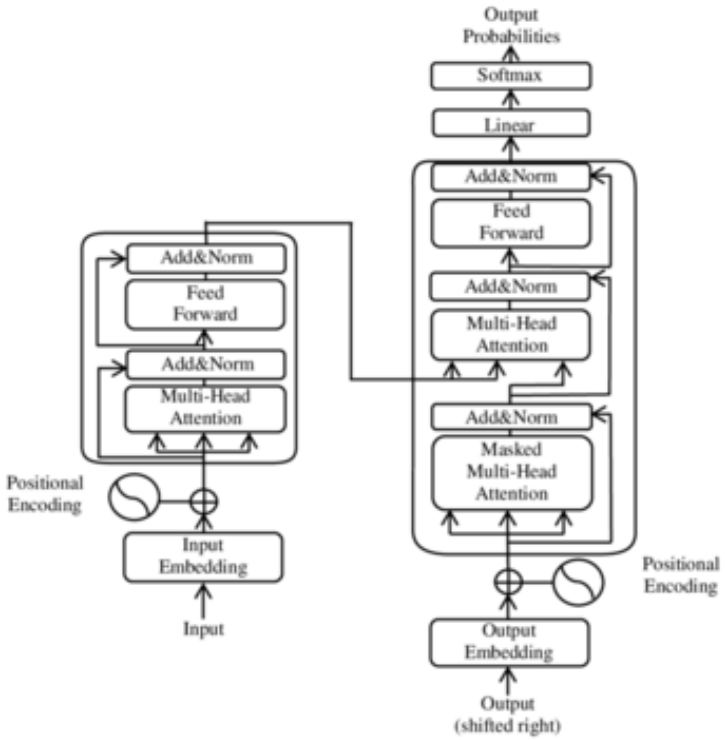


**Figura 12. Arquitectura fundamental de una Red Neuronal *Transformer* (fuente: Wikipedia)**

En la Figura 12, se muestra la arquitectura fundamental que sustenta a las redes neuronales *Transformer*. Aunque la representación visual puede parecer abrumadora, los conceptos clave que subyacen a esta innovadora arquitectura son los siguientes.

- **Codificaciones posicionales o *Positional Encoding*,** consiste en la adición de un mecanismo que le asigna a cada *input* un número que indica la posición u orden dentro del resto de *inputs*. Esto le proporciona a la red información sobre la ubicación relativa de cada elemento de la secuencia.
- **Atención o *Attention*,** sistema que emplean estas redes neuronales para representar la importancia que tienen otros *inputs* en la entrada para una posterior codificación del *input* actual.
- **Autoatención,** variante del sistema anterior, *Attention*, en el cual se calculan los pesos de atención considerando únicamente los inputs dentro de la misma secuencia o contexto. Esto permite que la red detecte las relaciones que existen y la importancia de elementos dentro de una misma secuencia.

A partir de estos conceptos vamos a entender el funcionamiento de esta arquitectura de acuerdo a la imagen de la Figura 12.

En primer lugar, tenemos el *input* **(x)** que es una secuencia de elementos, como palabras o *tokens* que se emplearán como entrada para crear el modelo. A este *input* se le aplica una función **g(x)** dónde se le asigna un vector de características a cada elemento de la secuencia, el objetivo de esta función, **g(x)**, es el de convertir los tipos de datos que tenemos en valores que puedan ser procesados por la red neuronal. La aplicación de esta función se denomina *Input embedding*.

A continuación, se aplica lo que es denominado como *Positional Encoding* $\mathbf{h}_i$, cuyo propósito es añadir información sobre la posición de cada elemento de la secuencia de entrada a los valores obtenidos en el paso anterior.

Sobre el valor de $\mathbf{h}_i$, calculado en la etapa anterior se aplica el mecanismo de atención, ***Attention* (q, K, V)**. De esta forma obtenemos para cada elemento de la secuencia de entrada un vector con los valores obtenidos de la aplicación del input embedding, el valor del *Positional Encoding* y el valor de atención obtenido. Dependiendo del problema la función *Attention* varía, las funciones más comunes son *Self-Attention*, *Masked-Attention* y *Encoder-Decoder Attention*.

Posteriormente se aplica el proceso ***Add&Norm*** o ***Layer Normalization***, técnica que se emplea para combinar y normalizar la información después de la aplicación de nuestra función Attention.

Tras haber normalizado los valores de nuestro vector *Attention* se le aplica la técnica ***Feed Forward*** que es una capa de la red que procesa y realiza una transformación lineal de la información para posteriormente aplicar una función de activación. En esta etapa se capturan relaciones y patrones dentro de los datos.

Seguidamente, los vectores resultantes de la etapa de *Feed Forward* pasan por la etapa de ***Output Embedding* (g(y))**, donde se asigna un vector de características a cada elemento de salida. Esta etapa es similar al *Input embedding*, pero se aplica a los elementos de salida en lugar de los elementos de entrada. Cada elemento de salida obtiene un vector de características específicas que captura su información relevante.

Después del *Output embedding*, se puede aplicar una atención enmascarada (Masked attention) en la cual se evita que ciertos elementos se atiendan entre sí. Esto se utiliza, por ejemplo, durante el entrenamiento del modelo para predecir elementos basándose en información anterior y no futura.

A continuación, se realiza un proceso de normalización y suma, ***Add&Norm*** o ***Layer Normalization*****,** para combinar y normalizar la información después de la etapa de atención.

Finalmente, los vectores normalizados se pasan por una capa lineal, ***Linear***, seguida de una función de activación *softmax* para obtener las probabilidades de salida. Estas probabilidades representan la distribución de probabilidad sobre los posibles valores de salida, y el modelo selecciona el valor con la mayor probabilidad como la predicción final.

Una vez realizada esta introducción vamos a estudiar el enfoque aplicado a nuestro problema inicial, el cálculo de la fecha y la latitud y la longitud de los futuros incidentes en San Francisco.

Nuestro objetivo es predecir la latitud, longitud y la fecha del incidente en formato dd/mm/yyyy hh:mm, variables de salida, a partir de unas variables de entrada que definiremos más adelante tras la limpieza.

Con la intención de crear modelos estadísticos con una precisión alta, previamente se realizará un *clustering* para la detección de grupos dentro de nuestros datos y para cada grupo se creará un modelo estadístico.

Para crear estos modelos estadísticos tendremos que seguir los siguientes pasos:

- **Análisis y preparación de los datos**
  En esta etapa, es fundamental llevar a cabo un análisis exploratorio exhaustivo de los datos con el objetivo de identificar posibles valores atípicos y anomalías que puedan afectar el entrenamiento de nuestro modelo. Asimismo, se abordará la detección de datos faltantes y se evaluará la necesidad de aplicar técnicas de imputación para completarlos de manera adecuada.

  Además, una vez finalizado el análisis exploratorio, se procederá a evaluar la conveniencia de aplicar técnicas de escalado, normalización o estandarización de variables con el fin de optimizar el desempeño del modelo.

- **Construcción del modelo**
  En esta fase, se procederá a la creación de la arquitectura de la red neuronal *Transformer*, la cual se compone de capas de codificación y decodificación.

  Para la codificación y decodificación, se definirán las capas de atención, capas de *feed-forward* y capas de normalización. Estas capas serán fundamentales para capturar las relaciones y dependencias en los datos de entrada, así como para generar una salida precisa.

  Asimismo, se incorporarán capas de salida especializadas que permitirán predecir la latitud, longitud y fecha/hora del incidente en cuestión. Estas capas serán diseñadas de manera adecuada para garantizar una predicción precisa y confiable de estos atributos clave.

- **Preprocesado de las variables de entrada.**
  En esta etapa, se llevará a cabo el preprocesamiento de las variables de entrada con el fin de preparar los datos para su posterior entrenamiento y evaluación en el modelo. Esto implica la realización de tareas como la limpieza de datos, la normalización, el escalado, la codificación de variables categóricas, la selección de características relevantes, entre otras técnicas apropiadas según la naturaleza de los datos.

- **Entrenamiento del modelo**
  Después de completar el preprocesamiento de los datos, se dará inicio al proceso de entrenamiento del modelo utilizando los datos preparados. En esta etapa crucial, se llevará a cabo un ajuste de los parámetros de la arquitectura de la red neuronal a través de un proceso iterativo de optimización. El objetivo principal será minimizar la función de pérdida y mejorar el rendimiento general del modelo. Para minimizar estos valores también se realizará un *clustering* para detectar clusters dentro de nuestros datos y crear un modelo estadístico por cada cluster.

  En nuestro caso particular, los parámetros a ajustar serán cuidadosamente seleccionados para optimizar el desempeño de la red neuronal. Estos parámetros incluirán el número de capas de la red, el número de cabezas de atención, el tamaño de lote utilizado durante el entrenamiento y la tasa de regularización aplicada. Al ajustar estos parámetros de manera adecuada, buscaremos lograr un

equilibrio óptimo entre la capacidad de aprendizaje del modelo y su capacidad para generalizar correctamente a nuevos datos.

- **Validación del modelo**
  Para la validación de cada uno de los modelos desarrollados, se emplearán métricas específicas para evaluar el desempeño y la precisión de las predicciones realizadas. Para las variables numéricas, como latitud y longitud, se utilizará el Error Cuadrático Medio (MSE), el cual proporciona una medida cuantitativa de la discrepancia entre los valores reales y los valores predichos. Por otro lado, para las variables categóricas, como fecha y hora, se empleará la Pérdida Logarítmica de Verosimilitud (LOG-LOSS), una métrica adecuada para evaluar la precisión de las predicciones en problemas de clasificación.

  Con el objetivo de visualizar y analizar estas medidas, se utilizarán curvas de aprendizaje que permitirán examinar el comportamiento del MSE a medida que se incrementa el tamaño de los conjuntos de datos utilizados en el entrenamiento. De igual manera, se emplearán curvas de pérdida para observar la evolución de la pérdida logarítmica de verosimilitud en relación con la cantidad de iteraciones realizadas durante el entrenamiento.

  Además, para una comprensión más intuitiva y una comparativa visual entre los valores reales y los valores predichos, se mostrarán gráficos de dispersión.

  Estos gráficos permitirán visualizar la distribución y la relación entre los valores predichos y los valores reales, lo que facilitará la identificación de posibles discrepancias o tendencias en las predicciones realizadas por los modelos.

- **Prueba y predicción**
  Una vez completado el entrenamiento y validación del modelo, se procederá a realizar pruebas y predicciones sobre conjuntos de datos independientes. Esta etapa tiene como objetivo evaluar la capacidad del modelo para realizar predicciones precisas en situaciones del mundo real y verificar su rendimiento en condiciones no vistas durante el entrenamiento y validación.

## Fase 2. Entendimiento de los datos.

Para asegurar la creación de modelos estadísticos con una alta precisión, previamente se realizó un análisis exploratorio para comprender la información proporcionada por los datos disponibles.

Partiendo del conjunto de datos, **Police Department Incident Reports: 2018 to Present**, el desarrollo de esta fase se dividió en las siguientes etapas.

- **Etapa 1. Exploración de las dimensiones y selección de variables.**

En esta fase inicial del trabajo, se llevó a cabo una revisión exhaustiva del conjunto de datos con el objetivo de comprender en detalle su estructura general, así como la cantidad de registros y variables disponibles. Esta revisión fue fundamental para sentar las bases de nuestro modelo.

Además, durante esta etapa también se establecieron los pasos a seguir en el preprocesamiento de los datos, con el fin de obtener nuevas variables que contribuyan a mejorar la precisión de nuestros modelos.

El conjunto de datos descargado para este estudio cuenta con 717.040 registros y 27 variables. Estas variables se encuentran descritas en el apartado de Obtención e Integración de los datos.

De estas 27 variables decidimos descartar las siguientes:

- En vista de la redundancia de información presente en las variables relacionadas con la fecha, como **'incident_datetime', 'incident_date', 'incident_time', 'incident_year'** y **'incident_day_of_week'**, hemos realizado una selección cuidadosa. Con el objetivo de simplificar y mejorar nuestro modelo, hemos decidido conservar únicamente dos variables: **'incident_datetime' e 'incident_day_of_week**'.

  Estas variables son de particular relevancia para nuestro estudio, ya que nos permiten capturar el horario real del incidente y el día de la semana en los valores predichos. Al centrarnos en estas dos variables clave, evitamos la duplicación de información y nos enfocamos en aspectos fundamentales para nuestro análisis.

- La variable **'report_datetime'** será excluida de nuestro análisis debido a su naturaleza informativa sobre la fecha y hora en que se notificó a las autoridades sobre el incidente. En el contexto de nuestro estudio, nos interesa principalmente predecir el momento y lugar exactos del incidente, en lugar de centrarnos en el momento en que se reporta dicho incidente.

  Nuestro objetivo es anticiparnos a los acontecimientos y predecir de manera precisa los incidentes en sí, en lugar de simplemente registrar cuándo se notifican. Por lo tanto, al eliminar la variable '*report_datetime*', nos aseguramos de que nuestro modelo se centre en los aspectos fundamentales del incidente y pueda proporcionar predicciones más precisas y útiles.

- Debido a la naturaleza de las variables ***'police_district'*** y ***'analysis_neighborhood'***, las cuales contienen información geográfica, hemos decidido no incluirlas en nuestro análisis inicial. Esto se debe a que no

tenemos la certeza de que los distritos policiales y los vecindarios hayan mantenido una estructura constante a lo largo de todo el período abarcado por el conjunto de datos.

Para abordar este problema, emplearemos código *Python* y los conjuntos de datos disponibles en la plataforma *OpenDataSF*, **Analysis neighborhoods** y **Current Police Districts**, los cuales contienen información actualizada sobre las latitudes y longitudes asociadas a los distintos distritos policiales y vecindarios. Utilizando estos conjuntos de datos, calcularemos posteriormente las variables '*police_district*' y '*analysis_neighborhood*' de manera precisa.

Este enfoque nos garantiza que los vecindarios y distritos policiales utilizados en nuestras predicciones sean los correctos, de acuerdo con la información más reciente proporcionada por el Departamento de Policía de San Francisco. De esta manera, aseguramos la validez y precisión de los datos geográficos utilizados en nuestro análisis y en las predicciones resultantes.

- Las variables ***'row_id'***, ***'incident_id'***, ***'incident_number'*** ya que nos aportan sólo información de identificación del incidente y no es información relevante para nuestro estudio.
- La variable ***'filed_online'*** ya que no es de nuestro interés predecir si el incidente fue reportado de forma online.
- La variable ***'cad_number'***, no interesa predecir el número del sistema de despacho de policía.
- La variable ***'intersection'*** y ***'cnn'***, aportan información relevante a la latitud y a la longitud. Además no tenemos información sobre el número de calle que define a cada calle por lo que posteriormente en el preprocesado calcularemos la dirección y la calle correspondientes a la latitud y a la longitud por medio de la librería *Geopy* de *Python*.
- Las variables ***'supervisor_district'*** y ***'supervisor_district_2012'*** porque son información redundante entre sí y con la variable distrito policial que será calculada posteriormente en la fase de preprocesado de los datos.
- La variable ***'point'*** debido a que es una tupla calculada a partir de la unión de la latitud y de la longitud.
- Las variables ***'report_type_code'*** y ***'report_type_description'*** ya que proporcionan información sobre el tipo de informe realizado para un incidente, esto abarca información no relevante para nuestro futuro modelo predictivo.

Una vez seleccionadas nuestras variables de interés se obtuvo un conjunto de datos con 717.040 registros y 9 variables.

| **Columna** | **Descripción** |
|---|---|
| *incident_datetime* | Fecha del incidente en formato year-mm-dd hh:mm:ss |
| *incidente_day_of_week* | Día de la semana en el que ocurrió el incidente |

| | |
|---|---|
| *incident_code* | Código del incidente |
| *incident_category* | Categoría del incidente |
| *incident_subcategory* | Subcategoría del incidente |
| *resolution* | Resolución del incidente |
| *latitude* | Latitud geográfica del incidente |
| *longitude* | Longitud geográfica del incidente |
| *description* | Descripción del incidente |

**Figura 13. Variables seleccionadas para el estudio**

- **Etapa 2. Análisis exploratorio de los datos.**

Para situarnos en este análisis primero mostramos las primeras 5 líneas de nuestro conjunto de datos para tener una visión global y posteriormente comenzamos con el análisis exploratorio.

| incident_datetime | incident_day_of_week | incident_code | incident_category | incident_subcategory | incident_description | resolution | latitude | longitude |
|---|---|---|---|---|---|---|---|---|
| 2018-01-01T01:30:00.000 | Monday | 71000 | Lost Property | Lost Property | Lost Property | Open or Active | 37.788721 | -122.402066 |
| 2018-01-01T00:00:00.000 | Monday | 68030 | Courtesy Report | Courtesy Report | Courtesy Report | Open or Active | 37.798442 | -122.409879 |
| 2018-01-01T17:00:00.000 | Monday | 9029 | Fraud | Fraud | False Personation to Receive Money or Property | Open or Active | 37.754736 | -122.507674 |
| 2018-01-01T12:00:00.000 | Monday | 71013 | Larceny Theft | Theft From Vehicle | License Plate, Stolen | Open or Active | 37.737318 | -122.447810 |
| 2018-01-01T16:00:00.000 | Monday | 6304 | Larceny Theft | Larceny Theft - From Building | Theft, From Building, >$950 | Open or Active | 37.764664 | -122.404497 |

**Figura 14. Primeros 5 registros del conjunto de datos**

En base a los registros visualizados en la Figura 14, se ha observado que nuestras variables pueden agruparse en tres categorías principales: variables relacionadas con la fecha y el momento (por ejemplo, *incident_datetime*, *incident_day_of_week*), variables descriptivas del incidente (como *incident_code* e *incident_category*) y variables geográficas (*latitude* y *longitude*).

Conscientes de esta clasificación, se llevó a cabo un análisis exploratorio detallado para cada uno de estos bloques, con el objetivo de comprender mejor la distribución y las características de las variables. Esto nos ha permitido obtener información significativa sobre los patrones temporales, la naturaleza de los incidentes y las ubicaciones geográficas asociadas.

- ***Bloque 1. Variables datetime***

Para comenzar, hemos identificado los valores mínimos y máximos de la variable *incident_datetime*. Esto nos ha permitido determinar el rango de fechas en el cual se

registraron los incidentes, lo cual es crucial para comprender la duración y el alcance de nuestro conjunto de datos. Con esta información, hemos obtenido una visión clara del intervalo de tiempo al que pertenecen los incidentes en estudio.

| | |
|---|---|
| **Valor mínimo** | 01/01/2018 0:00 |
| **Valor máximo** | 11/04/2023 22:39 |

**Figura 15. Valor mínimo y máximo para la variable incident_datetime**

Como se observa los incidentes de nuestro conjunto de datos se produjeron entre el 01/01/2018 y el 11/04/2023.

Tras haber detectado estos valores, separamos la variable en año, mes, día, hora y minutos para realizar un recuento de cada uno de los valores de esta variable y así visualizar su distribución.


**Figura 16. Gráfico de barras, incidentes por año**

A partir de la figura anterior vemos que el año con más incidentes ha sido 2018 y los que menos, el año 2020 debido a la pandemia del COVID y el año 2023 porque únicamente tenemos datos hasta abril.

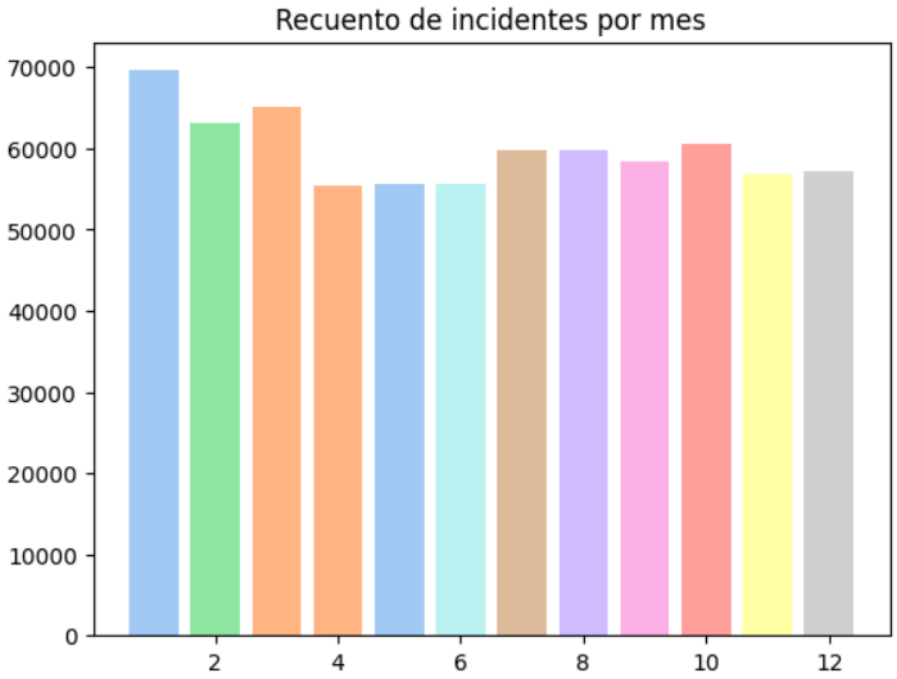


**Figura 17. Gráfico de barras, incidentes por mes**

En la figura anterior se observa que los meses en los que más se producen incidentes en San Francisco son enero, febrero y marzo. Algo que puede estar relacionado con la estación y las temperaturas, y curiosamente, estos tres meses pertenecen a épocas frías del año.

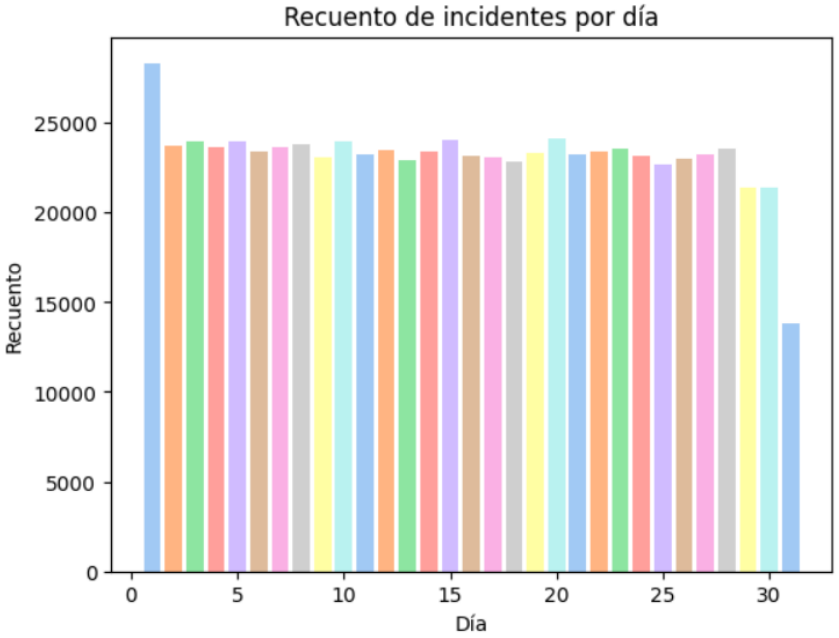


**Figura 18. Gráfico de barras, incidentes por día**

En el gráfico anterior se puede observar que el día 1 es en el que más incidentes se han producido. Esto puede deberse a que los agentes de policía aplazan algunas gestiones burocráticas relacionadas con la recogida de la información de los incidentes para el primer día del mes siguiente. Por otro lado, el día 31 es el que menos incidentes registra, aunque esto se debe a que no todos los meses tienen 31 días.

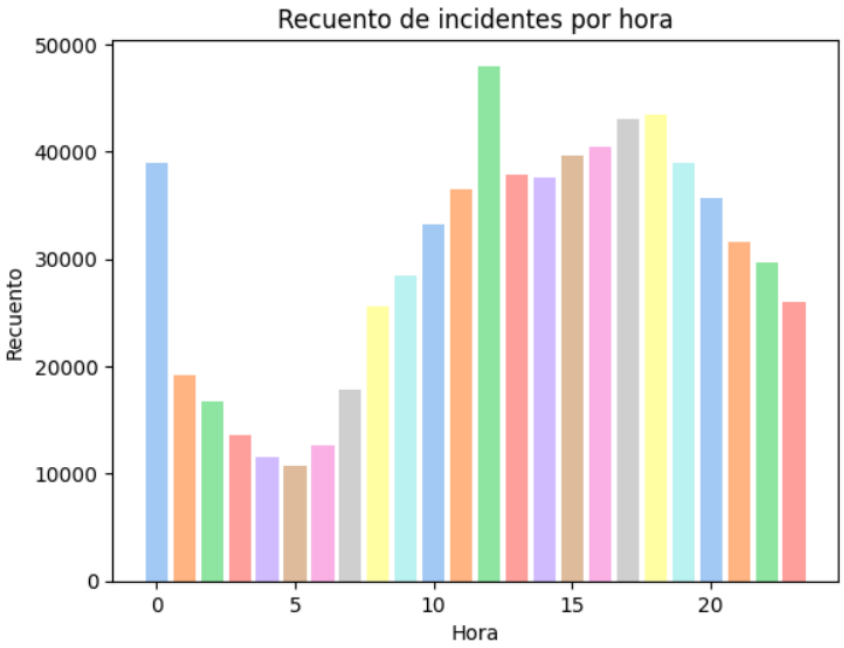


**Figura 19. Gráfico de barras, incidentes por hora**

En la figura anterior se aprecia que las horas en las que más incidentes se producen son 12, 17, 18 y 00 horas. También se observa que las horas en las que menos incidentes tenemos son 4 y 5 de la mañana, algo que parece lógico ya que a esas horas la mayoría de la gente suele estar descansando.

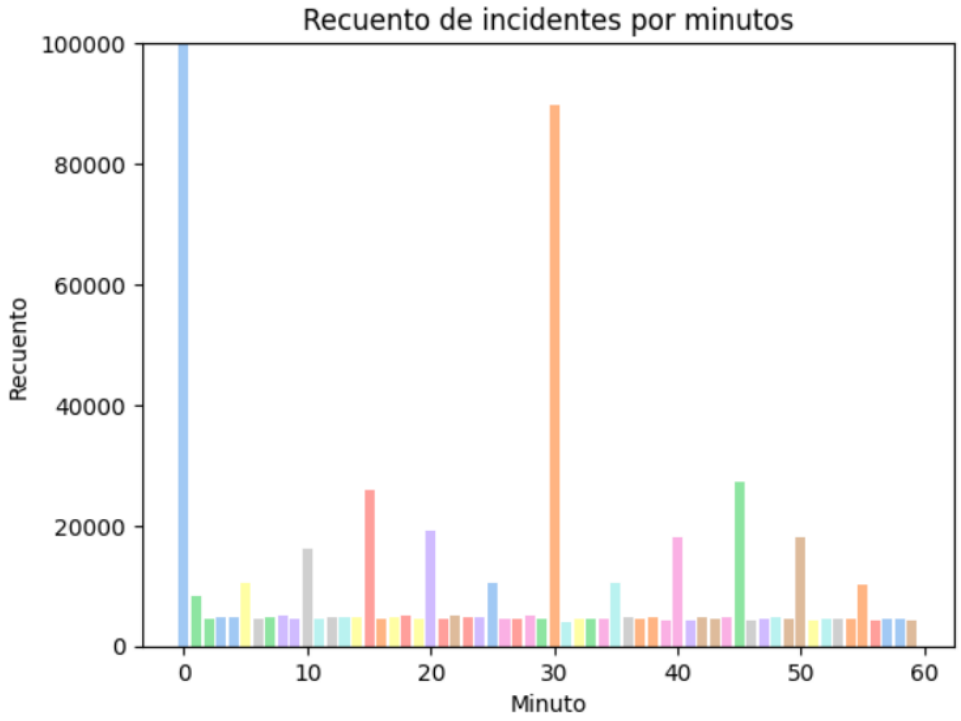


**Figura 20. Gráfico de barras, incidentes por minuto**

En el gráfico anterior se puede ver que la mayoría de los incidentes ocurren en horas en punto, y a las medias horas. Esto también puede deberse a la generalización de las horas que ingresan los agentes de policía en los datos.

Asimismo, también hemos examinado la variable *incident_day_of_week* para comprender la distribución de los incidentes en función de los días de la semana.

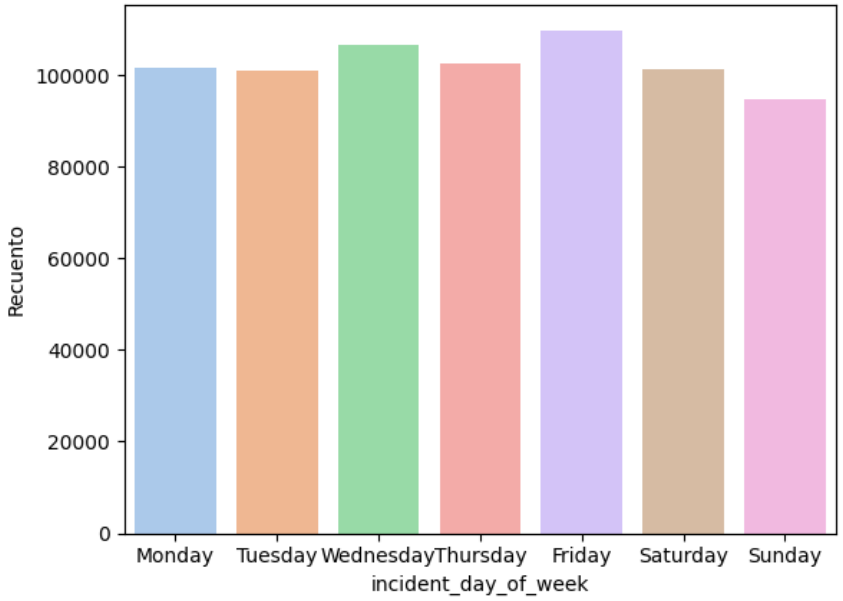


**Figura 21. Gráfico de barras, incidentes por *incident_day_of_week***

Vemos a partir del gráfico anterior que no hay mucha variabilidad entre el número de incidentes para cada día de la semana, aunque el día en el que más incidentes se producen es el viernes.

Durante el análisis de las variables *incident_datetime* y *incident_day_of_week*, se ha procurado detectar casos anómalos y atípicos y se ha observado que la distribución de los valores, se mantuvo relativamente constante a lo largo de las fechas registradas, sin desviaciones o valores atípicos que pudieran indicar anomalías en la ocurrencia de los incidentes. Esto sugiere que los datos recopilados son consistentes y no presentan observaciones inusuales en términos de las variables de fecha y día de la semana.

- ***Bloque 2. Variables descriptivas del incidente***

Como punto de partida para el análisis exploratorio de estas variables, se generó un gráfico de recuento que muestra la distribución de los distintos valores que estas variables pueden tomar. Esta representación visual nos permite tener una visión general de cómo se distribuyen los datos.

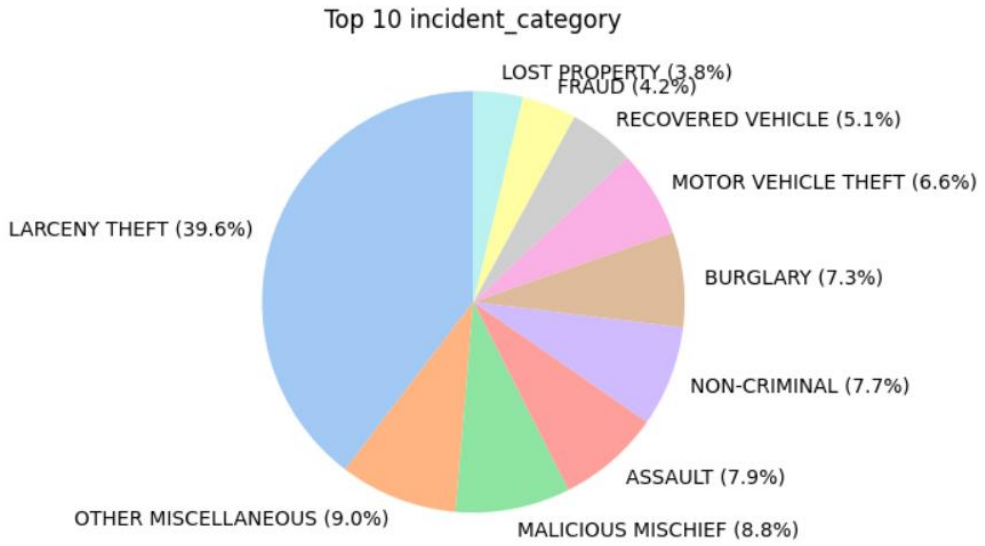


**Figura 22. Gráfico circular para *incident_category***

En la figura anterior se puede observar que las categorías más presentes en nuestro conjunto de datos son Robo sin fuerza (*Larceny Theft*), Otros diversos (*Other miscellaneous*), y vandalismo malicioso (*Malicious Mischief*).

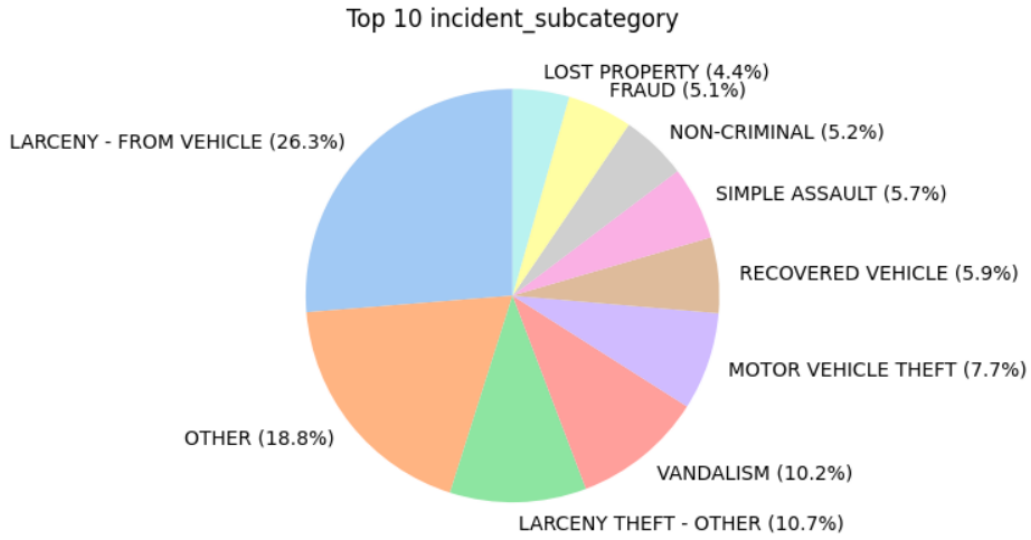


**Figura 23. Gráfico circular para *incident_subcategory***

A partir del gráfico anterior se aprecia que las subcategorías más presentes en nuestro conjunto de datos son Robo de vehículo (*Larceny - from vehicle*) Otros (*Other*) y Robo sin fuerza - otros (*Larceny theft - other*).

Como estas tres variables conforman una tupla, también se realizó un recuento de incidentes agrupando por las variables *incident_code*, *incident_category* y *incident_subcategory*.

| ***incident_code*** | ***incident_category*** | ***incident_subcategory*** | **Recuento** |
|---|---|---|---|
| 6244 | *Larceny Theft* | *Larceny - From Vehicle* | 90769 |
| 28150 | *Malicious Mischief* | *Vandalism* | 23844 |
| 4134 | *Assault* | *Simple Assault* | 21176 |
| 71000 | *Lost Property* | *Lost Property* | 20827 |
| 7041 | *Recovered Vehicle* | *Recovered Vehicle* | 20310 |
| 6372 | *Larceny Theft* | *Larceny Theft - Other* | 20220 |
| 7021 | *Motor Vehicle Theft* | *Motor Vehicle Theft* | 19657 |
| 6374 | *Larceny Theft* | *Larceny Theft - Other* | 17137 |
| 64020 | *Non-Criminal* | *Other* | 15785 |
| 6224 | *Larceny Theft* | *Larceny - From Vehicle* | 13686 |

**Figura 24. Top 10 Recuento de incidentes por *incident_code*, *incident_category* y *incident_subcategory*.**

A partir de la tabla anterior podemos observar de forma conjunta los resultados obtenidos en las figuras 22 y 23.

En resumen, los incidentes más frecuentes son:

- **Incidente 830403 (Robo de objetos), Categoría: *Larceny Theft* (Robo de objetos), Subcategoría: *Larceny From Vehicle* (Robo de objetos de vehículo)- Recuento de 90769**. Los robos de objetos de vehículos representan una preocupación significativa en San Francisco, siendo uno de los incidentes más frecuentes identificados en los datos analizados.
- **Incidente 2446483 (Daño intencional), Categoría: *Malicious Mischief* (Daño intencionado) , Subcategoría: *Vandalism* (Vandalismo) - Recuento: 23844.** Los actos de vandalismo constituyen otro problema recurrente en la ciudad, lo cual ha sido evidenciado por el elevado número de casos registrados.
- **Incidente 451712 (Agresión), Categoría: *Assault* (Agresión), Subcategoría: *Simple Assault* (Agresión simple) - Recuento: 21176.** Los incidentes de agresión de menor gravedad se encuentran entre los eventos más comunes, indicando una preocupación en términos de seguridad pública.

- **Incidente 2901492 (Propiedad perdida), Categoría: *Lost Property* (Propiedad perdida), Subcategoría: *Lost Property* (Propiedad perdida) - Recuento: 20827.** La pérdida de objetos personales constituye una incidencia frecuente en San Francisco, generando la necesidad de acciones preventivas y de educación para fomentar la responsabilidad en el cuidado de las pertenencias.

En este bloque para la detección de casos anómalos y atípicos se hizo uso del archivo ***Reference Incident Code Crosswalk***, en el que se encuentra la correspondencia entre los códigos de incidente, la categoría y su correspondiente subcategoría. Para realizar este estudio lo que se hizo fue crear una tupla formada por los valores de las variables *incident_code, incident_category* y *incident_subcategory* para cada registro de nuestro conjunto de datos y se revisó si esa tupla existía en el dataset publicado en la plataforma *OpenDataSF*.

Antes de realizar la comparativa se hizo conversión a minúsculas de nuestras variables y de las descargadas para realizar el estudio de forma correcta.

Posteriormente realizamos una búsqueda de cada una de nuestras tuplas en el archivo descargado y se encontró con 615 registros cuyas tuplas (*incident_code, incident_category, incident_subcategory*) no se encontraban en el archivo de referencia.

Con el objetivo de detectar si existía algún patrón en estos 615 registros, decidimos hacer un recuento de los incidentes agrupados por esta tupla para dar con las tuplas que no se estaban detectando como correctas. Al ejecutar el código para realizar esta agrupación nos encontrábamos ante una salida vacía, por lo que decidimos mostrar los primeros 5 registros en los que la tupla no coincide con los datos de referencia.

| incident_datetime | incident_day_of_week | incident_code | incident_category | incident_subcategory | incident_description | resolution | latitude | longitude |
|---|---|---|---|---|---|---|---|---|
| 2018-02-09 10:17:00 | Friday | 12075 | NaN | NaN | Military Ordinance | Open or Active | 37.762342 | -122.450737 |
| 2018-02-09 10:17:00 | Friday | 12075 | NaN | NaN | Military Ordinance | Open or Active | 37.762342 | -122.450737 |
| 2018-02-24 10:00:00 | Saturday | 12075 | NaN | NaN | Military Ordinance | Unfounded | 37.764898 | -122.400633 |
| 2018-03-07 16:48:00 | Wednesday | 12075 | NaN | NaN | Military Ordinance | Open or Active | 37.759737 | -122.479005 |
| 2018-03-13 10:07:00 | Tuesday | 12075 | NaN | NaN | Military Ordinance | Open or Active | 37.788364 | -122.445589 |

**Figura 25. Primeros 5 registros con tuplas (*incident_code, incident_category, incident_subcategory*) inválidas**

Como observamos en estos registros nuestras variables *incident_category* y *incident_subcategory* se encuentran como valores NaN.

Para ver si la causa por la que las 615 tuplas que no han coincidido con el archivo de referencia se debe a valores faltantes realizamos un recuento de nulos en los registros que hemos clasificado como inválidos.

| Variable | Recuento de NaN |
|---|---|
| *incident_category* | 615 |
| *incident_subcategory* | 615 |

**Figura 26. Recuento de nulos para *incident_category* y *incident_subcategory* en los registros con tuplas inválidas.**

Observamos que todos los registros en los que la tupla no se ha encontrado es porque tiene las variables *incident_category* y *incident_subcategory* como valores faltantes.

El tratamiento de estos datos ausentes se determinará en la Fase 3. Preparación de los datos.

Además de las variables anteriores, también teníamos presentes en nuestro conjunto de datos las variables *resolution* y *description*. Para el análisis de la primera realizamos un gráfico de barras para visualizar la distribución de la variable.

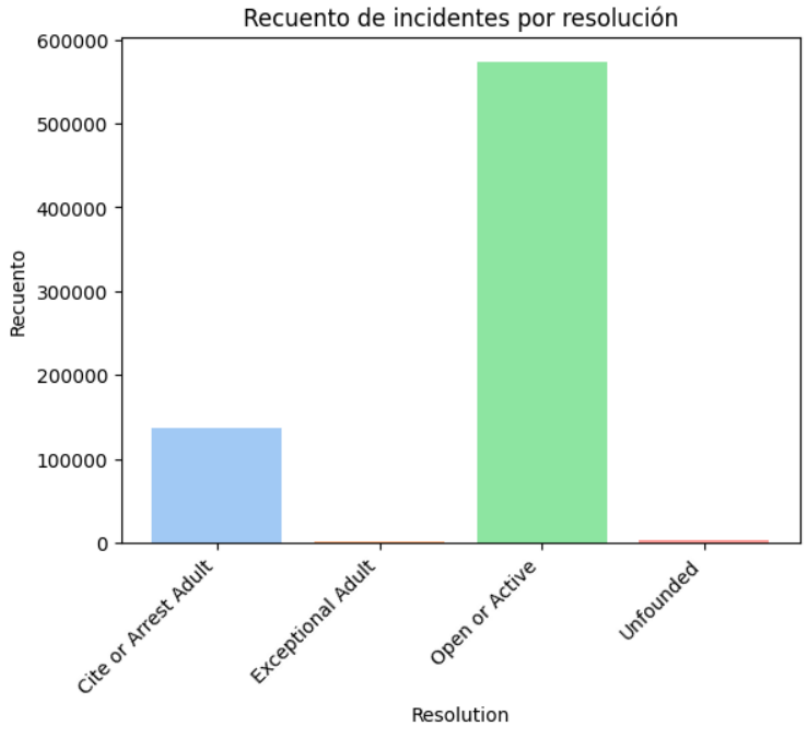


**Figura 27. Gráfico de barras, incidentes por *resolution***

Como observamos en el gráfico anterior, la mayoría de incidentes de nuestro conjunto de datos han tenido como resolución Abierto o activo (*Open or Active)*. Asimismo, vemos que el resto de incidentes han tenido como resolución Cita o Arresto de Adulto (*Cite or Arrest Adult)*.

También vemos que hay muy pocos incidentes que terminan con una resolución de delito sexual por adulto a un menor (*Exceptional Adult* o *Unfounded)*.

La variable *resolution* en el conjunto de datos de incidentes de San Francisco describe el resultado final de la investigación del incidente.

A continuación se presentan las descripciones de los diferentes valores que puede tomar:

| | |
|---|---|
| ***NONE*** | No se ha tomado ninguna medida para resolver el incidente. |
| ***OPEN O ACTIVE*** | El incidente está en curso y aún no se ha resuelto. |
| ***ARREST, BOOKED*** | Se realizó un arresto y el sospechoso fue detenido y fichado. |
| ***ARREST, CITED*** | El sospechoso fue citado y liberado, pero se espera que se presente en la corte en una fecha posterior. |
| ***CITE OR ARREST ADULT*** | El sospechoso fue detenido y liberado después de ser citado y acusado formalmente de un delito. |
| ***PSYCHOPATHIC CASE*** | El incidente involucró a una persona con trastornos mentales y se tomaron medidas para garantizar su seguridad y la de los demás. |
| ***UNFOUNDED*** | Se determinó que el incidente no ocurrió como se informó inicialmente. |
| ***LOCATED*** | La persona o propiedad perdida o robada fue encontrada. |
| ***COMPLAINANT REFUSES TO PROSECUTE*** | El denunciante se negó a presentar cargos contra el sospechoso |
| ***NOT PROSECUTE*** | El sospechoso fue arrestado, pero no se presentaron cargos contra él. |
| ***DISTRICT ATTORNEY REFUSES TO PROSECUTE*** | El fiscal de distrito se negó a presentar cargos contra el sospechoso. |
| ***PROSECUTED BY OUTSIDE AGENCY*** | El sospechoso fue arrestado y acusado de un delito por una agencia policial diferente. |
| ***EXCEPTIONAL CLEARANCE*** | Se resolvió el incidente, pero no se realizó ningún arresto. |

| | |
|---|---|
| ***EXCEPTIONAL ADULT*** | Un caso en el que un adulto es sospechoso de cometer un delito sexual y la víctima tiene menos de 18 años. |
| ***PROSECUTED FOR LESSER OFFENSE*** | El sospechoso fue arrestado y acusado de un delito menor. |

**Figura 28. Valores variable *Resolution***

Pasando a analizar la variable description, con el objetivo de visualizar si es un campo libre o si tiene valores discretos realizamos un análisis exploratorio con la intención de determinar si se debe incluir esta variable como variable de entrada a nuestro modelo.

```
incident_description
Theft, From Locked Vehicle, >$950                    90769
Malicious Mischief, Vandalism to Property            23844
Battery                                              21176
Lost Property                                        20827
Vehicle, Recovered, Auto                             20310
                                                     ...
Sale or Manufacture of Deceptive ID                      1
Sale of Satellite Telephone Numbers                      1
Robbery, Vehicle for Hire, Att., W/ Other Weapon         1
Robbery, Vehicle for Hire, Att., W/ Gun                  1
Wiretaps, Unauthorized                                   1
```

**Figura 29. Recuento de valores para la variable *description***

A partir de la figura anterior se aprecia que el campo *description* de cada incidente, es un campo libre por lo que para no introducir ruido a nuestro modelo en la Fase 3. Preparación de los datos. se eliminará esta variable.

Finalmente se hizo un recuento de valores faltantes por cada variable, en la siguiente figura se puede visualizar la proporción de valores faltantes dentro de nuestro conjunto de datos.

- ***Bloque 3. Variables geográficas***

Por último, las variables geográficas presentes en el conjunto de datos estudiado son latitud y longitud. Estas variables se han de estudiar de forma exhaustiva ya que serán las variables utilizadas para el cálculo del vecindario, *neighborhood* y el distrito policial, *police_district*.

Para el análisis de esta variables creamos una función mediante código *Python* que determinaba si la tupla conformada por la latitud y longitud de nuestro incidente pertenecía, o no, a San Francisco, también nos indicaba si la latitud y la longitud no eran reales o correctas.

Para ello, empleamos el archivo de datos ***Bay Area Countries*** que contiene todos los polígonos geográficos que conforman el área de San Francisco. Tras ejecutar nuestra función sobre nuestros registros encontramos un total de 120 registros que no pertenecían a San Francisco, por lo que estas tuplas de latitud y longitud nos parecían tuplas atípicas. Para asegurarnos empleamos un segundo archivo de datos, ***Current Police Districts***, en el que tenemos todos los polígonos que definen a cada distrito policial de San Francisco y revisamos si esos 120 casos pertenecían a algún distrito

policial de San Francisco. La función creada en esta validación nos reveló que estos 120 casos no pertenecían tampoco a ningún distrito por lo que en la siguiente etapa Fase 3. Preparación de los datos se determinará el tratamiento de estos valores atípicos.

Finalmente se hizo un recuento de valores faltantes por cada variable, en la siguiente figura se puede visualizar la proporción de valores faltantes dentro de nuestro conjunto de datos.

| Variable | Recuento de NaN | % NaN |
|---|---|---|
| *Incident_datetime* | 0 | 0.00% |
| *incident_day_of_week* | 0 | 0.00% |
| *incident_code* | 0 | 0.00% |
| *incident_category* | 615 | 0.09% |
| *incident_subcategory* | 615 | 0.09% |
| *resolution* | 0 | 0.00% |
| *latitude* | 38192 | 5.33% |
| *longitude* | 38192 | 5.33% |

**Figura 30. Recuento de valores ausentes por variable**

En la siguiente etapa, se eliminará la variable *description*, y se determinará el tratamiento para los 120 registros cuyas latitudes y longitudes no pertenecen a San Francisco y para los datos ausentes que hemos localizado en la Figura 30.

Todo el código implementado para esta etapa se encuentra publicado en el archivo Cleaning_sf_Incidents_part1 del *GitHub* creado para este proyecto.

## Fase 3. Preparación de los datos.

Una vez realizado el análisis exploratorio del apartado anterior se procedió a preparar nuestro conjunto de datos para que sea una base sólida sobre la cuál construir los modelos estadísticos. Para ello tendremos en cuenta los siguientes pasos:

- **Etapa 1. Tratamiento de valores atípicos y ausentes.**

Durante la fase previa de nuestro estudio, se identificaron ciertas particularidades en los datos. En primer lugar, se observó que la variable *description* presentaba múltiples valores únicos, lo cual indicaba que se trataba de un campo de texto libre. Para evitar introducir ruido en nuestro modelo, se decidió eliminar esta variable.

Además, se detectaron 120 registros cuyas coordenadas de latitud y longitud no corresponden al polígono de San Francisco ni a ningún distrito policial conocido. Dado que estos registros no estaban dentro del alcance geográfico de nuestro estudio, se optó por descartarlos del conjunto de datos.

En relación a los valores ausentes, se identificaron 38,192 registros en los cuales las coordenadas de latitud y longitud presentaban valores faltantes. Dentro de este

subconjunto, se encontraron 616 registros en los cuales la categoría y subcategoría del incidente también estaban ausentes (NaN).

Como nuestro conjunto de datos era muy grande, procedimos a estudiar si la presencia de estos valores era de patrón aleatorio, para ello se realizó un mapa de calor para visualizar de forma gráfica la existencia o no existencia de este patrón.

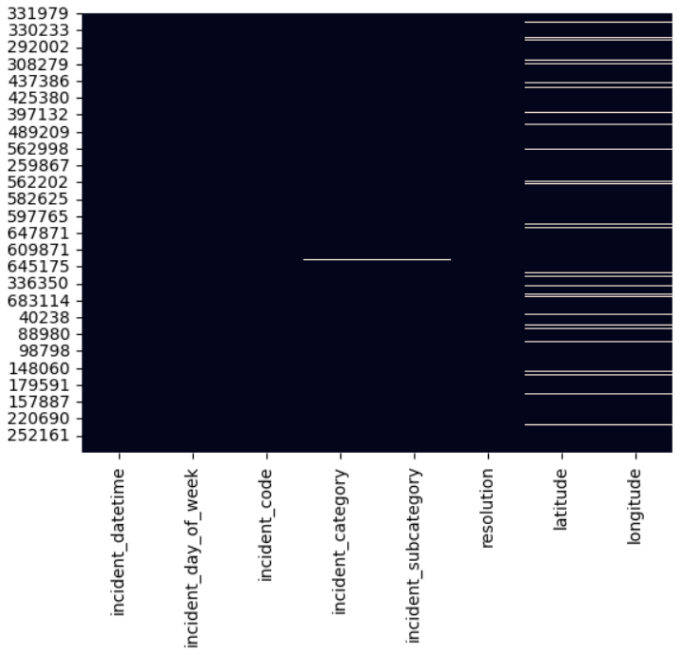


**Figura 31. Mapa de calor valores ausentes**

Como vemos en la figura anterior, los valores ausentes de nuestro conjunto de datos se deben a un patrón aleatorio, por lo que debido al tamaño considerable de nuestro conjunto de datos principal, se decidió eliminar los registros con valores de latitud y longitud que no pertenecían a San Francisco, así como aquellos registros con valores faltantes. Esta medida fue adoptada con el propósito de evitar la necesidad de realizar imputaciones y minimizar la introducción de ruido en nuestro modelo.

Tras esta eliminación obtuvimos un conjunto de datos con 678125 registros y 8 variables.

- **Etapa 2. Cálculo de variables**

En esta etapa se desarrolló código *Python* para calcular las variables relacionadas con las características geográficas del incidente y las temporales. Las variables calculadas fueron las siguientes:

- ***Street***. El valor de la calle para cada uno de los registros se obtuvo creando una función que realizaba una solicitud a la librería de *Geopy* proporcionando como argumentos la latitud y la longitud.
- ***Neighborhood***. El vecindario se obtuvo mediante una función creada que hacía uso del archivo ***Analysis neighborhoods***, proporcionando como argumentos la latitud y la longitud de cada registro. Esta función buscaba el polígono al que pertenecía el registro y su correspondiente vecindario.

- ❖ ***Police_district***. El distrito policial se calculó creando una función que exploraba el archivo ***Current Police Districts***, y a partir de la latitud y la longitud devolvía el nombre del distrito al que pertenecía ese punto geográfico.
- ❖ ***Holiday***. Esta variable se calculó para determinar si el día en el que ocurrieron los incidentes eran días festivos o no.

  Para el cálculo de esta variable se creó una función que descargaba los días festivos en San Francisco y de California durante los años analizados y para cada registro comprobaba si el día del incidente era un día festivo en San Francisco, en California o en ambos. La librería que permitió crear esta función fue la librería *Holidays*.

Para ejecutar todas estas funciones sobre nuestro conjunto de datos, debido a la gran cantidad de datos y a los escasos recursos, se aplicó una paralelización de procesos aprovechando todos los hilos del procesador del ordenador mediante la librería concurrent.futures. El código empleado en esta parte del proyecto se encuentra publicado en el archivo Preprocess_data_part1 del *GitHub* creado para este proyecto.

- **Etapa 3. Análisis exploratorio de las variables calculadas**

Una vez calculadas las variables de nuestro interés procedimos a realizar un análisis exploratorio sobre las nuevas variables y una posterior limpieza en el caso de detectar datos anómalos, atípicos o ausentes.

❖ ***Neighborhood***

Para analizar esta variable realizamos un gráfico de barras que nos mostraba el recuento de incidentes por cada vecindario.

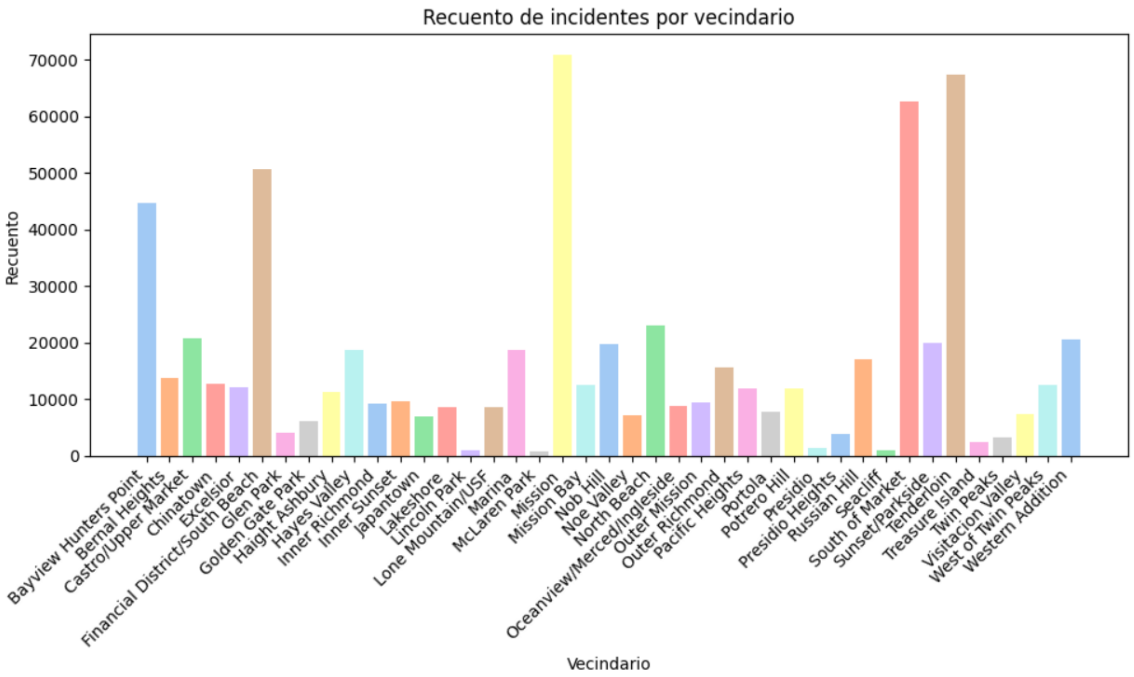


**Figura 32. Gráfico de barras para *neighborhood***

En la Figura 32, se aprecia que los vecindarios en los que más incidentes ocurren son *Mission, Tenderloin, South of Market* y *Financial District/ South Beach*.

En el análisis exploratorio de esta variable no se aplicó ninguna técnica de detección de atípicos y anómalos debido a que se calculó a partir del archivo de datos, ***Analysis neighborhoods***, publicado por el departamento de planificación del territorio de San Francisco en la plataforma *OpenDatSF*.

- ***Police_district***

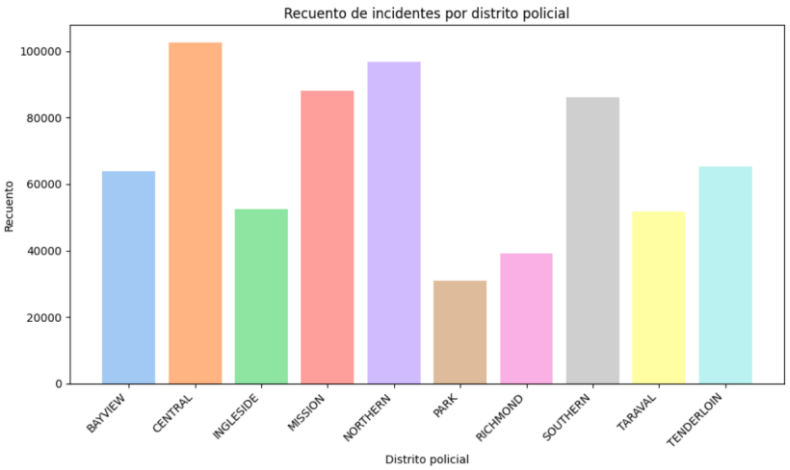


**Figura 33. Gráfico de barras para distrito policial**

En la Figura 33, se observa que los distritos policiales en los que más incidentes ocurren son *Central, Nothern, Mission* y *Southern*.

En el análisis exploratorio de esta variable no se aplicó ninguna técnica de detección de atípicos y anómalos debido a que se calculó a partir del archivo de datos, ***Current Police Districts***, publicado por el departamento de planificación del territorio de San Francisco en la plataforma *OpenDatSF*.

- ***Holiday***

Al realizar el análisis exploratorio de la variable calculada holiday encontramos que esta variable se había calculado mal ya que para todos los registros tomaba el valor '*Both*' qué significa que el día era festivo tanto para California como para San Francisco.

Por ello, procedimos a descargar de nuevo los días festivos en California y San Francisco y revisamos que ambas zonas tuvieran los mismos días festivos del año 2018 al año 2023. Al realizar esta validación se detectó que la función empleada para el cálculo de esta variable no funcionaba de forma correcta.

Para corregir el anterior error, se creó una segunda función que únicamente nos indicaba si el día era festivo en San Francisco o no.

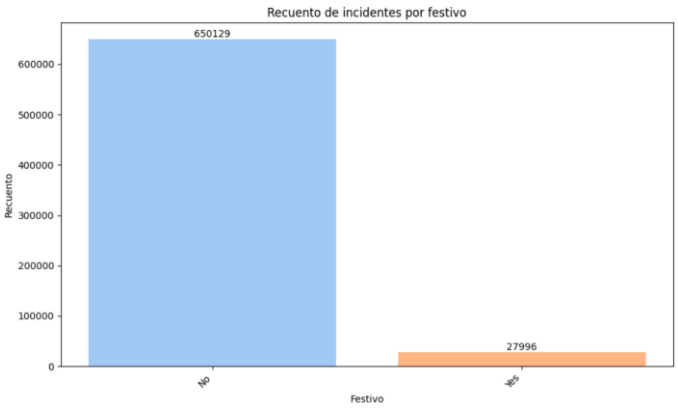


**Figura 34. Gráfico de barras para festivo**

En la Figura 34, se observa que los 27996 días en los que se han producido incidentes eran festivos, mientras que 650129 días eran días no festivos.

- ***Street***

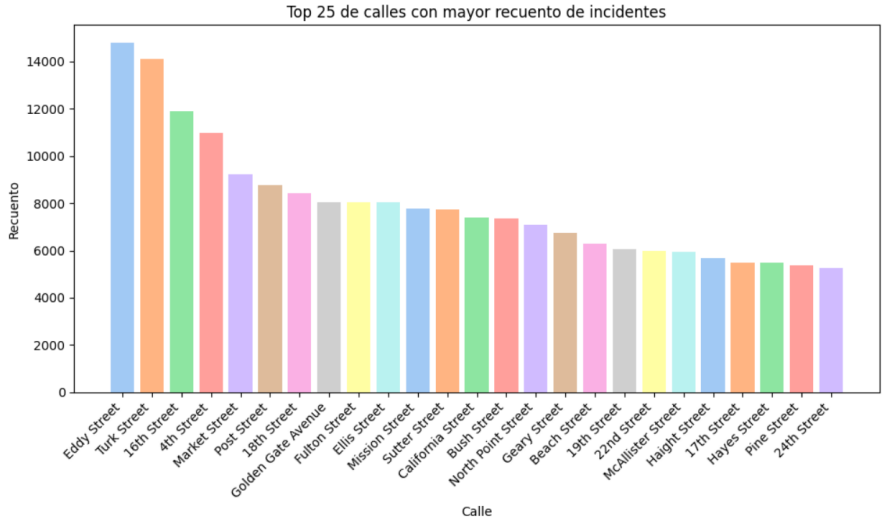


**Figura 35. Gráfico de barras para calle**

En la Figura 35, se observa que las calles en las que más incidentes ocurren son *Eddy Street, Turk Street, 16th Street* y *4th Street*.

En lo que respecta a la detección de valores atípicos en la variable Street lo que se hizo fue comprobar que todos los valores de esta variable existían en el conjunto de datos, ***Street Names***. Este archivo está publicado en la plataforma *OpenDatSF* y contiene todas las calles de la ciudad.

Para ello primeramente tuvimos que aplicar transformaciones a nuestra variable ya que en nuestra variable tenemos el tipo de calle sin abreviar mientras que en el archivo está abreviado.

```
print(Incidents_P['Street'][:5])
```

```
0    Fulton Street
1      16th Street
2        Navy Road
3    Sadowa Street
4      Hyde Street
```

**Figura 36. Formato calles obtenidas del preprocesado, sin abreviatura**

Tras realizar la transformación conseguimos tener el mismo formato que en el archivo de referencia, Figura 36.

```
0          FULTON ST
1            16TH ST
2            NAVY RD
3          SADOWA ST
4            HYDE ST
              ...
678120      BEACH ST
678121     ORTEGA ST
678122     CASTRO ST
678123      HAYES ST
678124      LINDA ST
```

**Figura 37. Formato calles obtenidas tras la transformación, con abreviatura**

Una vez comprobado que el formato entre ambos conjuntos es el mismo se creó una función que indicaba si la calle se encontraba en el archivo de referencia o no, en el caso de no encontrarse devolvía el valor de *'ATYPICAL'*.

Tras ejecutar esta función, obtuvimos que 42381 calles eran *'ATYPICAL'*, es decir que no se encuentran en el archivo estándar.

```
Incidents_P['Street_Atypical'].value_counts()
```

```
NO          596773
ATYPICAL     42381
```

**Figura 38. Recuento de valores atípicos de la variable calle**

Una vez detectadas estas calles mostramos por pantalla algunos de estos casos.

```
filetered_data = Incidents_P[Incidents_P['Street_Atypical']=='ATYPICAL']
print(filetered_data['Street'][:5])
```

```
6                    4TH ST
51                   4TH ST
68    CHARLES J. BRENHAM PL
75              O'FARRELL ST
86                VERMEHR PL
```

**Figura 39. Valores de la variable Street no encontrados en las referencias**

Como el registro 68 contiene un nombre propio se trató de localizar en el archivo de referencia las calles que contuviesen ese nombre propio para verificar si es que el problema es que las calles son las mismas pero el nombre varía ligeramente.

```
filtered_data = street_file[street_file['FullStreetName'].str.contains('BRENHAM', case=False)]
print(filtered_data['FullStreetName'])
```

```
497    CHARLES J BRENHAM PL
```

**Figura 40. Búsqueda de la cadena 'BRENHAM' en las calles de San Francisco**

Observamos la existencia de una calle que contiene la cadena *'BRENHAM'* pero que nuestra función no ha detectado debido a que en nuestra variable calculada tenemos la presencia de un punto, por lo que al hacer la búsqueda no la encuentra.

Para evitar este tipo de errores se modificó la función para que no tuviera en cuenta los carácteres especiales. Tras realizar esta codificación y ejecutar de nuevo la función conseguimos reducir la frecuencia de calles atípicas de un valor de 42381, Figura 38 a 35535, Figura 41.

```
Incidents_P['Street_Atypical'].value_counts()
```

```
NO          603619
ATYPICAL     35535
```

**Figura 41. Valores 'ATYPICAL' en la variable *Street* tras la modificación de la función**

Vemos que se ha conseguido reducir el número de atípicos, pero que siguen existiendo valores atípicos. Parece ser que las calles no están nombradas de la misma forma entre el conjunto de datos principal y el conjunto de datos de referencia, publicado en la plataforma *OpenDataSF*.

Para comprobar que no es únicamente la presencia de carácteres se calculó un índice de similitud entre las calles calculadas y la calle más parecida en el conjunto de datos descargado con los nombres correctos de las calles.

Esto se realizó mediante la librería ***fuzzywuzzy*** que nos permitió buscar de forma eficiente la cadena más cercana y calcular el índice de similitud. Por ello, realizamos el cálculo de dos columnas, una con la calle más parecida y otra con el ratio o índice de similitud. Posteriormente estudiamos esos ratios.

```
Street SimilarStreet  SimilarityIndex
4TH ST        04THST               83
4TH ST        04THST               83
4TH ST        04THST               83
4TH ST        04THST               83
3RD ST        03RDST               83
```

**Figura 42. Columnas de estudio para la detección de valores atípicos**

En la Figura 42, vemos que la diferencia entre las calles más parecidas en algunos casos es de un número. Para este caso, el índice de similitud es de 83.

El índice de similitud puede tomar valores del intervalo de 0 a 100, de forma que cuánto más alto sea este, más parecidas son las cadenas estudiadas.

```
Mínimo índice de similitud: 43
Máximo índice de similitud: 93
```

**Figura 43. Valores máximos y mínimos para el índice de similitud**

En la Figura 43, se aprecia que los índices de similitud calculados están entre 43 y 93.

Para continuar con el estudio de estos casos, ordenamos las calles en función del índice de similitud de forma ascendente para así observar qué diferencias existen entre las calles con un índice de similitud bajo.

```
Similarity Index: 43
Count: 227
Street Value: OCTAVIA BOULEVARD BIKE CONNECTOR
Correct Street: AMBROSEBIERCEST

Similarity Index: 62
Count: 9
Street Value: SKUNK ALY
Correct Street: PINKALY

Similarity Index: 67
Count: 682
Street Value: ECKER PLZ
Correct Street: DECKERALY

Similarity Index: 69
Count: 37
Street Value: MIDDLE DRIVE WEST
Correct Street: MIDDLEWESTDR

Similarity Index: 70
Count: 104
Street Value: CLEO RAND LN
Correct Street: CLEORANDAVE

Similarity Index: 71
Count: 68
Street Value: GOLDEN GATE BRIDGE
Correct Street: GOLDENGATEAVE

Similarity Index: 72
Count: 29
Street Value: CHAIN OF LAKES DRIVE EAST
Correct Street: CHAINOFLAKESDR

Similarity Index: 75
Count: 32
Street Value: BURNETT NORTH AVE
Correct Street: BURNETTAVENORTH

Similarity Index: 76
Count: 183
Street Value: GREAT HIGHWAY
Correct Street: GREATHWY

Similarity Index: 77
Count: 91
Street Value: COLIN P. KELLY JUNIOR ST
Correct Street: COLINPKELLYJRST

Similarity Index: 78
Count: 88
Street Value: LA PLAYA ST
Correct Street: LAPLAYA
```

**Figura 44. Calles atípicas, calles de referencia e índice de similitud, ordenadas en función de la similitud de forma ascendente.**

En la Figura 44, se observa que las cuatro primeras calles presentan un índice de similitud más bajo, lo cual indica que las cadenas de texto no tienen una similitud significativa entre ellas.

En cambio, para el resto de calles se puede apreciar un alto grado de similitud entre las dos cadenas comparadas. Las diferencias notables se encuentran principalmente en la presencia o ausencia de espacios y en la forma de abreviar el nombre de la calle.

En el caso específico de la calle *BURNETT NORTH AVE*, podemos observar que en nuestro conjunto de datos esta calle se encuentra registrada como *BURNETTAVENORTH* en el archivo de referencia. A pesar de la diferencia en la forma de expresar el nombre de la calle, es evidente que se refieren al mismo lugar. Esta variación en el orden de la orientación y el tipo de calle es común en la ciudad y no implica una diferencia significativa. Debido a que la diferencia en estos casos no es muy relevante ni supone diferencias entre los nombres de la calle, estos casos serán considerados como similares y no serán tratados como atípicos.

Los casos más preocupantes son los 4 primeros registros de la Figura 44, en los que no tienen ningún parecido el nombre de las calles a primera vista.

Tras analizar detenidamente estos casos, decidimos investigar la biblioteca utilizada para determinar la validez de los valores obtenidos. Durante nuestra investigación, descubrimos que *Geopy* es una biblioteca ampliamente reconocida y utilizada por programadores de *Python* para la geocodificación. Esta biblioteca ofrece una serie de funcionalidades que facilitan la conversión de direcciones en coordenadas geográficas.

Encontramos información relevante sobre *Geopy* en varios recursos en línea, como el artículo "15 librerías de Python para GIS" - MappingGIS , así como en el artículo MappingGIS. (2018). Geocodificación con GeoPy. Estos recursos proporcionan una visión general de las capacidades de *Geopy* y brindan instrucciones detalladas sobre cómo utilizarla en proyectos de geocodificación.

Tras revisar la documentación y los ejemplos proporcionados en estos recursos, podemos afirmar que *Geopy* es una opción confiable y popular para la geocodificación en Python. Su uso extensivo en la comunidad de programadores respalda su fiabilidad y precisión en la conversión de direcciones en coordenadas geográficas.

Para terminar con el análisis exploratorio de nuestras variables realizamos un recuento de valores ausentes para cada variable para ver si se habían calculado todas las nuevas variables para todos los registros de nuestro conjunto de datos. Al realizar este recuento vimos que existían valores NaN en las variables *neighborhood, street, holiday* y *police_district*.

| **Variable** | **Recuento de NaN** |
|---|---|
| *neighborhood* | 132 |
| *Police_district* | 657 |
| *Street* | 0 |
| *Holiday* | 38971 |

**Figura 45. Recuento de valores ausentes para las variables calculadas**

En la Figura 45, se aprecia que tenemos valores ausentes en todas las variables calculadas a excepción de la variable Holiday. Para la que más valores ausentes tenemos es para la variable *Street*, seguida de *Police_district* y *neighborhood*. En la siguiente etapa, se llevará a cabo un análisis exhaustivo para comprender la naturaleza de estos casos y se tomarán medidas para tratarlos de manera adecuada y evitar sesgos en nuestros resultados. El código empleado en esta parte del proyecto se encuentra publicado en el archivo Cleaning_DF_part2 del *GitHub* creado para este proyecto.

- **Etapa 4. Preprocesado de los datos**

En esta etapa del estudio se realizó un exhaustivo análisis del motivo de los valores ausentes en las variables calculadas. Se determinó el tratamiento más apropiado,

considerando la naturaleza de los datos y el contexto del estudio. Posteriormente, se procedió a la codificación de las variables necesarias para poder aplicar modelos estadísticos y realizar análisis más profundos. También se aplicó un *clustering* con el objetivo de estudiar los diferentes patrones y tendencias existentes en nuestro conjunto de datos.

Los pasos seguidos en esta etapa fueron:

- ***Estudio y tratamiento de los valores ausentes***

En la Figura 46, observamos que las variables *neighborhood*, *Street* y *Police_district* tenían valores faltantes, para realizar un estudio exhaustivo lo primero que se hizo fue analizar las latitudes y las longitudes de estos casos que fueron la base para el cálculo de estas variables. Para ello, mostramos en un mapa de San Francisco estas coordenadas geográficas para visualizar si pertenecen a una zona en concreto.

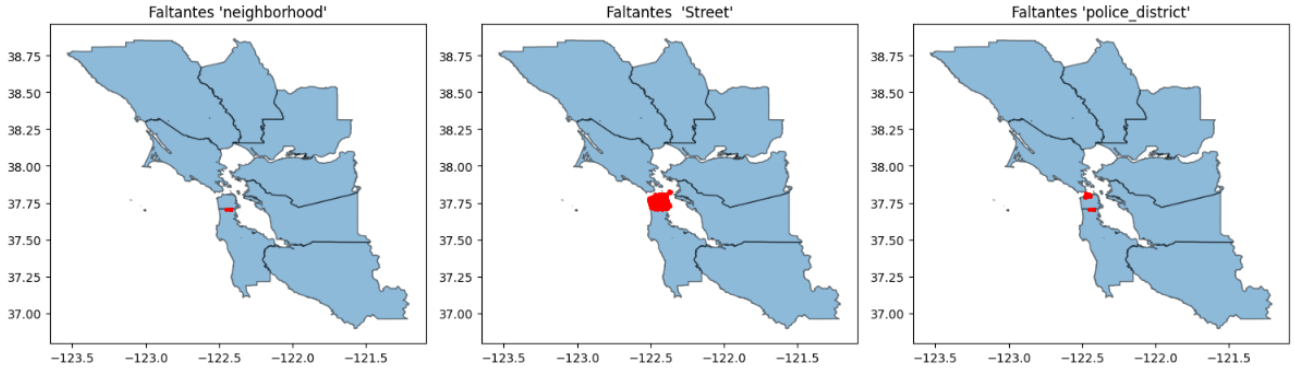


**Figura 46. Coordenadas geográficas de valores ausentes en las variables calculadas**

Como observamos en la figura anterior, todos los faltantes pertenecen a la misma zona geográfica, esto se debe a que los conjuntos de datos de ***Analysis neighborhoods*** y ***Current Police Districts*** no comprenden todos los polígonos de San Francisco definidos en el conjunto de datos **Bay Area Countries**. Hay zonas de este último archivo que no están asignadas a ningún distrito policial ni a ningún vecindario. Durante la recolección de datos hubo contacto con el Departamento de Planificación Territorial de San Francisco y el responsable nos recomendó utilizar el archivo ***Bay Area Countries*** para determinar si un punto geográfico se encontraba en San Francisco ya que los conjuntos de datos relacionados con los vecindarios y los distritos policiales no contenían algunas zonas del condado de San Francisco, por lo tanto, los incidentes que tienen valores faltantes en estas tres variables pertenecen a un área de San Francisco que no está tiene asignado ningún distrito policial ni vecindario.

Después de investigar las causas de los valores ausentes, se realizó una imputación de datos para abordar esta situación. En el caso de la variable *neighborhood*, los valores ausentes se imputaron con la categoría **"*Periphery Zone*"**. Para la variable *Street*, se asignó el valor **"*Unknown Street*"** a los datos faltantes. Por último, en la variable *Police_district*, se utilizó la etiqueta **"*Outside District Boundaries*"** para los registros sin información.

❖ ***Creación y codificación de variables***

En esta parte del preprocesado tuvimos como objetivo tener todas las variables codificadas ya que la mayoría de nuestras variables eran de tipo categórica.

A continuación, podemos observar las columnas del conjunto de datos principal sobre el que se aplicaran estas codificaciones.

| **Columna** | **Descripción** |
|---|---|
| *incident_datetime* | Fecha del incidente en formato yyyy-mm-dd hh:mm:ss |
| *incidente_day_of_week* | Día de la semana en el que ocurrió el incidente |
| *incident_code* | Código del incidente |
| *incident_category* | Categoría del incidente |
| *incident_subcategory* | Subcategoría del incidente |
| *resolution* | Resolución del incidente |
| *latitude* | Latitud geográfica del incidente |
| *longitude* | Longitud geográfica del incidente |
| *neighborhood* | Vecindario en el que tuvo lugar el incidente |
| *Police_district* | Distrito Policial en el que ocurrió el incidente |
| *Holiday* | Variable binaria que indica si el día era festivo o no |
| *Street* | Calle en la que ocurrió el incidente. |

**Figura 47. Variables del conjunto de datos resultante tras calcular las variables**

En la Figura 47, tenemos todas las variables iniciales, dentro de estas variables, las únicas variables numéricas eran *incident_code, latitude* y *longitude*.

Para aplicar los modelos estadísticos en la etapa de Modelado hemos de tener también las variables en formato numérico, por lo que para las *incident_day_of_week*, *incident_category*, *incident_subcategory*, *resolution*, *neighborhood*, *street*, *holiday* y *police_district* hemos de aplicar una codificación de etiquetas o *Label Encoding*.

Con el objetivo de capturar más información y relaciones en nuestro conjunto de datos, se realizó un análisis de la variable *incident_datetime*, la cual estaba en formato "yyyy-mm-dd hh:mm:ss". Para facilitar su inclusión en los modelos estadísticos, se procedió a crear nuevas variables a partir de esta información.

En primer lugar, se generaron las variables *year*, *month*, *day*, *hour* y *minutes* a partir de la variable original *incident_datetime*. Estas variables permiten descomponer la fecha en sus componentes individuales, lo que facilita el análisis y modelado de los datos en función de la fecha.

Además, se crearon las variables *interval_hour* e *interval_minutes* para representar el intervalo en el que ocurrió el incidente. Para la variable de la hora, se definieron 6 intervalos de 4 horas cada uno, lo que permite agrupar los incidentes en períodos específicos del día. Para los minutos, se establecieron 4 intervalos de 15 minutos cada uno, lo que ayuda a identificar patrones temporales más precisos.

Mediante esta descomposición de la variable original *incident_datetime* y la creación de las variables de intervalo, se amplió la información disponible y se facilitó el análisis de los datos en función del tiempo.

Adicionalmente, se generaron otras variables relevantes para el análisis estadístico, tales como *week*, *quarter* y *season*, las cuales proporcionan información adicional sobre la temporalidad de los incidentes.

La variable *week* indica la semana en la que ocurrió cada incidente, permitiendo detectar posibles patrones o variaciones semanales en los datos. Por su parte, la variable *quarter* indica el cuatrimestre al que pertenece cada incidente, lo cual brinda una visión más amplia y estructurada del tiempo en relación con los eventos.

Además, se calculó la variable *season* que identifica la estación del año en la que ocurrió cada incidente. Esta información es relevante para comprender posibles relaciones entre la temporalidad y la ocurrencia de eventos, considerando las características propias de cada estación.

Tras el cálculo de estas variables obtuvimos un conjunto de datos con las siguientes variables.

| **Columna** | **Descripción** |
|---|---|
| *incident_datetime* | Fecha del incidente en formato yyyy-mm-dd hh:mm:ss |
| *incidente_day_of_week* | Día de la semana en el que ocurrió el incidente |
| *incident_code* | Código del incidente |
| *incident_category* | Categoría del incidente |

| | |
|---|---|
| *incident_subcategory* | Subcategoría del incidente |
| *resolution* | Resolución del incidente |
| *latitude* | Latitud geográfica del incidente |
| *longitude* | Longitud geográfica del incidente |
| *neighborhood* | Vecindario en el que tuvo lugar el incidente |
| *police_district* | Distrito Policial en el que ocurrió el incidente |
| *holiday* | Variable binaria que indica si el día era festivo o no |
| *street* | Calle en la que ocurrió el incidente. |
| *day* | Número del día del mes del incidente |
| *month* | Mes del incidente |
| *year* | Año del incidente |
| *hour* | Hora del incidente |
| *minutes* | Minuto en el que ocurrió el incidente |
| *week* | Semana del mes en la que ocurrió el incidente |
| *quarter* | Cuatrimestre en el que ocurrió el incidente |
| *season* | Estación del año en la que ocurrió el incidente |
| *interval_hour* | Intervalo horario en horas en el que tuvo lugar el incidente |
| *interval_minutes* | Intervalo horario en minutos en el que ocurrió el incidente |

**Figura 48. Variables del conjunto de datos resultante tras calcular las variables temporales**

Una vez alcanzado este punto, se procedió a realizar la codificación de las variables, teniendo en cuenta que la variable *incident_datetime* fue desglosada en múltiples variables y no se incluirá en el proceso de codificación.

Las variables *latitude* y *longitude* no requieren codificación, ya que son variables numéricas. Del mismo modo, las variables *year*, *month*, *day*, *hour* y *minutes* tampoco necesitan ser codificadas, ya que representan valores numéricos que mantienen su naturaleza original.

Para llevar a cabo la codificación de las variables categóricas, se creó una función que asignaba a cada valor único de la variable un número específico y generaba un diccionario. Este diccionario se construía con el valor codificado como clave y el valor original de la variable como valor asociado.

De esta manera, se logró transformar las variables categóricas en una representación numérica que facilita su análisis y procesamiento posterior en los modelos estadísticos utilizados. El código empleado en esta parte del proyecto se encuentra publicado en el archivo Preprocess_SF_part2 del *GitHub* creado para este proyecto.

- ***Clustering***

En este trabajo se persigue analizar y predecir la ubicación y la fecha de incidentes en la ciudad de San Francisco. Con el fin de mejorar la precisión de los modelos predictivos, se aplicará una técnica de *clustering* para identificar grupos de incidentes similares. A partir de estos clusters, se desarrollarán modelos estadísticos específicos para cada uno de ellos. Este enfoque permitirá reducir la variabilidad de las variables de respuesta y mejorar la precisión de las predicciones.

Para la detección de estos grupos dentro de nuestro conjunto de datos, se seleccionó una muestra para aplicar sobre ella un *clustering*. El motivo por el cual se seleccionó esta muestra, fue para reducir la complejidad computacional y temporal del algoritmo de *clustering* en el conjunto completo de datos. Al aplicar el *clustering* a una muestra más pequeña, podemos obtener resultados rápidos y exploratorios que nos permitirán comprender la estructura y distribución de los grupos en el conjunto de datos general.

A continuación, se encuentran detallados todos los pasos seguidos para la aplicación del *clustering*.

- **Selección de muestras**

Con el objetivo de manejar eficientemente la gran cantidad de datos en el conjunto principal, se optó por realizar un muestreo estratificado de 30.000 registros. Este muestreo se llevó a cabo de manera que se respetarán las proporciones de todas y cada una de las variables presente en el conjunto principal. Al reducir la cantidad de datos, podremos aplicar el algoritmo de clustering a esta muestra y posteriormente extrapolar los resultados obtenidos al conjunto de datos principal. El código empleado para la selección de esta muestra se encuentra en el archivo Muestreo publicado en el *GitHub* creado para este proyecto.

➢ **Preparación de las variables**

Una vez seleccionada la muestra base para el clustering procedimos a preparar las variables con el objetivo de que éstas estén en un formato adecuado y sean comparables entre sí antes de aplicar el *clustering*.

Las variables principales del conjunto de datos empleados tras el preprocesado son:

`'incident_datetime', 'incident_day_of_week', 'incident_code', 'incident_category', 'incident_subcategory', 'resolution', 'latitude', 'longitude', 'neighborhood', 'Police_district', 'Holiday', 'Street', 'day', 'month', 'year', 'hour', 'minutes', 'week', 'quarter', 'season', 'interval_hour', 'interval_minutes', 'incident_day_of_week_cod', 'incident_category_cod', 'incident_subcategory_cod', 'resolution_cod', 'neighborhood_cod', 'Police_district_cod', 'Holiday_cod', 'Street_cod', 'week_cod', 'quarter_cod', 'season_cod', 'interval_hour_cod', 'interval_minutes_cod'.`

Las variables que se emplearán para el *clustering* serán las variables codificadas, excluyendo las variables: `'latitude', 'longitude';` debido a que son de naturaleza numérica continua,`'incident_datetime'`; por el formato y ya que se han extraído las variables `'day', 'month', 'year', 'hour', 'minutes'`. Por otro lado también se excluyeron las variables `'day', 'month', 'year', 'hour', 'minutes'` y en su lugar teníamos la información de estas variables en las variables calculadas `'week', 'quarter', 'season', 'interval_hour', 'interval_minutes'` que fueron codificadas.

Las variables empleadas en el *clustering* fueron: `'incident_day_of_week_cod', 'incident_category_cod', 'incident_subcategory_cod', 'resolution_cod', 'neighborhood_cod', 'Police_district_cod', 'Holiday_cod', 'Street_cod', 'week_cod', 'quarter_cod', 'season_cod', 'interval_hour_cod', 'interval_minutes_cod'.`

Tras haber seleccionado las variables a emplear se procedió con la preparación de estas variables. En esta parte, se tuvo en cuenta los valores faltantes de las variables y las escalas diferentes. Como en el conjunto de datos empleado para la selección de muestra no presentaba valores faltantes, no fue necesario ningún tratamiento. Sin embargo, para estudiar las escalas de las diferentes variables que tenemos se analizaron los máximos y mínimos de las variables.

| Variable | Valor mínimo | Valor máximo |
|---|---|---|
| *incident_day_of_week_cod* | 0 | 6 |
| *incident_code* | 1000 | 75030 |
| *incident_category_cod* | 0 | 47 |

| | | |
|---|---|---|
| *incident_subcategory_cod* | 0 | 66 |
| *resolution_cod* | 0 | 3 |
| *neighborhood_cod* | 0 | 41 |
| *Police_district_cod* | 0 | 10 |
| *Street_cod* | 0 | 1331 |
| *Holiday_cod* | 0 | 1 |
| *week_cod* | 0 | 3 |
| *season_cod* | 0 | 3 |
| *quarter_cod* | 0 | 3 |
| *interval_hour_cod* | 0 | 5 |
| *interval_minutes_cod* | 0 | 3 |

**Figura 49. Intervalos de las variables codificadas**

Tras analizar los rangos, las variables difieren bastante en cuanto a escala. Para la variable *incident_code* teníamos un intervalo de 1000 a 75030, mientras que para la variable *Holiday_cod* se distribuye en el intervalo 0 a 1. Por este motivo se procedió a estandarizar las variables codificadas. Estas nuevas variables fueron renombradas con el nombre de la variable con la adición de la coletilla "_sca". Finalmente, las variables de nuestro conjunto de datos para aplicar el *clustering* fueron:

```
'incidentCode_sca',                    'incident_day_of_week_sca',
'incident_category_sca',               'incident_subcategory_sca',
'resolution_sca',   'neighborhood_sca',   'Police_district_sca',
'Holiday_sca',    'Street_sca',    'week_sca',    'quarter_sca',
'season_sca',    'interval_hour_sca',    'interval_minutes_sca'.
```

➢ **Estudio de relaciones entre variables**

Después de haber seleccionado las variables de entrada al modelo *clustering*, se realizó un estudio de las relaciones existentes entre las variables con el objetivo de identificar relaciones significativas. Para hacer este estudio se calculó la matriz de correlaciones para las variables.

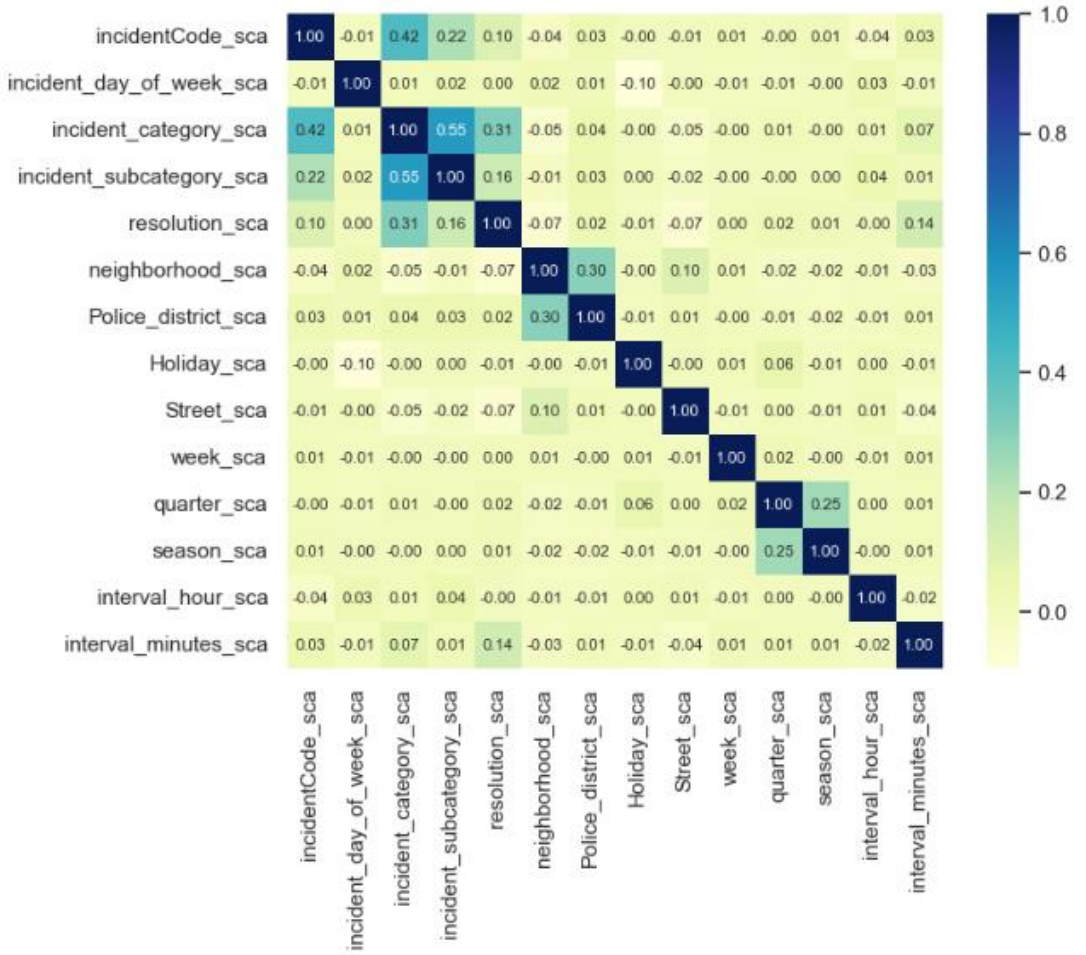


**Figura 50. Matriz de correlaciones entre variables escaladas**

En la figura anterior, se puede observar que los mayores coeficientes, ordenados de mayor a menor se encuentran entre los pares de variables *incident_subcategory_sca-incident_category_sca*, *incident_category_sca-incidentCode_sca,resolution_sca-incident_category_sca,neighborhood_sca-Police_district_sca,quarter_sca- season_sca* y *incident_subcategory_sca - incidentCode_sca*.

En conclusión, se han encontrado relaciones entre algunos pares de variables, pero el coeficiente de correlación no es muy alto.

### ➢ Elección de número de clusters y distancia

Tras completar el estudio de agrupamiento de los datos, se llevó a cabo un análisis para identificar la distancia óptima y el número adecuado de *clusters* para aplicar el *clustering*. Esto se hizo considerando los diferentes algoritmos disponibles según la naturaleza de los datos, que eran categóricos y estaban codificados.

Dado que los datos eran categóricos y estaban codificados, se consideraron dos opciones de algoritmos: KModes y *clustering* jerárquico aglomerativo. Una vez seleccionados los algoritmos aplicables a los datos del estudio, se procedió a examinar las distintas medidas de distancia que se podían utilizar. Debido a la

naturaleza de los datos, se requería una medida de similitud para poder comparar los diferentes registros en el conjunto de datos y determinar las agrupaciones existentes.

Para ello, se tendría que aplicar distancias como la de *hamming, gower* o *jaccard* y explorar el número de clusters para agrupar los datos.

Con el propósito de obtener un *clustering* de calidad se creó código *Python* mediante el cual se calcularon el coeficiente de silhouette, el índice de calinski y la inercia del método elbow para el intervalo k de 2 a 20 clusters teniendo en cuenta las distancias *hamming, gower* o *jaccard* aplicando el algoritmo de *KMeans*.

En la Figura 51 se observan los coeficientes obtenidos para cada número de *clusters* y por distancia de similitud.

| Distance | Clusters | Silhouette | Elbow | Calinski |
|---|---|---|---|---|
| gower | 2 | 0,05107600 | 378782,57 | 722,635783 |
| hamming | 3 | 0,04276500 | 350172,08 | 550,074761 |
| hamming | 2 | 0,03763100 | 378782,09 | 660,45327 |
| hamming | 5 | 0,03381600 | 314349,56 | 534,423527 |
| hamming | 4 | 0,03338600 | 332346,82 | 476,227633 |
| hamming | 7 | 0,03142400 | 288264,49 | 370,831648 |
| hamming | 8 | 0,02850500 | 280085,55 | 301,777588 |
| hamming | 14 | 0,02820900 | 250146,42 | 269,521666 |
| hamming | 6 | 0,02709700 | 300305,60 | 428,753002 |
| hamming | 9 | 0,02502100 | 273388,20 | 302,028604 |
| hamming | 18 | 0,02097300 | 237828,35 | 214,443414 |
| hamming | 13 | 0,02053400 | 253541,55 | 278,720511 |
| hamming | 11 | 0,02037800 | 261827,37 | 278,769543 |

| | | | | |
|---|---|---|---|---|
| hamming | 12 | 0,02009400 | 257219,79 | 301,349628 |
| hamming | 10 | 0,01946400 | 266866,99 | 349,630634 |
| hamming | 17 | 0,01903300 | 239790,31 | 233,505626 |
| hamming | 16 | 0,01872800 | 243381,03 | 204,339893 |
| hamming | 19 | 0,01854100 | 235177,16 | 211,713121 |
| gower | 3 | 0,01513000 | 357377,30 | 531,357689 |
| hamming | 15 | 0,01480900 | 245771,70 | 216,202411 |
| hamming | 20 | 0,01385100 | 232785,25 | 172,66859 |
| gower | 5 | 0,00854000 | 314346,26 | 418,785847 |
| gower | 4 | 0,00762000 | 331535,49 | 490,030907 |
| jaccard | 2 | 0,00000000 | 378782,15 | 354,052823 |
| jaccard | 3 | 0,00000000 | 357377,32 | 502,111556 |
| jaccard | 4 | 0,00000000 | 334212,49 | 490,511007 |
| jaccard | 5 | 0,00000000 | 314346,71 | 511,987768 |
| jaccard | 6 | 0,00000000 | 300882,06 | 365,676176 |
| jaccard | 7 | 0,00000000 | 290505,65 | 425,690937 |
| jaccard | 8 | 0,00000000 | 279519,91 | 251,419807 |
| jaccard | 9 | 0,00000000 | 273125,82 | 312,943719 |
| jaccard | 10 | 0,00000000 | 267020,76 | 335,704126 |
| jaccard | 11 | 0,00000000 | 261834,12 | 321,177379 |

| | | | | |
|---|---|---|---|---|
| jaccard | 12 | 0,00000000 | 257275,60 | 276,176464 |
| jaccard | 13 | 0,00000000 | 253324,11 | 301,809044 |
| jaccard | 14 | 0,00000000 | 250025,52 | 251,425383 |
| jaccard | 15 | 0,00000000 | 247116,58 | 246,200544 |
| jaccard | 16 | 0,00000000 | 243327,98 | 193,357864 |
| jaccard | 17 | 0,00000000 | 240518,79 | 249,420301 |
| jaccard | 18 | 0,00000000 | 237839,08 | 208,633128 |
| jaccard | 19 | 0,00000000 | 235318,23 | 206,256684 |
| jaccard | 20 | 0,00000000 | 233214,74 | 208,730331 |
| gower | 6 | -0,01284100 | 302439,27 | 325,068337 |
| gower | 7 | -0,01899500 | 288263,97 | 425,16402 |
| gower | 8 | -0,02434600 | 279822,50 | 355,61463 |
| gower | 10 | -0,02807400 | 268131,71 | 357,125621 |
| gower | 9 | -0,03049500 | 271721,96 | 359,154381 |
| gower | 12 | -0,03736600 | 256997,03 | 250,627094 |
| gower | 11 | -0,03783700 | 261466,00 | 275,333986 |
| gower | 16 | -0,03895100 | 243580,34 | 240,487462 |
| gower | 13 | -0,03989800 | 253378,98 | 230,800409 |
| gower | 14 | -0,04105100 | 250418,53 | 249,255407 |
| gower | 15 | -0,04332100 | 247007,35 | 224,799616 |

| | | | | |
|---|---|---|---|---|
| gower | 17 | -0,04821800 | 239982,70 | 243,447946 |
| gower | 18 | -0,04841100 | 237838,85 | 232,498019 |
| gower | 20 | -0,04935700 | 232710,61 | 194,348276 |
| gower | 19 | -0,05366400 | 234884,79 | 207,981557 |

**Figura 51. Coeficientes de evaluación con k de 1 a 20 y distancias *hamming*, *gower* y *jacquard* aplicando *KModes*.**

Tras realizar el estudio de agrupamiento, se evaluaron distintas distancias y números de clusters para determinar la calidad de los resultados obtenidos. A continuación, se presenta un resumen de los coeficientes obtenidos:

La distancia *Hamming* mostró coeficientes de silhouette más altos en comparación con la distancia *Gower*. Para la distancia *Hamming*, se obtuvo un coeficiente de *Silhouette* máximo de 0.05107600 con 2 *clusters*, seguido de 0.04276500 con 3 *clusters*. Estos valores indican una buena separación y cohesión de los grupos.

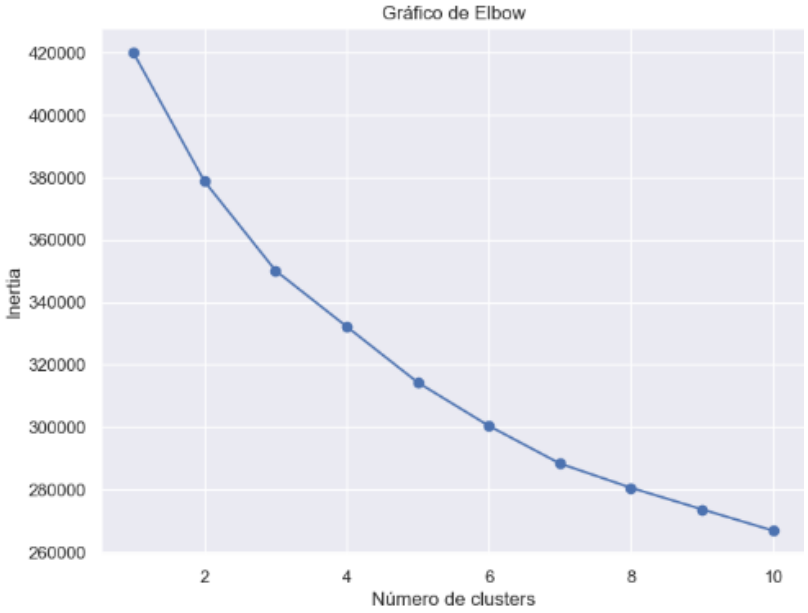


**Figura 52. Gráfico método *Elbow* para la elección del número óptimo de clusters**

En cuanto a los coeficientes de inercia, Figura 51 y 52, se encontró que el valor más bajo fue de 378782.57 con 2 *clusters*, lo cual sugiere que este número de *clusters* proporciona una buena representación de los datos.

El índice de Calinski mostró un valor máximo de 722.635783 con 2 *clusters*, lo que indica una alta densidad y separación entre los grupos.

En el caso de la distancia *Gower*, el coeficiente de *silhouette* más alto obtenido fue de 0.01513000 con 3 *clusters*. Aunque este valor es más bajo en comparación con la distancia *Hamming*, sigue indicando una separación aceptable entre los grupos.

En resumen, los resultados sugieren que la distancia *Hamming* con 2 o 3 *clusters* podría ser la mejor opción para el agrupamiento de los datos categóricos analizados. Estos valores de coeficientes proporcionan una medida de la calidad de los agrupamientos y ofrecen información valiosa sobre la estructura de los datos y las agrupaciones existentes.

Tras haber explorado los coeficientes de evaluación aplicando el algoritmo de *KModes*, se exploró también la aplicación de un clustering jerárquico aglomerativo basado en el método *ward* y en el método *complete*.

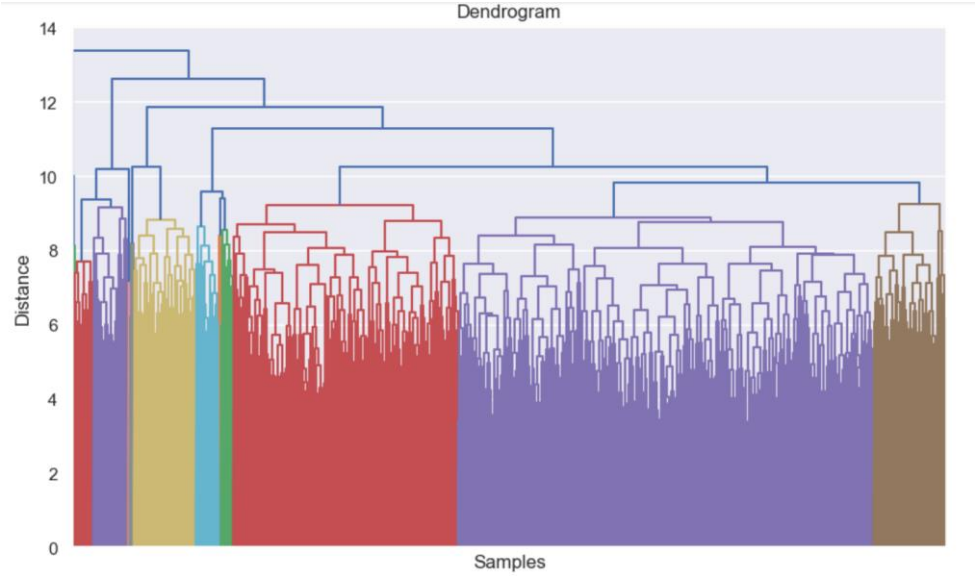


**Figura 53. Dendograma obtenido con clustering jerárquico aglomerativo, *method* = *'complete'*.**

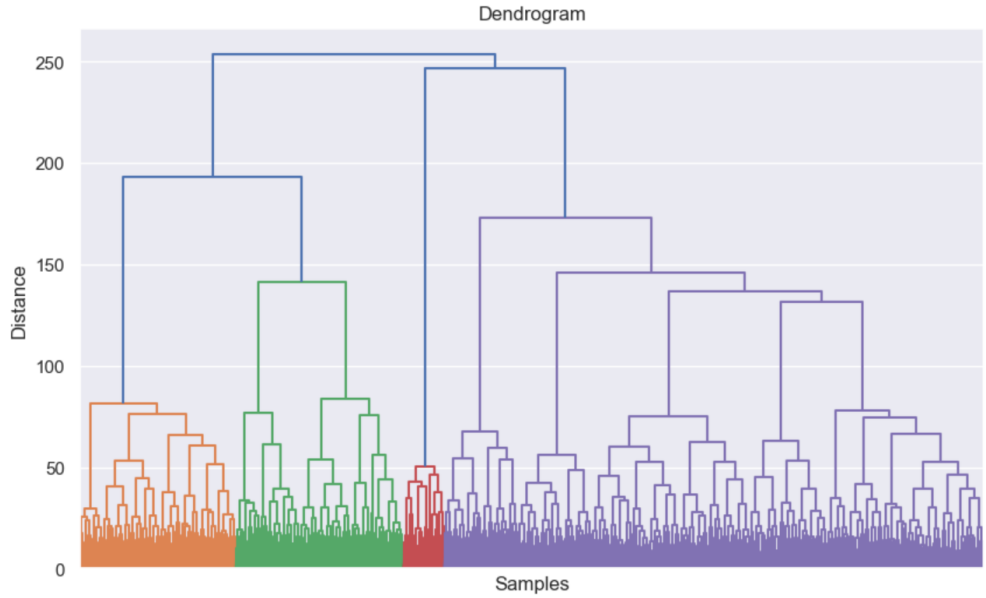


**Figura 54. Dendograma obtenido con *clustering* jerárquico aglomerativo, *method* = *'ward'*.**

En la Figura 53 se muestra el dendrograma generado mediante la aplicación del *clustering* jerárquico utilizando el método completo. En este dendrograma, se puede observar claramente la separación de los datos en 8 *clusters* distintos.

En la Figura 54 se presenta el dendrograma obtenido mediante el uso del clustering jerárquico con el método de *Ward*. En este caso, se identifican 4 *clusters* bien definidos. El código empleado para la aplicación del *clustering* se encuentra en el archivo Clustering_SF publicado en el *GitHub* creado para este proyecto.

La decisión de aplicar el clustering jerárquico se fundamenta en la claridad y la interpretación visual que proporcionan los dendrogramas. En contraste, al utilizar el algoritmo *KModes*, los coeficientes de Silhouette obtenidos fueron muy bajos, lo que indica una agrupación deficiente.

En la siguiente etapa de Evaluación, se explicarán las decisiones que se tomaron a partir de los resultados obtenidos en esta etapa de Modelado del *clustering*.

Dado que los coeficientes de evaluación obtenidos para ambos valores de k, indican que los grupos no están bien definidos y existe superposición entre ellos, se ha decidido aplicar los modelos estadísticos por distrito policial. La división organizativa por distrito policial proporciona una estructura clara y definida, lo que puede ayudar a mejorar la interpretación y la aplicabilidad de los modelos estadísticos. Al considerar la división por distrito policial, podemos analizar y comprender mejor los patrones y las características específicas de cada área, lo que puede ser más útil para la toma de decisiones y la implementación de medidas específicas en cada distrito.

En una primera instancia se dividió el conjunto de datos por cada distrito policial y para cada distrito policial se desarrollaron los modelos estadísticos correspondientes a las redes neuronales MLP y a las redes neuronales Keras Regressor..

## Fase 4. Modelado.

El objetivo principal de este proyecto de investigación es utilizar redes neuronales recurrentes y redes neuronales tipo Transformer para predecir el tiempo y lugar de futuros incidentes en San Francisco. Para lograr esto, se llevó a cabo un estudio exhaustivo de los diferentes tipos de redes neuronales disponibles, evaluando cuál se ajustaba mejor a la naturaleza de nuestras variables. Además, se realizó un ajuste de hiperparámetros para localizar la combinación óptima de parámetros que maximiza la precisión de los modelos.

En el proceso de selección de modelos, se consideró la inclusión de mecanismos de atención en uno de los modelos y la exclusión de estos en otro. Esto se hizo con el propósito de investigar si la incorporación de la atención mejoraba la precisión de los modelos en comparación con aquellos que no la incluían.

Se optó por utilizar las siguientes redes neuronales:

- ***Regression Multilayer Perceptron (MLP)*,** ya que es un enfoque basado en redes neuronales para realizar regresión, donde se utilizan múltiples capas y funciones de activación no lineales para predecir valores numéricos continuos a partir de características de entrada. En este caso, son ideales ya que la intención fue predecir las variables: *latitude, longitude, day, month, year, hour* y *minutes*, todas ellas de tipo numérico.

- ***Keras Regressor*,** La arquitectura *Keras* tiene ventajas clave en la tarea de regresión. Permite aprender características relevantes, manejar tanto datos categóricos como numéricos, y capturar relaciones complejas en los datos de entrada. Asimismo también permite añadir mecanismos de atención sobre el modelo.
  Al aprovechar las capacidades de aprendizaje y el manejo versátil de datos que ofrece *Keras Regressor*, es posible obtener resultados precisos y eficientes en la generación de predicciones numéricas en nuestro contexto de regresión.
  Además, la biblioteca *Keras* ofrece una interfaz intuitiva y flexible para diseñar, entrenar y evaluar modelos de regresión. Su amplia gama de capas, activaciones y optimizadores permite adaptar el modelo a las necesidades específicas del problema de regresión. Con la combinación de las ventajas de la arquitectura *Keras* y la flexibilidad de *Keras Regressor*, podemos aprovechar al máximo el potencial de las redes neuronales en la tarea de regresión. Esto nos permite obtener resultados precisos y eficientes, y facilita el desarrollo de modelos robustos y adaptables a diferentes escenarios de regresión.

Antes de implementar el código necesario para aplicar estos modelos es necesario definir las variables de entrada y salida del mismo.

A partir de las variables resultantes de la Fase 3 de preprocesado y el objetivo planteado, las variables se clasificaron de la siguiente manera:

- **Variables de salida:** *latitude*, *longitude*, *day*, *month*, *year*, *hour* y *minutes*.
- **Variables de entrada:** *incident_category*, *incident_subcategory*, *incident_code*, *neighborhood*, *season*, *week*, *quarter*, *street*, *interval_hour*, *interval_minutes*.

Tras clasificar las variables, surgió la duda de si al intentar predecir las variables *day*, *month*, *year*, *hour* y *minutes* a partir de las variables *season*, *week*, *quarter*, *street* se introduciría multicolinealidad al modelo ya que ambas, tanto las temporales de salida como las de entrada, estaban calculadas a partir de la variable inicial *incident_datetime*.

Para verificar que no existe multicolinealidad entre las variables predictoras temporales y las variables objetivo temporal, se realizó un estudio de las relaciones existentes entre ambas.

En la Figura 55, se muestra un gráfico de correlaciones para todas las variables base de los modelos estadísticos a desarrollar.

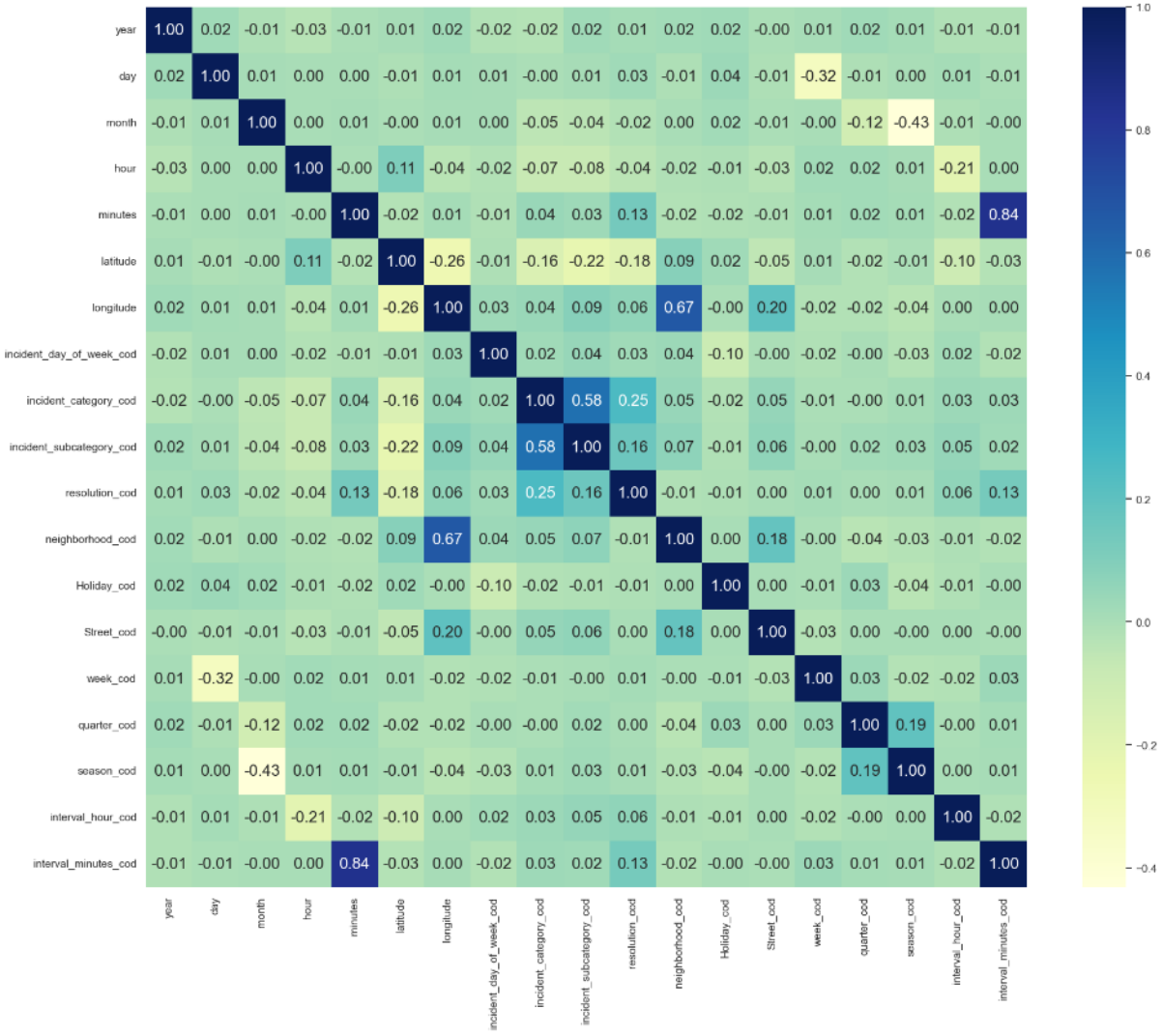


**Figura 55. Matriz de correlaciones para las variables de entrada y salida**

Podemos observar que entre las variables predictoras, la variable *interval_minutes* tiene una correlación positiva con la variable de salida, minutes. Por ello se procedió a descartar la variable *interval_minutes* de las variables de entrada al modelo con la intención de eliminar cualquier posible causante de una multicolinealidad, y así los modelos desarrollados podrán capturar patrones relacionados con la temporalidad de los datos.

Tras esta validación, las variables definitivas a emplear en el modelo fueron:

- **Variables de salida:** *latitude*, *longitude*, *day*, *month*, *year*, *hour* y *minutes*.
- **Variables de entrada:** *incident_category*, *incident_subcategory*, *incident_code*, *neighborhood*, *season*, *week*, *quarter*, *street*, *interval_hour*.

En lo que respecta a la implementación del código, se tuvieron que desarrollar dos propuestas, debido a que la primera de ellas falló por motivos computacionales, y se tuvo que optar por la segunda propuesta como alternativa. A continuación se detallan brevemente las características de cada una de estas propuestas:

- **Propuesta 1.** Entrenamiento de modelos por distrito policial
  Se entrenaron modelos MLP y modelos *Keras Regressor* utilizando *GridSearch* y *Pipeline* para ajustar los hiperparámetros en cada distrito policial. Luego, se generaron representaciones gráficas de los resultados para cada distrito policial, mostrando el mejor modelo MLP desarrollado y el mejor modelo *Keras Regressor* entrenado. A partir de este código, se generaron dos archivos CSV por cada distrito policial: uno con los resultados de todas las combinaciones de parámetros para los modelos MLP entrenados, y otro con los resultados de las combinaciones de parámetros para las redes neuronales utilizando *Keras Regressor*.

  Finalmente, se presentaban los resultados de los modelos finales seleccionados, incluyendo métricas como R2, MAE y MSE, así como gráficos de dispersión que comparan las predicciones con los valores reales, un histograma de errores y un gráfico de línea que mostraba el porcentaje acumulado de error.

  Para acelerar el tiempo de ejecución, se utilizaron técnicas de paralelización para aprovechar todos los recursos disponibles. Sin embargo, debido a limitaciones de memoria, se optó por la Propuesta 2, que está basada en un distrito policial y aplicando tamaños de batch más pequeños como parámetro.
  El código empleado para la aplicación de esta propuesta se encuentra publicado en el *GitHub* creado para este proyecto.

- **Propuesta 2.** Esta propuesta fue implementada como alternativa a los problemas de ejecución por falta de memoria de procesamiento. Este código filtraba el conjunto de datos sobre el que se aplicó el *clustering* para un distrito policial, *CENTRAL*, ya que era el que más datos tenía, y para los años 2021 y 2022.
  Sobre este conjunto de datos se aplicaban los modelos estadísticos, primero se seleccionaba el mejor modelo entrenado resultado de aplicar un ajuste de

hiperparámetros al modelo MLP y luego se entraron los modelos *Keras Regressor* realizando también un ajuste de hiperparámetros para seleccionar el mejor de ellos.
Finalmente se obtuvieron dos modelos, el mejor modelo MLP y el mejor modelo Keras Regressor. Posteriormente, para cada uno de estos modelos se mostraban las métricas obtenidas resultantes del entrenamiento y las métricas de los modelos finales. También se lograban visualizaciones con gráficos de dispersión entre las predicciones y los valores reales, histograma de errores y gráfico de línea del porcentaje de error acumulativo. El código empleado para la aplicación de esta propuesta se encuentra publicado en el *GitHub* creado para este proyecto.

Tras haber definido las variables a emplear y haber presentado una visión general del código implementado, ahora nos adentraremos en los detalles sobre la aplicación de cada uno de los modelos utilizados.

- **Regression Multilayer Perceptron (MLP)**

  Para la aplicación de redes neuronales MLP, primero se procedió a dividir el conjunto de datos de los incidentes correspondientes al distrito policial, *CENTRAL* de los años 2021 y 2022.
  El conjunto de datos sobre el que se entrenó el modelo, consta originalmente de 4052 registros. A partir de este conjunto de datos, se realizó una validación cruzada dividiendo el conjunto en un 80% para el conjunto de entrenamiento y un 20% para el conjunto de validación. De esta manera, se garantiza una adecuada distribución de los datos y se proporciona una base sólida para evaluar y comparar el rendimiento de los modelos.

  Tras seleccionar el conjunto de entrenamiento y validación, se continuó preparando las variables que se iban a emplear. Con el propósito de mejorar el rendimiento de los modelos que fueron diseñados, se aplicó un escalado a las variables objetivo para que estas variables estén en la misma escala y así facilitar al modelo la interpretación y el análisis de la relación espacial de los incidentes.

  Este escalado se realizó por medio de la clase *Transformed Target Regressor* de la biblioteca *scikit-learn* para normalizar las variables objetivo en el momento de ingresar al modelo. Esta clase permite aplicar transformaciones a las variables objetivo, lo cual resulta útil para mejorar el rendimiento del modelo. Además, al mostrar los resultados, la clase realiza la función inversa de la transformación para presentar los valores de las variables objetivo en su rango original. De esta manera, se logra una interpretación más adecuada de los resultados del modelo.

  Por otro lado, como las variables de entrada o predictoras se distribuían entre rangos bastante diferentes, tal y como se visualizó en el *clustering* realizado, se aplicó un escalado de las variables antes de introducirlas en el modelo.

  Una vez preparadas las variables de entrada y salida para los conjuntos de entrenamiento y validación, se definieron los hiperparámetros con los que se crearían diversos modelos mediante el uso de *Grid Search*, se aplicó

paralelización computacional y se generaron todos los modelos resultantes de la combinación de los hiperparámetros definidos. Los parámetros empleados fueron los siguientes:

- **Tamaño de capas ocultas (*Hidden Layer Sizes*):** (64,), (64, 64), (64, 64, 64), (64, 64, 64, 64). Estos valores indican que se van a hacer combinaciones con 1, 2, 3 y 4 capas y en cada una de las capas estará definida por 64 neuronas.
- **Tasa de aprendizaje inicial (*Learning Rate Init*):** 0.0001, 0.001, 0.01, 0.1. Esta tasa indica qué tan rápido el modelo ajusta los pesos durante el proceso de entrenamiento.
- **Número de iteraciones máximas (*Max iter*):** 10, 15, 20, 25, 50, 100. Este número limitará el número de veces que el modelo ajustará los pesos durante el entrenamiento.
- **Tamaño de lote (*Batch Size*):** 2, 4, 6, 8, 10.

Tras definir los parámetros y emplear *Grid Search* para entrenar los modelos y seleccionar el mejor en base a el coeficiente de determinación, se almacenaron los resultados de los modelos resultantes de todas las combinaciones del ajuste de hiperparámetros en un archivo csv para poder realizar tablas en *Excel* y publicarlo en el proyecto de GitHub creado.

- **Redes Neuronales *Keras Regressor***

Para la aplicación de redes neuronales MLP, primero se procedió a dividir el conjunto de datos de los incidentes correspondientes al distrito policial, CENTRAL de los años 2021 y 2022.
El conjunto de datos sobre el que se entrenó el modelo, consta originalmente de 4052 registros A partir de este conjunto de datos, se realizó una validación cruzada dividiendo el conjunto de datos en un 80% para el conjunto de entrenamiento y un 20% para el conjunto de validación. De esta manera, se garantiza una adecuada distribución de los datos y se proporciona una base sólida para evaluar y comparar el rendimiento de los modelos.

Tras seleccionar el conjunto de entrenamiento y validación, se continuó preparando las variables que se iban a emplear. Con el propósito de mejorar el rendimiento de los modelos que fueron diseñados, se aplicó un escalado a las variables objetivo para que estas variables estén en la misma escala y así facilitar al modelo la interpretación y el análisis de la relación espacial de los incidentes. Esto se realizó antes de entrenar el modelo y posteriormente se aplicó la transformación inversa por medio de la función *inverse_transform*.

Por otro lado, como las variables de entrada o predictoras se distribuían entre rangos bastante diferentes, tal y como se visualizó en el *clustering* realizado, se aplicó un escalado de las variables antes de introducirlas en el modelo.

Una vez preparadas las variables de entrada y salida para y los conjuntos de entrenamiento y validación se definieron los hiperparámetros con los que se

crearían diversos modelos mediante el uso de *Grid Search*, se aplicó paralelización computacional y se generaron todos los modelos resultantes de la combinación de los hiperparámetros definidos. En la implementación del código se creó el modelo añadiendo una capa de atención.

Los parámetros empleados fueron los siguientes:

- **Tamaño de capas ocultas (*Hidden Layer Sizes*):** (64,), (64, 64) (64,), (64, 64), (64, 64, 64), (64, 64, 64, 64). Estos valores indican que se van a hacer combinaciones con 1, 2, 3 y 4 capas y en cada una de las capas estará definida por 64 neuronas.
- **Tasa de aprendizaje inicial (*Learning Rate Init*):** 0.0001, 0.001, 0.01, 0.1. Esta tasa indica qué tan rápido el modelo ajusta los pesos durante el proceso de entrenamiento.
- **Número de iteraciones máximas (*Max iter*):** 10, 15, 20, 25, 50, 100. Este número limitará el número de veces que el modelo ajustará los pesos durante el entrenamiento.
- **Tamaño de lote (*Batch Size*):** 2, 4, 6, 8.

En este caso también se empleaba *Grid Search* para entrenar los modelos y seleccionar el mejor en función del coeficiente de determinación, asimismo se almacenaron los resultados de los modelos resultantes de todas las combinaciones resultantes del ajuste de hiperparámetros en un archivo para publicarlo posteriormente en GitHub.

## Fase 5. Evaluación.

Mediante la comparación de los resultados obtenidos por ambos modelos, se busca determinar cuál de ellos ofrece un mejor desempeño en la predicción de los incidentes en San Francisco. Esto permitirá evaluar las fortalezas y limitaciones de cada enfoque y obtener una visión más completa de su aplicabilidad en este contexto específico.

- **Regression Multilayer Perceptron (MLP)**

Tras haber definido los parámetros a ajustar y haber desarrollado todo el código necesario, se ejecutó el código y se obtuvo un modelo por cada combinación de parámetros. Para cada combinación se almacenaron los valores del MSE, MAE y R2 para poder comparar los resultados. A continuación se muestran las primeras 5 combinaciones con las que mejores resultados se obtienen. Estos resultados se encuentran publicados en el archivo *Resultados_MLP* del proyecto creado para el desarrollo de este trabajo en GitHub.

Posteriormente se mostraron los mejores parámetros con los que se obtuvo el mejor modelo, en el caso de la aplicación de MLP, el mejor modelo surgió a partir de la combinación de un tamaño de lote (*batch size*) de 6, número de capas ocultas (*hidden layer size*) de (64, 64, 64, 64), es decir 4 capas de 64 neuronas cada una, una tasa de aprendizaje (*learning rate init*) de 0.01 y con un máximo de iteraciones (*max iter*) de 100.

Para este modelo se obtuvo un R2 de 0.5368, un MSE de 52.1973 y un MAE de 3.1089. Esto indica que el modelo tiene una capacidad moderada para explicar la variabilidad de los datos, con un R2 de 0.5368. Sin embargo, los errores cuadráticos medios y absolutos medios, con valores de MSE de 52.1973 y MAE de 3.1089, respectivamente, sugieren que hay margen de mejora en la precisión de las predicciones del modelo.

También se calcularon estas métricas para cada variable de salida, en la siguiente figura se pueden observar los valores adquiridos.

| **Variable** | **R2** | **MAE** | **MSE** |
|---|---|---|---|
| *Day* | 0.84651 | 2.70108 | 11.75848 |
| *Month* | 0.93712 | 0.52819 | 0.40183 |
| *Year* | 0.0015 | 0.48355 | 0.27153 |
| *Hour* | 0.89541 | 1.52648 | 3.62459 |
| *Minutes* | 0.01140 | 16.5049 | 349.3464 |
| *Latitude* | 0.73532 | 0.00282 | 1.40135 |
| *Longitude* | 0.59881 | 0.003110 | 1.36318 |

**Figura 56. R2, MSE, MAE por variable de salida**

A partir de la figura anterior, los resultados muestran que el modelo tiene un buen ajuste para algunas variables (por ejemplo, *month, day y hour*) con valores altos de R2 y bajos

de MAE y MSE. Sin embargo, para otras variables (por ejemplo, *year*, *minutes* y *longitude*), el modelo tiene un ajuste deficiente, como se indica por los valores negativos de R2 y los altos valores de MAE y MSE.

Después de evaluar las métricas obtenidas se mostraron visualizaciones para cada una de las variables de salida con el propósito de evaluar el modelo de forma gráfica.

➢ ***DAY***

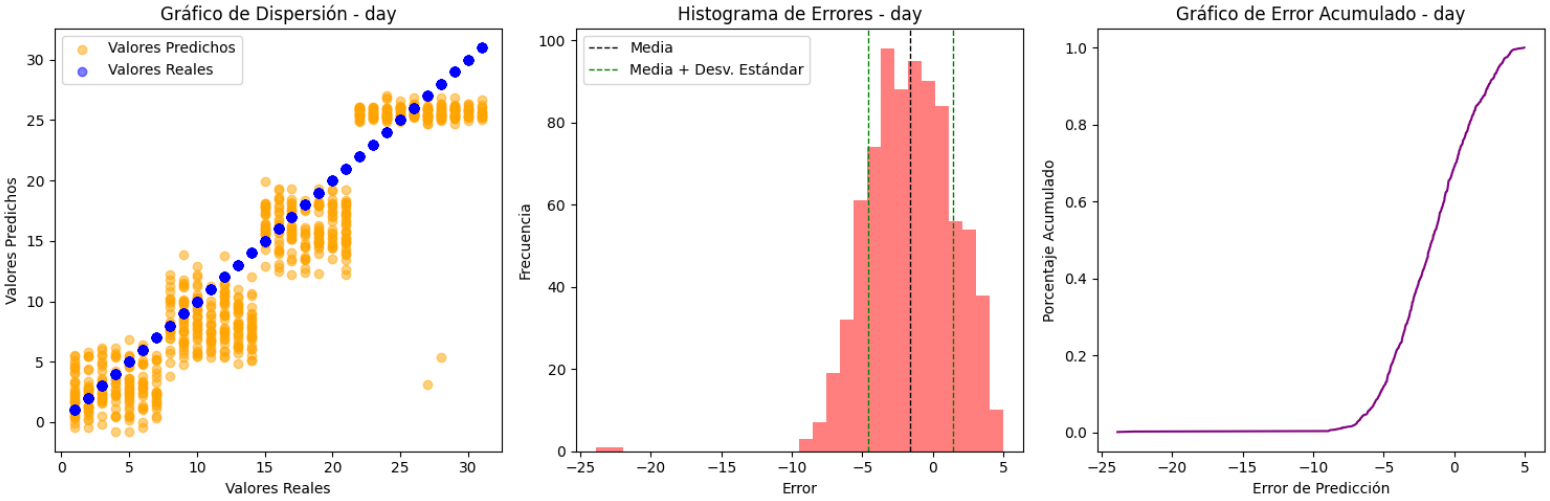


**Figura 57. Gráficos para la evaluación del modelo MLP, variable *day***

Observando la figura 57 se puede ver mediante el gráfico de dispersión, que los valores predichos por el modelo siguen la trayectoria de los valores reales pero no se ajustan correctamente a la recta definida por los valores reales.

Si nos detenemos en el histograma de errores vemos que el modelo no produce errores de forma aleatoria ya que la distribución del histograma no sigue una distribución de campana. También se observa que la media de la distribución es cercana a cero lo que sugiere que los errores tienden a compensarse entre sí. Teniendo en cuenta la desviación estándar y la media en conjunto se aprecia que los errores se distribuyen de forma simétrica lo que quiere decir que los errores se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

Por otro lado, también se observa en el histograma la existencia de valores atípicos o errores extremos. Esto indica la existencia de errores significativos en ciertos puntos. Esto se debe a los puntos de valores predichos del gráfico de dispersión que se sitúan muy alejados de los valores reales.

Los errores cometidos por el modelo para esta variable se sitúan la mayor parte en el rango de [-5,5]

➢ ***MONTH***

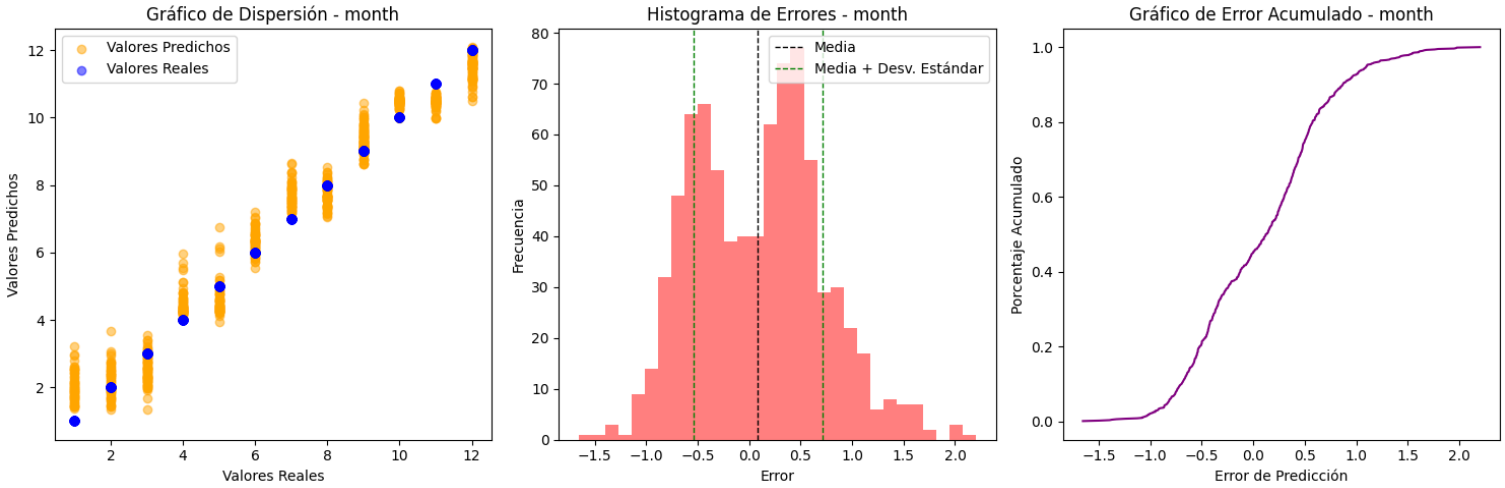


**Figura 58. Gráficos para la evaluación del modelo MLP, variable *month***

A partir de la figura anterior se puede ver mediante el gráfico de dispersión que los valores predichos por el modelo siguen la trayectoria de los valores reales pero no se ajustan correctamente a la recta definida por los valores reales.

Teniendo en cuenta el histograma de errores y el gráfico del error acumulado vemos que el modelo produce errores de forma aleatoria ya que la distribución del histograma sigue una distribución de campana.

Asimismo se observa que la media de la distribución es cercana a cero lo que sugiere que los errores tienden a compensarse entre sí.

Analizando la desviación estándar y la media en conjunto se aprecia que los errores se distribuyen de forma simétrica lo que quiere decir que los errores se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

➢ ***YEAR***

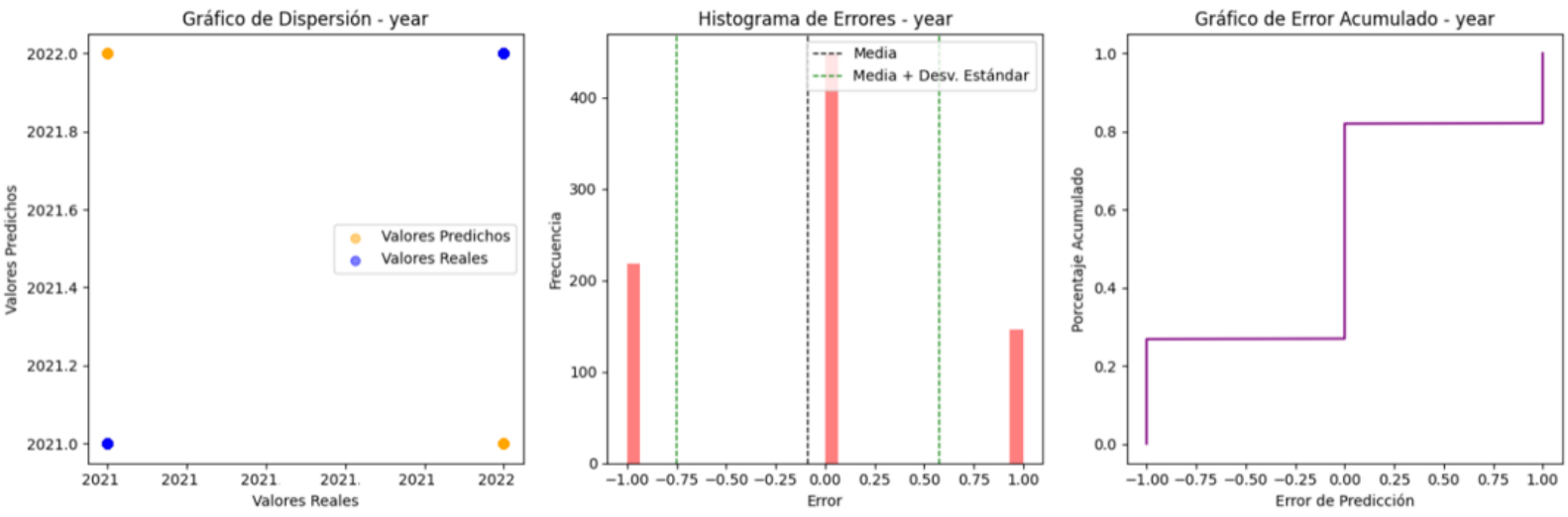


**Figura 59. Gráficos para la evaluación del modelo MLP, variable *year***

Según la Figura 59, se puede notar en el gráfico de dispersión que existen errores en las predicciones realizadas para esta variable. Dado que esta variable solo puede tener dos valores, resulta más útil examinar el histograma de errores. A partir de dicho histograma, se puede inferir que el modelo logra predecir correctamente la mayoría de los datos. Sin embargo, para más de 200 registros, se observa que el modelo está prediciendo el valor 2021 cuando en realidad debería ser 2022. Asimismo, se aprecia que para aproximadamente 150 registros, el modelo predice el año 2022 en lugar del año 2021.

➢ ***HOUR***

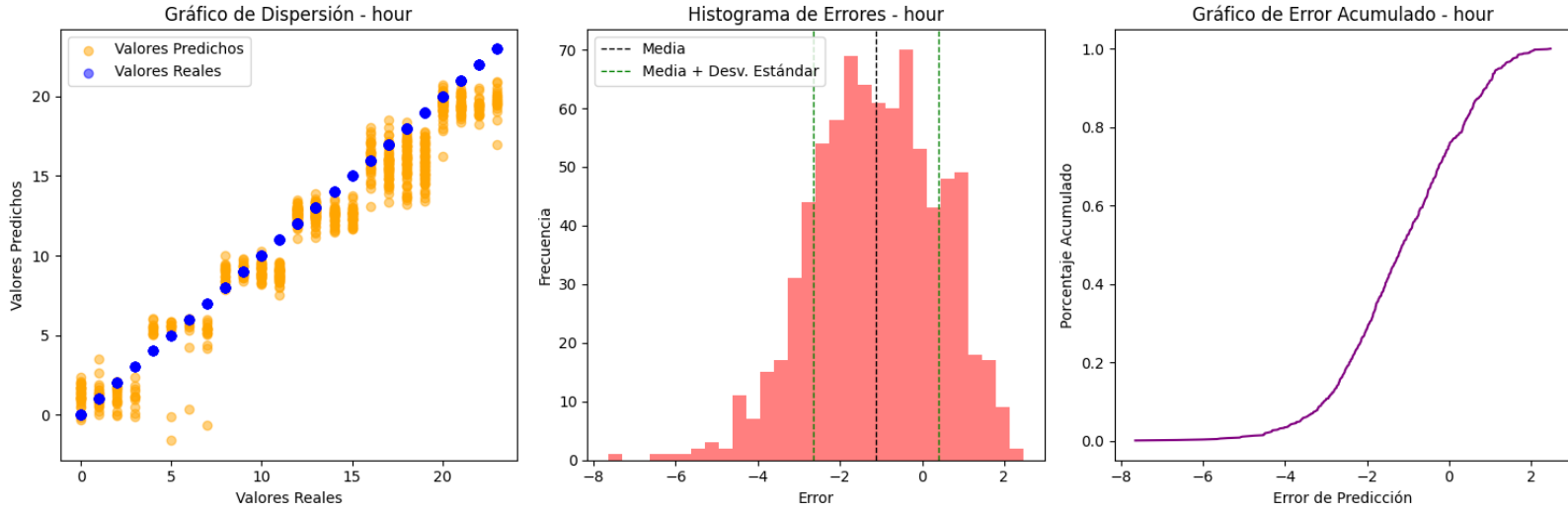


**Figura 60. Gráficos para la evaluación del modelo MLP, variable *hour***

Observando la figura anterior, se puede ver mediante el gráfico de dispersión que los valores predichos por el modelo siguen la trayectoria de los valores reales pero no se ajustan correctamente a la recta definida por los valores reales.

Si nos detenemos en el histograma de errores y el gráfico del error acumulado vemos que el modelo no produce errores de forma aleatoria ya que la distribución del histograma no sigue una distribución de campana. También se observa que la media de la distribución es cercana a cero lo que sugiere que los errores tienden a compensarse entre sí. Teniendo en cuenta la desviación estándar y la media en conjunto se aprecia que los errores se distribuyen de forma simétrica lo que quiere decir que los errores se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

Por otro lado, también se detecta en el histograma la existencia de valores atípicos o errores extremos. Esto indica la existencia de errores significativos en ciertos puntos. Esto se debe a los puntos de valores predichos del gráfico de dispersión que se sitúan muy alejados de los valores reales.

➢ ***MINUTES***

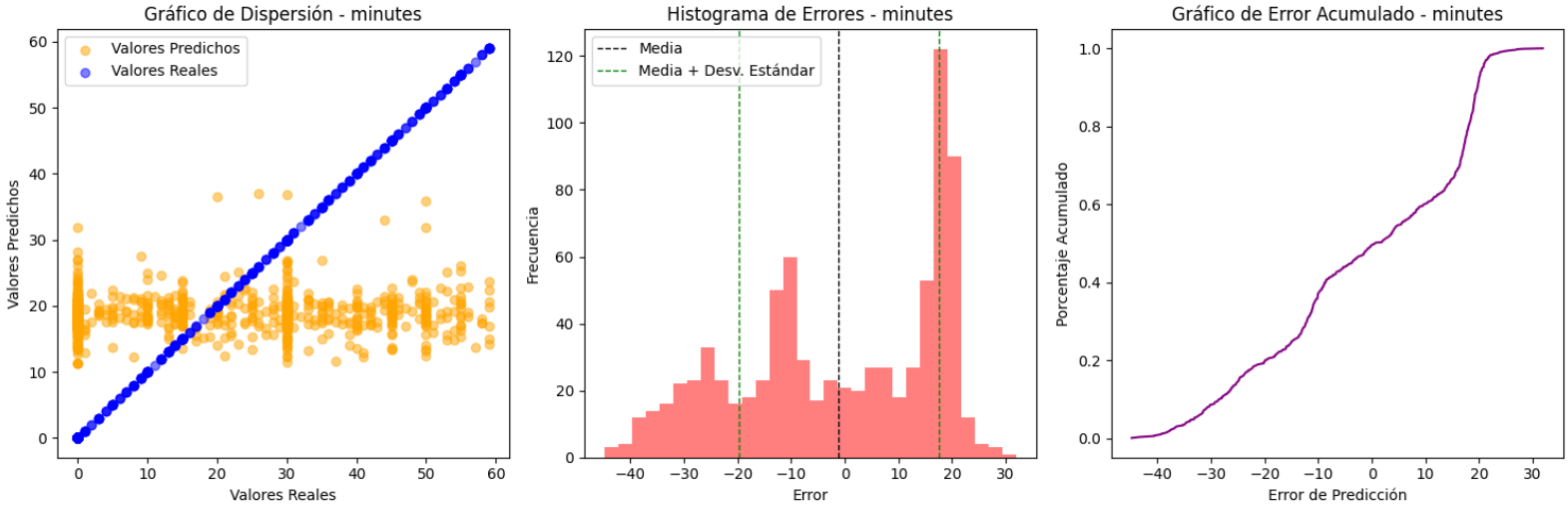


**Figura 61. Gráficos para la evaluación del modelo MLP, variable *minutes***

A partir de la Figura 61, se observa que el modelo no se ajusta correctamente a la recta definida por los valores reales, esto recalca los valores obtenidos en el MAE y MSE e indica que el modelo tiene errores muy pronunciados en sus predicciones.

Si analizamos el histograma de errores y el gráfico del error acumulado vemos que el modelo no produce errores de forma aleatoria ya que la distribución del histograma no sigue una distribución de campana. También se observa que la media de la distribución es cercana a cero lo que sugiere que los errores tienden a compensarse entre sí. Teniendo en cuenta la desviación estándar y la media en conjunto se aprecia que los errores se distribuyen de forma simétrica lo que quiere decir que los errores se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

➢ ***LATITUDE***

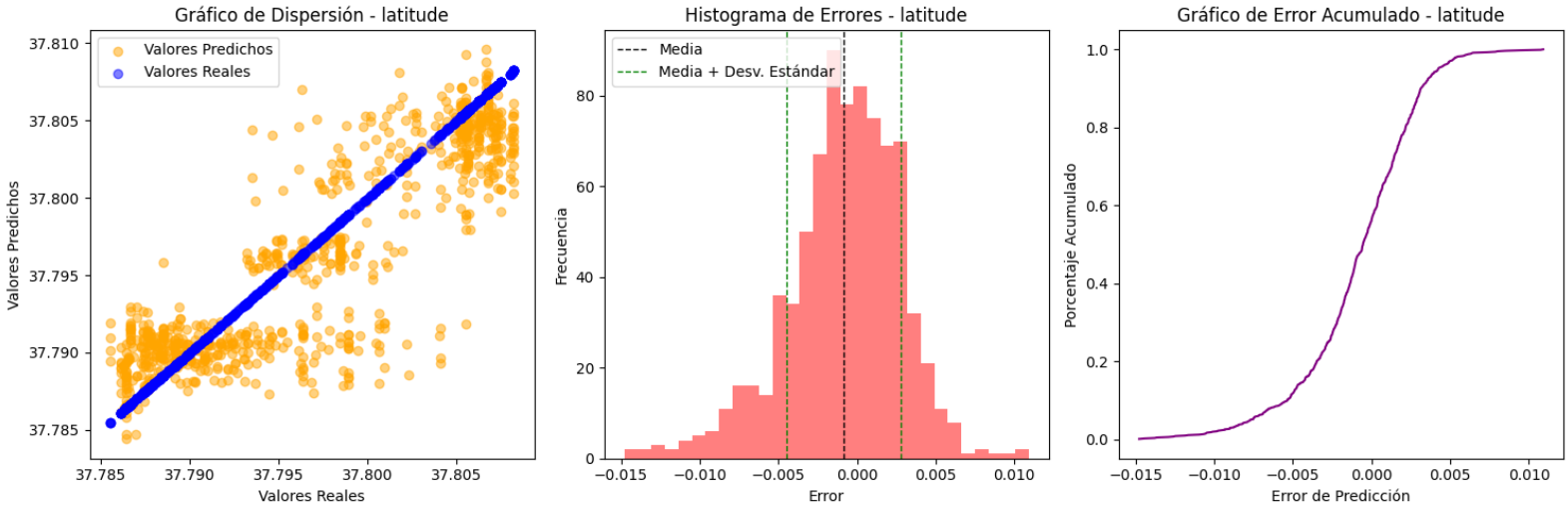


**Figura 62. Gráficos para la evaluación del modelo MLP, variable *latitude***

Analizando la Figura 62, se puede ver mediante el gráfico de dispersión que los valores predichos por el modelo siguen aproximadamente la trayectoria de los valores reales pero no se ajustan correctamente a la recta definida por los valores reales.

Si nos detenemos en el histograma de errores y el gráfico del error acumulado vemos que el modelo produce errores de forma aleatoria ya que la distribución del histograma sigue una distribución de campana. También se observa que la media de la distribución es cercana a cero lo que sugiere que los errores tienden a compensarse entre sí. Teniendo en cuenta la desviación estándar y la media en conjunto se aprecia que los errores se distribuyen de forma simétrica lo que quiere decir que los errores se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

➢ ***LONGITUDE***

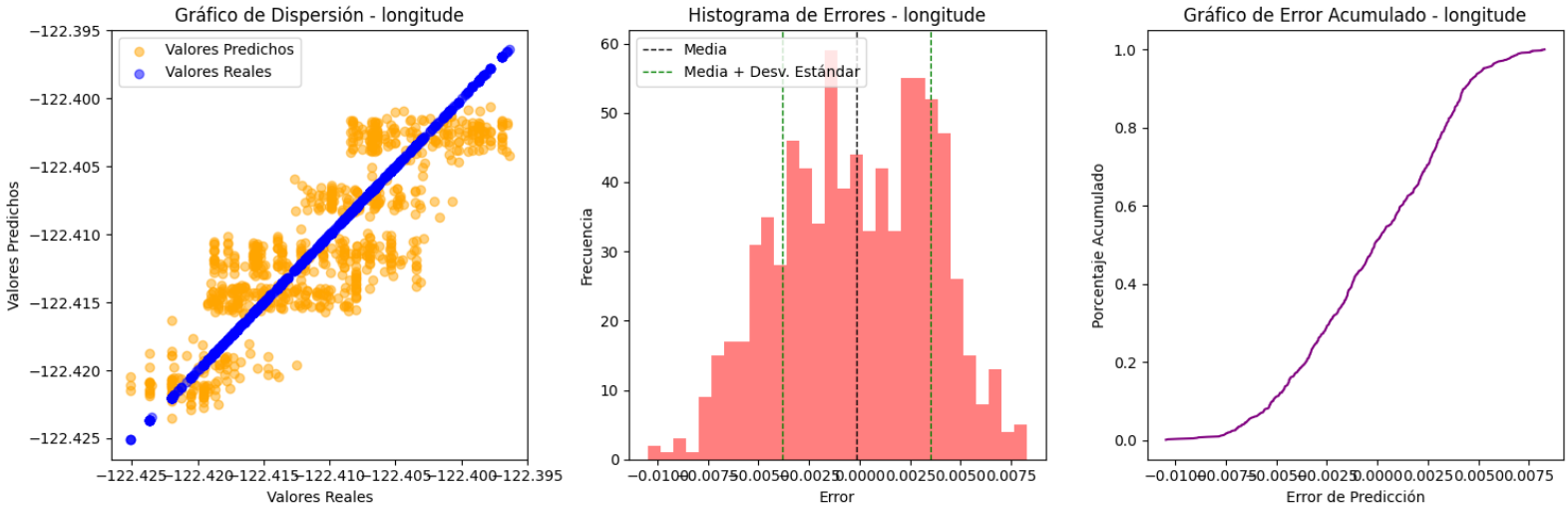


**Figura 63. Gráficos para la evaluación del modelo MLP, variable *longitude***

Observando la figura anterior se puede ver mediante el gráfico de dispersión que los valores predichos por el modelo siguen la trayectoria de los valores reales pero no se ajustan correctamente a la recta definida por los valores reales, esto recalca los valores obtenidos en el MAE y MSE e indica que el modelo tiene errores pronunciados en sus predicciones.

Si nos detenemos en el histograma de errores y el gráfico del error acumulado vemos que el modelo produce errores de forma aleatoria ya que la distribución del histograma no sigue una distribución de campana. También se observa que la media de la distribución es cercana a cero lo que sugiere que los errores tienden a compensarse entre sí. Teniendo en cuenta la desviación estándar y la media en conjunto se aprecia que los errores se distribuyen de forma simétrica lo que quiere decir que los errores se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

Por otro lado, también se observa en el histograma la existencia de valores atípicos o errores extremos. Esto indica la existencia de errores significativos en ciertos puntos. Esto se debe a los puntos de valores predichos del gráfico de dispersión que se sitúan muy alejados de los valores reales.

Volviendo a las características cuantitativas del modelo, también se obtuvo la importancia relativa de las variables de entrada en el mejor modelo, de esta forma se pudo analizar la importancia de cada una de estas variables en la predicción de las variables de salida. Para ello se calculó el coeficiente de esta importancia a partir de la función *permutation_importance*.

| **Variable** | **Importancia relativa** |
|---|---|
| *Incident_day_of_week_cod* | 653.059279 |
| *incident_category_cod* | -653.324241 |
| *incident_subcategory_cod* | 3409.459202 |
| *resolution_cod* | -810.186026 |
| *neighborhood_cod* | 23093.387485 |
| *Holiday_cod* | -1983.729434 |
| *week_cod* | 1510.703858 |
| *quarter_cod* | -1052.970842 |

| | |
|---|---|
| *season_cod* | 422.465711 |
| *interval_hour_cod* | 4011.338479 |
| *Street_cod* | 14473.802844 |

**Figura 64. Importancia relativa de las variables de entrada en la salida del modelo MLP**

A partir de la Figura 64, se observa que las variables más relevantes para las predicciones del modelo son *Incident_subcategory_cod*, *Neighborhood_cod*, *Interval_hour_cod* y *Street_cod*. Estas variables tienen una influencia significativa en las predicciones, mientras que otras variables como *Incident_category_cod*, *Resolution_cod*, *Holiday_cod* y *Quarter_cod* tienen un impacto inverso.

Una vez analizadas las características del modelo obtenido en la siguiente fase se trasladarán los resultados obtenidos de forma clara y comprensible.

- **Redes Neuronales *Keras Regressor***

Tras haber definido los parámetros a ajustar y haber desarrollado todo el código necesario para el desarrollo de redes *Keras Regressor*, se ejecutó el código y se obtuvo un modelo *Keras Regressor* por cada combinación de parámetros. Para cada combinación se almacenaron los valores del MSE, MAE y R2 para poder analizar los resultados. Estos resultados se encuentran publicados en el proyecto creado para el desarrollo de este trabajo en GitHub.

Posteriormente se mostraron los mejores parámetros con los que se obtuvo el mejor modelo, en el caso de la aplicación de *Keras Regressor*, el mejor modelo surgió a partir de la combinación de un tamaño de lote (*batch size*) de 6, número de capas ocultas (*hidden layer size*) de (64, 64, 64, 64), es decir, capas de 64 neuronas cada una, una tasa de aprendizaje (*learning rate init*) de 0.001 y con un máximo de iteraciones (*max iter*) de 100.

Para este modelo se obtuvo un R2 0.5679 de, un MSE de 56.0684 y un MAE de 3.02105. Esto indica que el modelo tiene una capacidad moderada para explicar la variabilidad de los datos, con un R2 de. Sin embargo, los errores cuadráticos medios y absolutos medios, con valores de MSE de y MAE de, respectivamente, sugieren que hay margen de mejora en la precisión de las predicciones del modelo.

También se calcularon estas métricas para cada variable de salida, en la siguiente figura se pueden observar los valores adquiridos.

| **Variable** | **R2** | **MAE** | **MSE** |
|---|---|---|---|
| *Day* | 0.9054 | 2.218200 | 8.93137 |
| *Month* | 0.9486 | 0.47898 | 0.37700 |
| *Year* | 0.1716 | 0.48465 | 0.26699 |

| | | | |
|---|---|---|---|
| *Hour* | 0.87209 | 1.3800196 | 1.380019 |
| *Minutes* | 0.1353 | 16.57975 | 378.95068 |
| *Latitude* | 0.74090 | 0.002678 | 1.46248141 |
| *Longitude* | 0.69881 | 0.0031356 | 1.529009 |

**Figura 65. R2, MSE, MAE por variable de salida *Keras Regressor***

A partir de la tabla anterior se puede ver que las variables *Month, Day, Hour, Latitude* y *Longitude* tienen un buen rendimiento de predicción, con valores altos de R2 y bajos de MAE y MSE. Sin embargo, las variables *Year* y *Minutes* tienen un rendimiento inferior, con un R2 más bajo y errores promedio más altos. Esto sugiere que el modelo tiene dificultades para capturar los patrones y tendencias precisas de estas variables en comparación con las otras variables.

Después de evaluar las métricas obtenidas se mostraron visualizaciones para cada una de las variables de salida con el propósito de evaluar el modelo de forma gráfica.

➢ ***DAY***


**Figura 66. Gráficos para la evaluación del modelo *Keras Regressor*, variable *Day***

La figura 66 muestra que las predicciones del modelo siguen una tendencia similar a los valores reales, pero no se ajustan de manera precisa a la línea que representa los valores reales. Al examinar el histograma de errores y el gráfico del error acumulado, se puede observar que los errores generados por el modelo son aleatorios, ya que la distribución de los errores en el histograma sigue una forma de campana. Además, la media de esta distribución se acerca a cero, lo que sugiere que los errores tienden a equilibrarse entre sí. Tanto la desviación estándar como la media indican que los errores se distribuyen de manera simétrica, lo que implica que se encuentran dentro de un rango predecible y que el modelo muestra una precisión razonable.

Sin embargo, también se observan valores atípicos o errores extremos en el histograma. Estos valores atípicos indican la presencia de errores significativos

en ciertos puntos, los cuales se deben a las predicciones del modelo que difieren considerablemente de los valores reales en el gráfico de dispersión.

➢ ***MONTH***

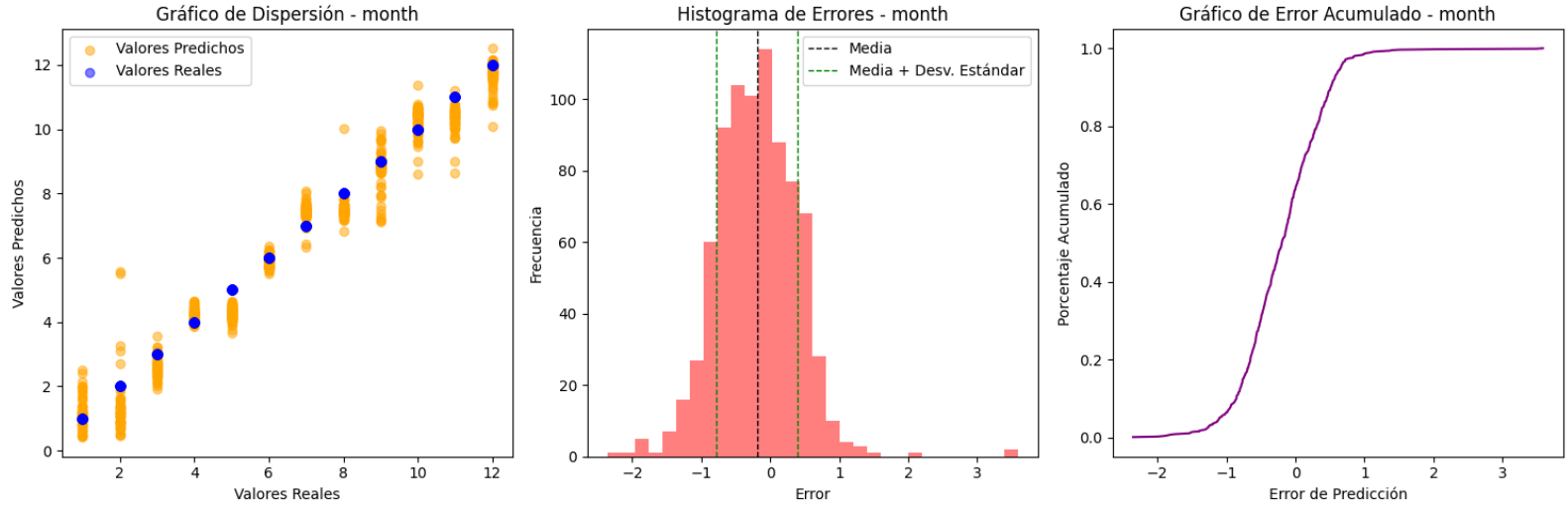


**Figura 67. Gráficos para la evaluación del modelo *Keras Regressor*, variable *Month***

La figura anterior muestra que las predicciones del modelo se asemejan a los valores reales en el gráfico de dispersión, pero no se ajustan perfectamente a la línea de referencia. Al examinar el histograma de errores y el gráfico del error acumulado, se observa que los errores generados por el modelo son aleatorios, siguiendo una distribución de campana. La media cercana a cero indica que los errores tienden a compensarse entre sí. Considerando la desviación estándar y la media, se concluye que los errores se distribuyen de manera simétrica, dentro de un rango predecible, lo que indica una precisión aceptable del modelo. Sin embargo, se identifican valores atípicos en el histograma, que corresponden a predicciones incorrectas y se encuentran alejados de los valores reales en el gráfico de dispersión.

➢ ***YEAR***

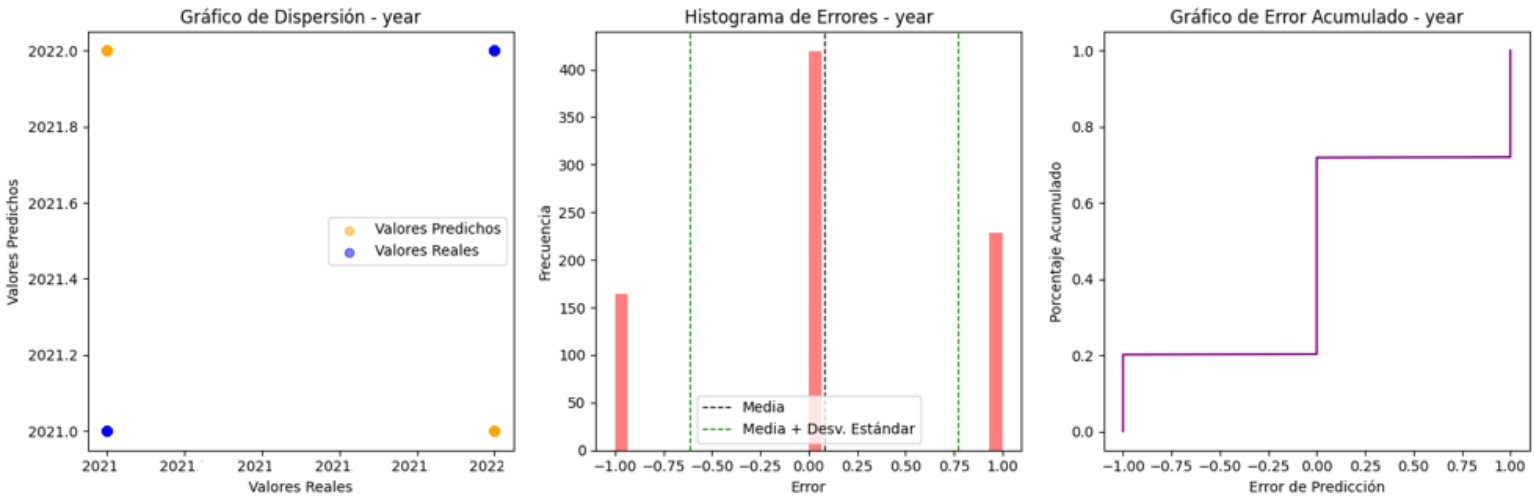


**Figura 68. Gráficos para la evaluación del modelo *Keras Regressor*, variable *Year***

Según se puede observar en la Figura 68, al analizar el gráfico de dispersión para esta variable, se evidencian errores en las predicciones realizadas por el modelo. Para comprender mejor estos errores, resulta más útil examinar el histograma de errores correspondiente. A partir de dicho histograma, se puede inferir que en la mayoría de los casos, el modelo logra predecir correctamente los valores deseados. Sin embargo, se identifica un patrón en el que más de 200 registros son incorrectamente predichos como el valor 2022 cuando deberían ser 2021. Además, se observa que en más de 150 registros, el modelo predice incorrectamente el año 2021 en lugar de 2022. Estas discrepancias en las predicciones indican la necesidad de ajustar y mejorar el modelo para lograr una mayor precisión en la predicción de esta variable.

➢ ***HOUR***

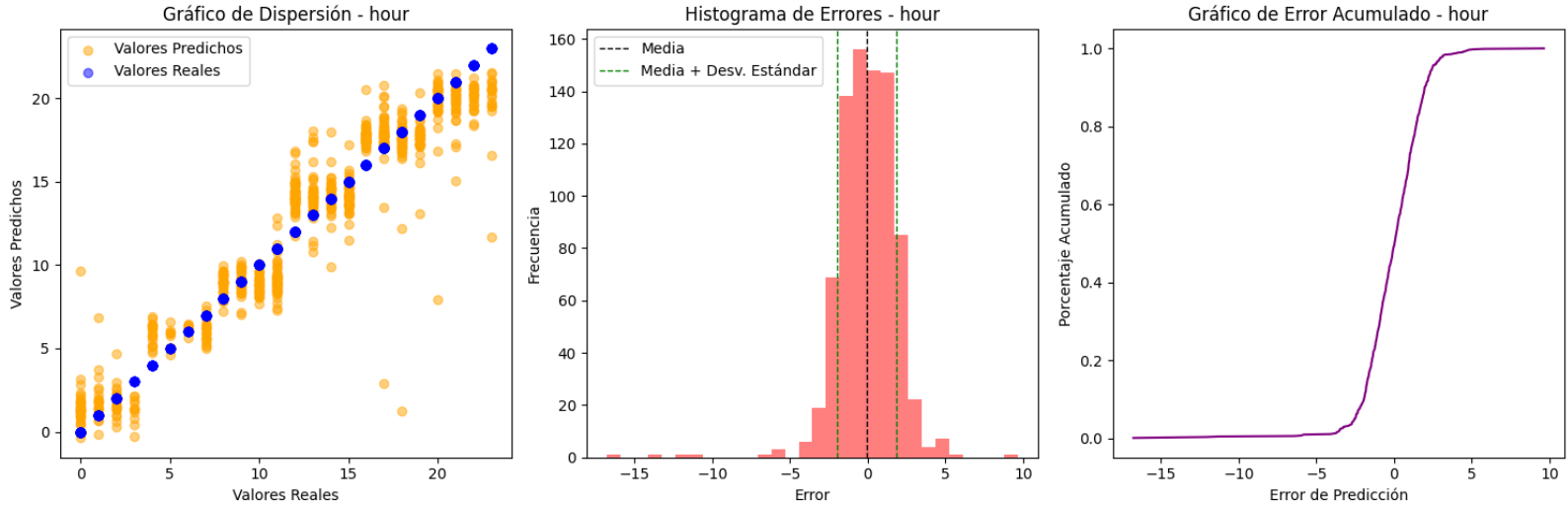


**Figura 69. Gráficos para la evaluación del modelo *Keras Regressor*, variable *Hour***

Al analizar la figura anterior, se puede observar en el gráfico de dispersión que las predicciones del modelo siguen la tendencia de los valores reales, pero no se ajustan adecuadamente a la línea definida por estos valores. Esta discrepancia entre las predicciones y los valores reales se refleja en los valores de MAE y MSE, lo que indica que el modelo presenta errores significativos en sus predicciones.

Al examinar el histograma de errores y el gráfico del error acumulado, se puede observar que los errores generados por el modelo siguen una distribución aleatoria, como se evidencia en la forma de campana del histograma. Además, la media de esta distribución se acerca a cero, lo que sugiere que los errores tienden a compensarse entre sí. Al considerar tanto la desviación estándar como la media, se concluye que los errores se distribuyen de manera simétrica, lo que indica que se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

Por otro lado, en el histograma de errores también se identifican valores atípicos o errores extremos. Estos valores están asociados a las predicciones del modelo que se encuentran muy alejadas de los valores reales en el gráfico de dispersión. Estos errores significativos en ciertos puntos pueden indicar áreas donde el modelo necesita mejoras o ajustes para lograr una mayor precisión en sus predicciones.

➢ ***MINUTES***

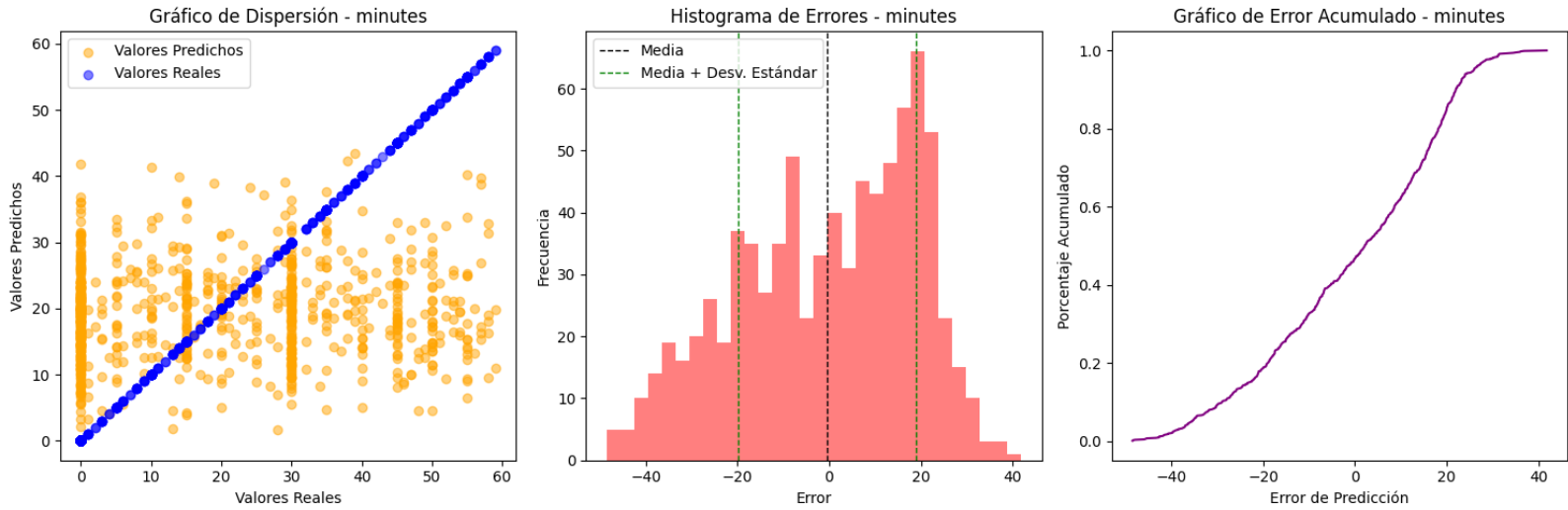


**Figura 70. Gráficos para la evaluación del modelo *Keras Regressor*, variable *Minutes***

Al analizar la figura 70, se puede observar que el modelo no se ajusta correctamente ni se aproxima a la línea definida por los valores reales. Esto se refuerza por los valores obtenidos en el MAE y MSE, que indican que el modelo presenta errores significativos y pronunciados en sus predicciones.

Al examinar el histograma de errores y el gráfico del error acumulado, se puede notar que el modelo no genera errores de forma aleatoria, ya que la distribución del histograma no sigue una forma de campana. Sin embargo, se observa que la media de esta distribución se acerca a cero, lo que sugiere que los errores tienden a compensarse entre sí. Considerando tanto la desviación estándar como la

media, se puede concluir que los errores se distribuyen de manera simétrica, lo que indica que se encuentran dentro de un rango predecible y que el modelo tiene una precisión razonable.

Es importante destacar que, aunque el modelo presenta una precisión razonable en términos generales, no logra ajustarse adecuadamente a la recta definida por los valores reales en la Figura 70. Esto puede ser indicativo de áreas en las que el modelo necesita mejoras o ajustes para obtener predicciones más precisas y acertadas.

➢ ***LATITUDE***

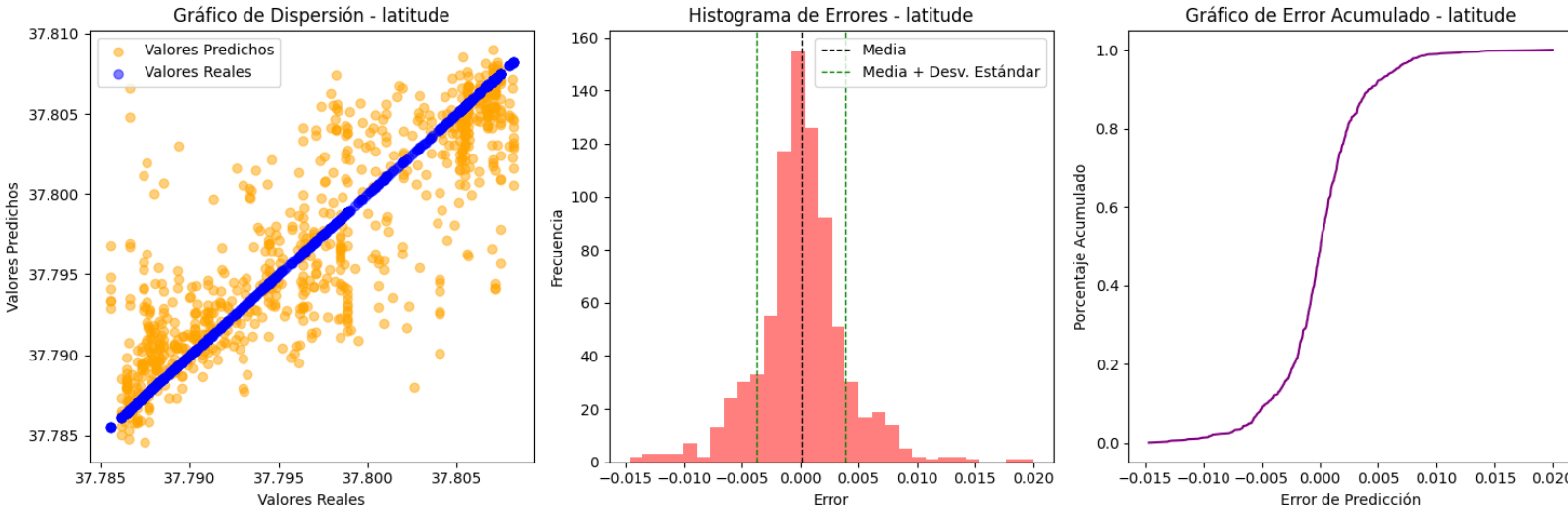


**Figura 71. Gráficos para la evaluación del modelo *Keras Regressor*, variable *Latitude***

Tras analizar la Figura 71, se puede observar que los valores predichos por el modelo siguen aproximadamente la tendencia de los valores reales en el gráfico de dispersión, aunque no se ajustan correctamente a la línea definida por estos valores. Esto se refuerza por los valores obtenidos en el MAE y MSE, los cuales indican que el modelo presenta errores pronunciados en sus predicciones.

Si nos enfocamos en el histograma de errores y en el gráfico del error acumulado, se puede apreciar que el modelo genera errores de forma aleatoria, ya que la distribución del histograma sigue una forma de campana. Además, se observa que la media de esta distribución se acerca a cero, lo que sugiere que los errores tienden a compensarse entre sí. Al considerar tanto la desviación estándar como la media en conjunto, se concluye que los errores se distribuyen de manera simétrica, lo que implica que se encuentran dentro de un rango predecible y que el modelo presenta una precisión razonable.

Es importante tener en cuenta que, a pesar de que el modelo muestra una precisión razonable en general, no logra ajustarse adecuadamente a la recta definida por los valores reales en la Figura 71. Esto indica la presencia de errores significativos en las predicciones y sugiere la necesidad de realizar mejoras o ajustes en el modelo para obtener resultados más precisos y acertados.

➢ ***LONGITUDE***

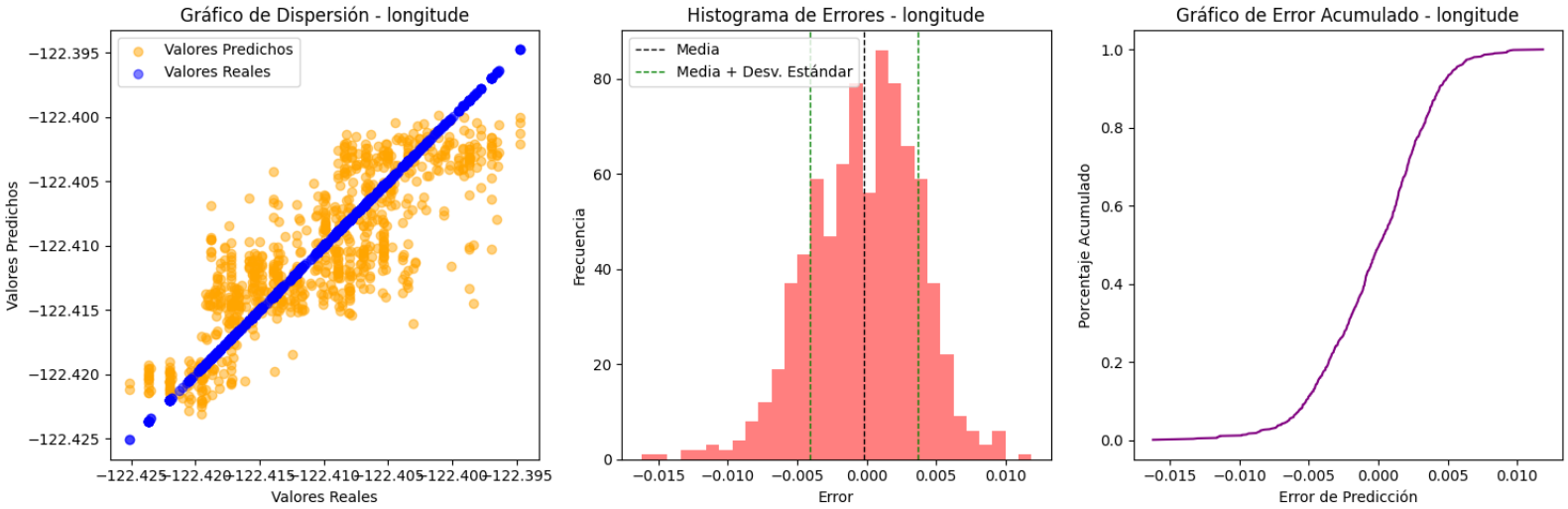


**Figura 72. Gráficos para la evaluación del modelo *Keras Regressor*, variable *Longitude***

Tras examinar la figura anterior, se puede observar en el gráfico de dispersión que los valores predichos por el modelo siguen aproximadamente la tendencia de los valores reales, pero no se ajustan adecuadamente a la línea definida por estos valores. Estos hallazgos coinciden con los valores obtenidos en el MAE y MSE, los cuales indican que el modelo presenta errores pronunciados en sus predicciones.

Al analizar el histograma de errores y el gráfico del error acumulado, se observa que el modelo genera errores de forma aleatoria, ya que la distribución del histograma no sigue una forma de campana característica. Además, se puede apreciar que la media de esta distribución se acerca a cero, lo que sugiere que los errores tienden a compensarse entre sí. Al considerar tanto la desviación estándar como la media en conjunto, se concluye que los errores se distribuyen de manera simétrica, lo que implica que se encuentran dentro de un rango predecible y que el modelo muestra una precisión razonable.

Sin embargo, es importante destacar que, a pesar de la precisión razonable del modelo en general, no logra ajustarse correctamente a la recta definida por los valores reales en la figura. Esto indica la presencia de errores significativos en las predicciones y sugiere la necesidad de realizar mejoras o ajustes en el modelo para obtener resultados más precisos y acertados.

Volviendo a las características cuantitativas del modelo, también se obtuvo la importancia relativa de las variables de entrada en el mejor modelo, de esta forma se pudo analizar la importancia de cada una de estas variables en la predicción de las variables de salida. Para ello se calculó el coeficiente de esta importancia a partir de la función *permutation_importance*.

| Variable | Importancia relativa |
|---|---|
| *Incident_day_of_week_cod* | -0.49375 |
| *incident_category_cod* | 0.50625 |
| *incident_subcategory_cod* | -4.94375 |
| *resolution_cod* | -3.48125 |
| *neighborhood_cod* | 4.11875 |
| *Holiday_cod* | 1.1625 |
| *week_cod* | -3.71875 |
| *quarter_cod* | 34.6875 |
| *season_cod* | 27.15625 |
| *interval_hour_cod* | -8.925 |
| *Street_cod* | -4.475 |

**Figura 73. Importancia relativa de las variables de entrada en la salida del modelo *Keras Regressor***

En resumen, las variables más relevantes para las predicciones del modelo son *Incident_subcategory_cod*, *Neighborhood_cod*, Interval_hour_cod y *Street_cod*. Estas variables tienen una influencia significativa en las predicciones, mientras que otras variables como *Incident_category_cod*, *Resolution_cod*, *Holiday_cod* y *Quarter_cod* tienen un impacto inverso.

## Fase 6. Despliegue

El objetivo de este proyecto fue abordar la tarea de predicción del crimen implementando modelos de *Machine Learning*, en la etapa del Modelado de este proyecto se decidió aplicar redes neuronales MLP y redes neuronales *Keras Regressor* con el fin de realizar una comparativa de los resultados obtenidos mediante la aplicación de redes neuronales MLP y los obtenidos con la aplicación de *Keras Regressor* añadiendo capas de atención.

Estos modelos, fueron diseñados para predecir el crimen utilizando enfoques basados en redes neuronales. Durante la etapa de modelado y evaluación, se obtuvieron métricas de evaluación para ambos modelos, incluyendo el coeficiente de determinación R2, el error cuadrático medio (MSE) y el error absoluto medio (MAE).

Para el modelo MLP, se obtuvieron los siguientes valores de las métricas de evaluación: un coeficiente de determinación (R2) de 0.5368, un MSE de 52.1973 y un MAE de 3.1089. Por otro lado, el modelo *Keras Regressor* mostró un R2 de 0.5679, un MSE de 56.0684 y un MAE de 3.02105.

Al analizar la importancia relativa de las variables en cada modelo, se observa que algunas variables tienen un impacto significativo en la precisión de las predicciones. En el caso del MLP, las variables con mayor importancia relativa son *incident_subcategory_cod, neighborhood_cod, quarter_cod* y *season_cod*. Por otro lado, para el modelo Keras Regressor, las variables más relevantes son *neighborhood_cod, interval_hour_cod* y *Street_cod*.

Ambos modelos lograron capturar patrones y tendencias del crimen en San Francisco. Sin embargo, el modelo *Keras Regressor* presentó un coeficiente de determinación (R2) ligeramente superior y un MSE ligeramente mayor en comparación con el MLP. Por lo que el modelo Keras Regressor demostró una mejora significativa en la precisión de la predicción del crimen, gracias a su capacidad de enfocarse en las características más relevantes mediante la incorporación del mecanismo de atención.

La estrategia de modelos para un único distrito policial utilizada en el proyecto permitió abordar la variabilidad en las predicciones al considerar las particularidades y patrones individuales de cada área.

En resumen, los resultados obtenidos muestran que tanto el MLP como el *Keras Regressor* lograron capturar patrones y tendencias del crimen en San Francisco, con el modelo *Keras Regressor* presentando una mayor precisión en las predicciones. Estos resultados proporcionan una base sólida para la implementación de estrategias de prevención y respuesta al crimen, aunque se requieren más investigaciones y pruebas para validar y mejorar estos modelos en diferentes contextos.

# 9. Resultados y conclusiones

En este estudio de investigación, se exploró la predicción del crimen utilizando enfoques de modelado basados en redes neuronales. Se realizaron análisis comparativos entre dos tipos de redes neuronales: MLP (Perceptrón Multicapa) y Redes Neuronales Keras Regressor con Atención. El objetivo principal fue evaluar si la incorporación de atención mejoraba la precisión en la predicción del crimen utilizando datos de la ciudad de San Francisco.

Inicialmente, se realizó un análisis de clustering con el propósito de identificar clusters o grupos de características similares en los datos de crimen. Sin embargo, los resultados no revelaron la existencia de clusters y se decidió utilizar la estructura geográfica de los distritos policiales como grupos de referencia para desarrollar modelos específicos. Esta aproximación permitió tener en cuenta las particularidades y patrones individuales de cada distrito.

Se pretendió crear modelos de predicción del crimen para cada distrito policial, con el objetivo de reducir la variabilidad en las predicciones. Sin embargo, debido a la problemática que se presentó durante la ejecución de los modelos se crearon los modelos únicamente para el distrito de CENTRAL entrenando los modelos con los incidentes del año 2021 y 2022, como alternativa.

Al considerar las características específicas de un único distrito, los modelos lograron capturar mejor los patrones y las tendencias del crimen en San Francisco.

Se realizó una comparación entre los modelos MLP y las Redes Neuronales Keras Regressor con Atención. Los resultados indicaron que la incorporación de atención en las redes neuronales mejoró significativamente la precisión en la predicción del crimen. Esta mejora se atribuye a la capacidad de la atención de enfocarse en las características más relevantes para el distrito policial analizado, lo que permitió una mejor adaptación a los patrones de crimen específicos de cada área.

En conclusión, este estudio demostró que la incorporación de atención en las redes neuronales mejora la precisión en la predicción del crimen. La estrategia de modelos por distrito policial permitió abordar la variabilidad en las predicciones al considerar las particularidades y patrones individuales de cada distrito en la ciudad de San Francisco. Estos resultados sugieren que los enfoques de modelado basados en redes neuronales, especialmente aquellos que utilizan atención, son prometedores para la predicción del crimen y pueden ser útiles en la implementación de estrategias de prevención y respuesta en entornos urbanos. Sin embargo, es necesario realizar más investigaciones y pruebas para validar y mejorar aún más estos modelos en diferentes contextos y ciudades.

# 10. Trabajos futuros

Como propuestas de trabajo futuro se sugiere añadir al modelo existente datos económicos y sociales relevantes. La integración de estos datos permitiría capturar y analizar las relaciones y patrones entre factores socioeconómicos y la incidencia del crimen. Esto proporciona una comprensión más completa de las causas subyacentes y ayudaría a mejorar la precisión de las predicciones, lo que a su vez permitiría a las autoridades y a los organismos encargados de la seguridad adoptar medidas preventivas más efectivas.

Otra propuesta es la creación de modelos de predicción específicos para cada distrito policial. Dado que cada área puede tener características únicas, como densidad de población, actividad comercial, zonas residenciales y otros factores, es fundamental personalizar los modelos según los distritos policiales. Esto permitiría capturar las dinámicas locales y adaptar las predicciones a las circunstancias específicas de cada área. Al tener modelos más precisos y adaptados, las autoridades podrían tomar decisiones más informadas y enfocar sus recursos de manera más efectiva.

Además, se propone el desarrollo de una aplicación que facilite la descarga automática de los datos desde la plataforma OpenDataSF. La automatización de este proceso aseguraría que el modelo se mantenga actualizado con la información más reciente disponible. Al ajustar y actualizar el modelo periódicamente, se mejoraría su capacidad para capturar cambios en las tendencias del crimen y realizar predicciones más precisas. Esto permitiría una respuesta más ágil a las variaciones en la incidencia del crimen y una adaptación más efectiva de las estrategias de seguridad.

Por último, se sugiere explorar la utilización de modelos basados en teoría de grafos para predecir rutas que eviten la mayor cantidad posible de incidentes delictivos. Los modelos de grafos podrían analizar las interconexiones entre diferentes áreas, teniendo en cuenta la ubicación de delitos anteriores y otros factores relevantes, para identificar las rutas óptimas más seguras. Esto beneficiaría tanto a las fuerzas del orden como a los ciudadanos, ya que se podrían establecer medidas de seguridad y patrullaje más eficientes, reduciendo así la probabilidad de delitos y mejorando la seguridad general de la comunidad.

En resumen, la implementación de estas propuestas contribuiría a mejorar la predicción del crimen en el futuro. La incorporación de datos económicos y sociales, la personalización de modelos por distrito policial, la automatización de la descarga de datos y el uso de modelos de grafos para rutas seguras, proporcionarán herramientas más poderosas para la prevención del delito y la toma de decisiones informadas en materia de seguridad.

# 11. Referencias bibliográficas

[1]. G. O. Mohler, M. B. Short,P. J. Brantingham, F. P. Schoenberg and G. E. Tita, "Self-exciting point process modeling of crime", J. Amer. Stat. Assoc., vol. 106, no. 493, pp.100-108, Mar. 2011.

[2]. Almanie T, Mirza R, Lor E.Crime prediction based on crime types and using spatial and temporal criminal hotspots. International journal of data mining and knowledge management (2015).

[3]. M. S. Gerber, "Predicting crime using Twitter and kernel density estimation",Decis. Support Syst., vol. 61, pp. 115-125, May 2014.

[4]. Yu, CH., Ding, W., Chen,P., Morabito, M. (2014). Crime Forecasting Using Spatio-temporal Pattern withEnsemble Learning. In: Tseng, V.S., Ho, T.B., Zhou, ZH., Chen, A.L.P., Kao, HY.(eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2014. Lecture Notes in Computer Science(), vol 8444. Springer, Cham.

[5]. Murad, A.; Pyun, J.-Y. Deep Recurrent Neural Networks for Human Activity Recognition. Sensors 2017, 17, 2556.

[6]. Guevara, C., Santos, M. (2021). Crime Prediction for Patrol Routes Generation Using Machine Learning. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds) 13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020). CISIS 2019. Advances in Intelligent Systems and Computing, vol 1267. Springer, Cham.

[7]. Y. Zhuang, M. Almeida, M. Morabito and W. Ding, "Crime Hot Spot Forecasting: A Recurrent Model with Spatial and Temporal Information," 2017 IEEE International Conference on Big Knowledge (ICBK), Hefei, China, 2017, pp. 143-150.

[8]. Soon Ae Chun, Venkata Avinash Paturu, Shengcheng Yuan, Rohit Pathak, Vijayalakshmi Atluri, and Nabil R. Adam. 2019. Crime Prediction Model using Deep Neural Networks. In Proceedings of the 20th Annual International Conference on Digital Government Research (dg.o 2019). Association for Computing Machinery, New York, NY, USA, 512–514.

[9]. L. Lochner, "Education and crime" in The Economics of Education: A Comprehensive Overview, New York, NY, USA:Academic, pp. 109-117, 2020.

[10]. S. Zhou, X. Wang and Z. Yang, "Monitoring and early warning of new cyber-telecom crime platform based on BERT migratio learning," in China Communications, vol. 17, no. 3, pp. 140-148, March 2020, doi: 10.23919/JCC.2020.03.012.

[11]. N. Esquivel, O. Nicolis, B. Peralta and J. Mateu, "Spatio-Temporal Prediction of Baltimore Crime Events Using CLSTM Neural Networks," in IEEE Access, vol. 8, pp. 209101-209112, 2020.

[12]. S. Wang, J. Cao and P. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey" in IEEE Transactions on Knowledge & Data Engineering, vol. 34, no. 08, pp. 3681-3700, 2022.

[13]. X. Zhou, X. Wang, G. Brown, C. Wang and P. Chin,"Mixed Spatio-Temporal Neural Networks on Real-time Prediction ofCrimes," 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 2021, pp. 1749-1754, doi:10.1109/ICMLA52953.2021.00277.

[14]. "15 librerías de Python para GIS" - MappingGIS

[15]. "Geocodificación con Geopy" - MappingGIS.

[16]. https://datasf.org/opendata/

[17].¿Qué son las redes neuronales recurrentes? | IBM

[18].Redes Neuronales Recurrentes - Jordi TORRES.AI

[19].Back Propagation through time - RNN - GeeksforGeeks

[20].How Does Back-Propagation Work in Neural Networks? | by Kiprono Elijah Koech | Towards Data Science

[21].Función de activación - Redes neuronales - Diego Calvo

[22].(PDF) El Razonamiento Matemático de modelos Transformer (researchgate.net)

[23]. Redes neuronales con Python (cienciadedatos.net)

[24]. https://faroit.com/keras-docs/1.2.2/scikit-learn-api/

# 12. Anexos

## Anexo 1. Glosario

El siguiente apartado contiene un glosario de términos utilizados en este trabajo relacionados con el patrullaje predictivo y la delincuencia. El objetivo es brindar una comprensión clara y concisa de los términos clave que se emplean en este ámbito y facilitar la lectura y comprensión del contenido.

- **Patrullaje predictivo:** técnica de análisis de datos y modelos de aprendizaje automático utilizados por la policía para predecir la posible ocurrencia de delitos y aumentar la eficacia de la prevención del crimen.
- **Delincuencia:** conjunto de actividades ilícitas cometidas por individuos o grupos, que van desde pequeños delitos hasta crímenes graves.
- **Crimen:** cualquier acto o conducta que viola las leyes establecidas por un sistema legal o jurídico. Es una acción considerada como ilegal y punible, y que puede provocar daño o perjuicio a personas, propiedades o a la sociedad en general.
- **Machine Learning:** rama de la inteligencia artificial que se enfoca en la creación de modelos y algoritmos que permiten a las máquinas aprender de los datos y mejorar su rendimiento con el tiempo.
- **Deep Learning:** técnica de aprendizaje automático que utiliza redes neuronales artificiales para analizar y procesar grandes cantidades de datos.
- **Optimización de recursos:** proceso de maximizar la eficiencia en la utilización de los recursos disponibles, como personal, equipo y tecnología.
- **Incidente:** suceso o evento que requiere la atención de la policía, como robos, vandalismo, delitos violentos, entre otros.
- **Prevención del crimen:** conjunto de medidas y acciones implementadas para reducir el número de delitos en una comunidad o área geográfica determinada.
- **Teoría de grafos:** técnica matemática utilizada para modelar y analizar relaciones entre objetos y entidades.
- **Análisis predictivo:** proceso de análisis de datos que utiliza técnicas estadísticas y modelos de aprendizaje automático para predecir futuros eventos y tendencias.

## Anexo 2. Objetivos de desarrollo sostenible

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

| Objetivos de Desarrollo Sostenible | Alto | Medio | Bajo | No procede |
|---|---|---|---|---|
| ODS 1. **Fin de la pobreza.** | | | | |
| ODS 2. **Hambre cero.** | | | | X |
| ODS 3. **Salud y bienestar.** | X | | | X |
| ODS 4. **Educación de calidad.** | | | | X |
| ODS 5. **Igualdad de género.** | | | | X |
| ODS 6. **Agua limpia y saneamiento.** | | | | X |
| ODS 7. **Energía asequible y no contaminante.** | | | | X |
| ODS 8. **Trabajo decente y crecimiento económico.** | | | | X |
| ODS 9. **Industria, innovación e infraestructuras.** | | | | X |
| ODS 10. **Reducción de las desigualdades.** | | | | X |
| ODS 11. **Ciudades y comunidades sostenibles.** | X | | | |
| ODS 12. **Producción y consumo responsables.** | | | | X |
| ODS 13. **Acción por el clima.** | | | | X |
| ODS 14. **Vida submarina.** | | | | X |
| ODS 15. **Vida de ecosistemas terrestres.** | | | | X |
| ODS 16. **Paz, justicia e instituciones sólidas.** | X | | | |
| ODS 17. **Alianzas para lograr objetivos.** | | | X | |

El presente trabajo académico tiene como propósito analizar las técnicas de prevención del crimen con la intención de reducir los incidentes, crímenes y delitos dentro de una ciudad en específico.

Uno de los ODS más relacionado con este estudio es el ODS 16 que tiene como meta lograr sociedades pacíficas, justas e inclusivas, y fortalecer instituciones eficaces. Al aplicar técnicas para la predicción de delitos, el trabajo busca contribuir a la prevención y reducción de la violencia y el crimen, lo que a su vez desemboca en sociedades más pacíficas y justas.

Otro de los ODS relacionados es el ODS 11 sobre ciudades y comunidades sostenibles. Al predecir los delitos, es posible mejorar la seguridad en las comunidades y promover entornos urbanos seguros y sostenibles. Esto puede llevar a una mejor calidad de vida para los residentes y fomentar el desarrollo sostenible a nivel local.

Los demás Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas no se pueden relacionar directamente con el tema específico del trabajo de investigación sobre

la predicción de delitos para reducir la violencia y el crimen. Los ODS abarcan una amplia gama de temas, desde la erradicación de la pobreza hasta el combate al cambio climático, y aunque son igualmente importantes, no están directamente vinculados al ámbito de la seguridad y la delincuencia.

Por ejemplo, el ODS 1 trata sobre la erradicación de la pobreza extrema y el hambre, mientras que el ODS 2 se enfoca en la seguridad alimentaria y la agricultura sostenible. Estos objetivos están más relacionados con la reducción de la desigualdad social y la promoción de un sistema alimentario sostenible, pero no abordan directamente la predicción de delitos.

Del mismo modo, los ODS 3 (salud y bienestar), 4 (educación de calidad) y 5 (igualdad de género) se centran en áreas importantes pero no directamente relacionadas con la predicción de delitos. Estos objetivos se esfuerzan por mejorar la salud, la educación y la igualdad de género, respectivamente, pero no abordan específicamente la violencia y el crimen.

Si bien todos los ODS están interconectados y se superponen en muchos aspectos, es importante reconocer que cada objetivo tiene su propio enfoque y alcance. En el caso del trabajo de investigación sobre la predicción de delitos, se centra en los ODS 16 y 11, que son los más directamente relacionados con la seguridad y el desarrollo sostenible a nivel local y global.

En resumen, el trabajo de investigación sobre predicción de delitos para reducir la violencia y el crimen se alinea con los Objetivos de Desarrollo Sostenible de las Naciones Unidas, específicamente con el ODS 16 sobre sociedades pacíficas y el ODS 11 sobre ciudades y comunidades sostenibles. Al utilizar enfoques innovadores y tecnología, el estudio busca contribuir a la construcción de un mundo más seguro y sostenible para todos.