Evaluación de modelos profundos en datos de violencia de género

Óscar Sánchez Lima

7 de febrero de 2024

Resumen– Actualmente, la modelización de datos temporales se ha convertido en uno de los grandes logros del aprendizaje computacional, tanto para la predicción de los valores de Bitcoin o Ethereum a lo largo del tiempo como para la previsión del consumo de energía eléctrica en determinadas regiones. Recientemente, se han aplicado herramientas computacionales de modelización de datos en el contexto de las agresiones de género. Este trabajo se centrará en aprender a modelar el comportamiento humano mediante la modelización temporal de datos, utilizando datos de agresiones sexuales en una ciudad durante varios años. Los resultados de este proyecto pueden permitir un mayor conocimiento sobre las dinámicas en el número de denuncias cada año. Todo el código fuente desarrollado durante el proyecto puede consultarse aquí: `https://github.com/oscar-sanchez27/TFG_1527377.git`

Palabras clave– Inteligencia Artificial, Aprendizaje Computacional, Predicción, Violencia de género, Análisis de datos

Abstract– Currently, the modeling of temporal data stands out as a significant achievement in computational learning, serving both the prediction of Bitcoin or Ethereum values over time and the projection of electricity consumption in specific regions. Recently, computational data modeling tools have been applied within the context of gender-based assaults. This research project aims to focus on acquiring expertise in modeling human behavior through temporal data analysis, utilizing datasets pertaining to sexual assaults in a city spanning several years. The outcomes of this endeavor are expected to contribute substantial insights into the dynamics surrounding the annual count of reported incidents. All the source code developed during the project can be consulted here: `https://github.com/oscar-sanchez27/TFG_1527377.git`

Keywords– Artificial Intelligence, Computational Learning, Prediction, Gender-based Violence, Data Analysis

1 Introducción - Contexto del trabajo

La modelización y análisis de datos de series temporales están adquiriendo cada vez más relevancia. Esta disciplina se ha convertido en uno de los objetivos fundamentales del aprendizaje computacional actual.
Para ello se utilizando diferentes técnicas y métodos como modelos profundos. La evaluación de estos en datos relacionados con la violencia de género constituye un paso fundamental en la lucha contra esta problemática que afecta a nuestra sociedad. En la actualidad, la inteligencia artificial y el aprendizaje profundo han emergido como herramientas invaluables para analizar y comprender patrones complejos en estos conjuntos de datos.
El aprendizaje profundo, es una rama de la inteligencia artificial que capacita a las computadoras para procesar datos de manera sofisticada. Los modelos de aprendizaje profundo tienen la capacidad de identificar patrones complejos en imágenes, texto, sonido y otros tipos de información, lo que les permite generar predicciones precisas y obtener conocimientos significativos. La aplicación de métodos de aprendizaje profundo permite la automatización de tareas que an-

• E-mail de contacto: 1527377@uab.cat
• Mención realizada: Computación
• Trabajo tutorizado por: Jordi González Sabaté (Ciencias de la Computación)
• Curso 2023/2024

tes requerían la intervención de la inteligencia humana, como la descripción de imágenes o la transcripción de archivos de audio a texto. Este proyecto se centrará en aprender a modelar el comportamiento humano mediante la modelización temporal de datos. Los datos que se usaran para desarrollar el proyecto serán de agresiones sexuales durante diferentes años.

2 Objetivos

En la actualidad la modelización de datos de series temporales es uno de los aspectos más importantes del aprendizaje computacional. Para este proyecto centrado en modelar el comportamiento humano mediante la modelización temporal de datos, se han establecido diferentes objetivos.

El objetivo principal de este proyecto es el de comprender el funcionamiento y comportamiento de diversos modelos profundos de análisis de datos en diferentes conjuntos de datos temporales. Para lograr lo dicho anteriormente se llevará a cabo una exhaustiva investigación y análisis de modelos como XGBoost, DeepAR y Arima, con el objetivo de determinar su implementación más óptima.

Además de centrarse en el objetivo principal, tendrá otras metas más específicas como por ejemplo el de adquirir nuevos conocimientos y metodologías en el análisis de datos o el de aprender a modelar el comportamiento humano haciendo uso de datos de agresiones sexuales en diferentes lugares. Los lugares escogidos han sido Nueva York, Los Angeles y Chicago. Además, de las mencionadas anteriormente hubo la posibilidad de utilizar una base de datos de Cataluña proporcionada por los Mossos d'Esquadra, pero finalmente no se ha podido conseguir para este proyecto. Finalmente añadir que todo lo hecho y desarrollado en este proyecto podrá ser utilizado por los Mossos d'Esquadra.

3 Estado del Arte

Antes comenzar a hablar sobre en que punto se encuentra el análisis de datos temporales, debemos echar la vista atrás para ver como se llego al punto en el que nos encontramos hoy en día.

El análisis de series temporales tiene depende directamente de la cantidad de datos que haya disponibles, debido a esto no fue hasta el siglo XX donde surgieron numerosos conjuntos de datos consistentes y de alta calidad con el tamaño correspondiente para poder comenzar el análisis de datos. Uno de los modelos en intentar un análisis fue el modelo de autorregresión[1].Pese a ello no fue hasta la década de 1970 que el análisis de series temporales realmente despegó.

Si nos centramos en la previsión de series temporales, que permite aplicar el análisis de series temporales y realizar predicciones, a pesar de lo avanzados que estamos tecnológica y estadísticamente, es bastante difícil hacer cálculos precisos. A día de hoy uno de los modelos que se ha convertido en uno de los principales enfoques para la modelización de series temporales es el modelo ARIMA. Este modelo es un enfoque poderoso y ampliamente utilizado para el pronóstico de series temporales. Es tan destacado debido a que tiene en cuenta tanto las tendencias a largo plazo como las perturbaciones repentinas, pese a que es más óptimo utilizarlo en pronósticos a corto plazo.

Por otra parte también esta el modelo SARIMAX que es una extensión del modelo ARIMA, lo que a diferencia del este último SARIMAX permite modelar tanto la estacionalidad como los factores externos, siendo un modelo mucho más flexible y adecuado para diversos tipos de series temporales[2]. Pese a todo lo anterior, los modelos SARIMAX pueden ser más complicados de estimar e interpretar, a causa de los parámetros adicionales introducidos por los componentes estacionales y exógenos.

4 Planificación

Para llevar a cabo el proyecto se desarrolló una planificación inicial, la cual se ha podido llevar a cabo casi sin ningún percance o restricción. Inicialmente estaba previsto que solo se desarrollasen dos modelos, el DeepAR y ARIMA, pero como al final tuve más tiempo del establecido inicialmente, se añadió otro modelo, el XGBoost. El proyecto se dividió en 8 tareas las cuales estaban formadas por sus correspondientes subtareas. Para cada una de las tareas principales se estableció un periodo de tiempo el cual se ha podido seguir a la perfección sin ningún contratiempo que hiciese alargar el tiempo inicial establecido de cada tarea. Las primeras 2 tareas relacionadas con el análisis de datos, se centraban en la búsqueda de series datos temporales de los 3 lugares mencionados anteriormente y en la preparación de datos. El resto de tareas a excepción de la última se basan en la implementación de los modelos y realización de las correspondientes predicciones. Por último, la tarea final es la del desarrollo del informe, del dossier y la presentación. En la siguiente figura se puede observar el diagrama de Gantt del proyecto.

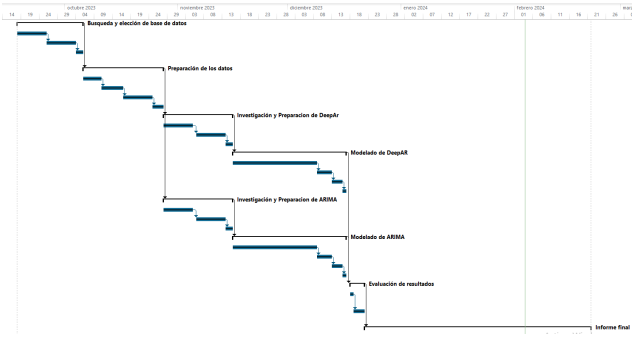

Fig. 1: Diagrama de Gannt

5 Metodologia

Para lograr los objetivos expuestos en el apartado anterior se ha establecido una metodología que permita abordar los diversos problemas para lograr el mejor resultado posible.

En relación con la parte técnica del proyecto, se realizará usando el lenguaje de programación Python. Además, se utilizarán diversas librerías para llevar a cabo el análisis de los datos y la implementación de los modelos correspondientes. Algunas de las librerías son: Numpy, Pandas,

Maltplotlib o Scikit-Learn.

Para comenzar a trabajar en el proyecto, lo primero que habrá que hacer será analizar las tres bases de datos escogidas, para extraer toda la información interesante para una óptima realización del trabajo. Este análisis se hará usando Python y el entorno de desarrollo Visual Studio o Jupyter Notebook.
Una vez completado el análisis de todos los datos y extraída la mejor información de las bases de datos. Se procederá a una investigación exhaustiva sobre los modelos escogidos para posteriormente implementarlos. Para implementarlo de forma que funcione lo mejor posible se necesitará experimentar con los parámetros del modelo mientras se desarrolla. Finalmente, en relación con DeepAR, se analizarán todos los resultados obtenidos.
Al acabar con DeepAR se utilizará el mismo procedimiento para el otro modelo llamado ARIMA.
Para finalizar el trabajo se cogerán todos los datos obtenidos por los modelos en diferentes bases de datos y se hará una evaluación comparativa entre ellos.
Ahora explicaré de forma detallada la base teórica de los modelos que se han utilizado para llevar a cabo los objetivos marcados para este trabajo.

5.1. XGBoost

El modelo XGBoost (Extreme Gradient Boosting) es un algoritmo de aprendizaje automático basado en arboles de decisión con aumento de gradiente. Este aumento significa que los modelos que son recién añadidos se crean para poder corregir los errores de los modelos anteriores, combinándolos posteriormente para crear la predicción final[3].Es decir, este técnica se basa en la combinación y entrenamiento de modelos individuales para obtener una única predicción. XGBoost minimiza una función regularizada(L1 y L2) que combina una función de pérdida convexa, que se basa en las diferencias entre las salidas previstas y el objetivo. El entrenamiento de este sucede de forma iterativa agregando nuevos árboles que predicen los residuos o error de árboles anteriores que finalmente de combinaran para realizar la predicción final.
Para entrenar el modelo necesitamos una función objetivo para poder medir que tan bien se ajusta el modelo a los datos de entrenamiento[4].

$$obj(\theta) = L(\theta) + \omega(\theta) \tag{1}$$

Generalmente un árbol solo no es lo suficientemente robusto, por lo que realmente se utiliza es un modelo de conjunto. Entonces si se considera un conjunto de datos D=(xi, yi), donde n es el número de ejemplos y m es el número de atributos, la predicción del conjunto se puede expresar como:

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{2}$$

donde K es la cantidad de árboles, f_k es una función en el espacio funcional F y F es el conjunto de todos los CARTS(conjunto de árboles de clasificación y regresión) posibles. La función objetivo a optimizar se define:

$$obj(\theta) = \sum_{i}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k) \tag{3}$$

donde ω es la función de regularización que se utiliza para controlar la complejidad del árbol y se define como:

$$\omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2 \tag{4}$$

Finalmente en la siguiente imagen se puede ver el funcionamiento:

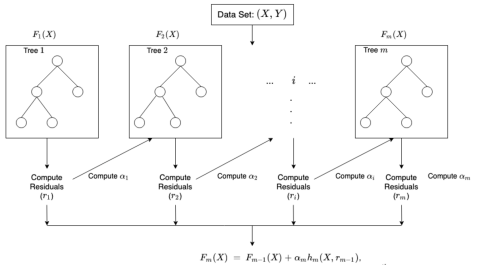

Fig. 2: Diagrama XGBoost[5]

5.2. ARIMA

Un modelo ARIMA(Autoregressive Integrated Moving Average) es una forma de análisis de regresión que se encarga de medir la fuerza de una variable dependiente en relación a otras que no lo son. El objetivo de este modelo es predecir movimientos futuros examinando las diferencias entre los valores de la serie temporal[6].
Para entender mejor un modelo ARIMA se puede hablar de sus 3 componentes principales, que son los siguientes:

- Autoregresion (AR): se refiere al modelo que muestra una variable cambiante que retrocede según sus propios valores anteriores.
- Integrado (I): representa la diferenciación de observaciones sin procesar. Esto permite que la serie se vuelva estacionaria.
- Media móvil (MA): Incorpora la dependencia entre una observación y el error residual de un modelo.

Un modelo ARIMA se podría comparar con un filtro que trata de distinguir la información valiosa de la interferencia del ruido. Luego, esta información valiosa se proyecta hacia el futuro para generar predicciones[7].
Un modelo ARIMA no estacional se clasifica como modelo $ARIMA(p, d, q)$, donde:

- p es el número de términos autorregresivos.
- d es el número de diferencias no estacionales necesarias para la estacionalidad.
- q es el número de errores de pronóstico rezagados en la ecuación de predicción.

La ecuación de pronóstico ARIMA para una serie temporal estacionaria es una ecuación lineal y se representar de la siguiente forma:
Valor previsto de Y = una constante y/o una suma ponderada de uno o más valores recientes de Y y/o una suma ponderada de uno o más valores recientes de los errores.
Además, se construye de la siguiente manera. Primero, sea y la d-ésima diferencia de Y , lo que significa:

- si d = 0, entonces $y_t = Y_t$
- si d = 1, entonces $y_t = Y_t - Y_{t-1}$
- si d = 2, entonces $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

En términos de y , la ecuación general de pronóstico es:

$$\hat{Y}_t = \mu + \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q} \quad (5)$$

5.3. DeepAR

El modelo DeepAR es un tipo de modelo de aprendizaje profundo que esta diseñado específicamente para identificar las incertidumbres asociadas a las predicciones futuras. A diferencia de otros modelos, DeepAR proporciona una distribución de probabilidad sobre valores futuros lo que permite evaluar el rango de resultados posibles y tomar decisiones más informadas[8].DeepAR es una red neuronal recurrente basada en LSTM que se entrena con los datos históricos de las series temporales del conjunto de datos. Al entrenar en múltiples series de tiempo simultáneamente, el modelo DeepAR aprende el comportamiento complejo y dependiente del grupo entre las series de tiempo que a menudo conduce a un mejor rendimiento que los métodos estándar ARIMA y ETS[9].

Algunos de los principios fundamentales de este modelo son:

- Arquitectura Autoregresiva: Al emplear asta arquitectura las predicciones dependen de una combinación de observaciones históricas y las predicciones pasadas del propio modelo para cada paso de tiempo. Esto permite que el algoritmo capture dependencias más complejas.
- Incorporación de características categóricas: La inclusión de tales características mejora la capacidad del modelo para discernir patrones y relaciones dentro de los datos.
- Mecanismo de atención temporal: Este mecanismo lo que permite es que el modelo se puede centrar en las partes relevantes de la serie temporal, adaptando su atención dinámicamente.
- Entrenamiento con pérdida de cuantiles: Esto implica que el modelo está diseñado para producir intervalos de predicción, los cuales indican el espectro de valores futuros posibles junto con niveles de confianza asociados.

Si hablamos del funcionamiento del modelo, este para facilitar el aprendizaje de patrones, crea automáticamente series de tiempo de características basadas en la frecuencia de la serie de tiempo objetivo.
DeepAR entrena un modelo muestreando de forma aleatoria diversos ejemplos de entrenamiento de cada una de las series temporales en el conjunto de datos de entrenamiento. Cada ejemplo de entrenamiento consta de un par de ventanas de contexto y predicción adyacentes con longitudes fijas predefinidas.Durante el entrenamiento, el algoritmo ignora los elementos del conjunto de entrenamiento que contienen series temporales que son más cortas que una longitud de predicción especificada[10].

Finalmente centrándonos en su arquitectura, el modelo utiliza una arquitectura RNN que incorpora una probabilidad binomial gaussiana/negativa para producir pronósticos probabilísticos y supera el pronóstico tradicional de un solo elemento. En la siguiente figura se puede observar la arquitectura que utiliza para entrenamiento y predicción[11]:

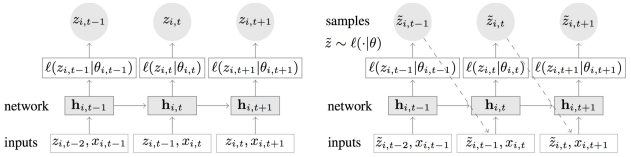

Fig. 3: Arquitectura DeepAR

5.4. Métricas de evaluación

Al embarcarnos en el desarrollo e implementación de nuestros modelos de machine learning, es esencial explorar métricas que faciliten la mejora del rendimiento.

La obtención de precisión en los datos de entrenamiento es esencial; sin embargo, es igualmente crucial que dicho resultado sea confiable y generalizable a datos desconocidos. Esto es particularmente relevante cuando los algoritmos no pueden alcanzar una eficiencia del 100 %, ya que podrían sesgarse y presentar sobreajuste [12].
Existen diversas métricas para conocer el error, de las cuales he escogido las que considero más importantes.

5.4.1. Error Absoluto Medio (MAE)

En los campos de la estadística y el aprendizaje automático es una métrica utilizada con frecuencia. Esta es una métrica muy simple que calcula la diferencia absoluta entre los valores reales y previstos.Se puede calcular de la siguiente manera:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (6)$$

Donde y_i es el valor real,$\hat{y}_i$ es el valor previsto y n es igual tamaño de la muestra.

5.4.2. Error Cuadrático Medio (MSE)

Mide la raíz cuadrada de las discrepancias promedio entre los valores reales de un conjunto de datos y los valores proyectados.Se utiliza para evaluar qué tan bien funcionan los modelos predictivos[13].La fórmula es la siguiente:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (7)$$

5.4.3. Raíz del Error Cuadrático Medio (RMSE)

El RMSE esta directamente relacionado con el MSE ya que simplemente es la raíz cuadrada de este. RMSE cuantifica qué tan bien se alinean los valores predichos de un modelo con los valores reales observados en el conjunto de

datos[14].Se calcula de la siguiente manera:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

5.4.4. Porcentaje de Error Medio Absoluto (MAPE)

Calcule el error porcentual absoluto medio (MAPE) dividiendo la diferencia absoluta entre los valores reales y previstos por el valor real. Este porcentaje absoluto se promedia en todo el conjunto de datos[15].

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i} * 100| \tag{9}$$

5.4.5. Error porcentual absoluto medio simétrico (SMAPE)

es una de las métricas de error más controvertidas, ya que existen diferentes definiciones y fórmulas disponibles. Algunos críticos sostienen que esta métrica no es verdaderamente simétrica, a pesar de su nombre. Se puede calcular siguiendo la siguiente formula[16]:

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \tag{10}$$

6 Desarrollo del Trabajo

El trabajo como ya se ha explicado anteriormente se divide en dos grandes bloques. El primer bloque que es el de análisis de datos y el segundo bloque que es el de investigación e implementación de los modelos. Centrándonos en el primer bloque, este se basa en analizar el conjunto de datos temporales de las diferentes bases de datos que nos servirá para entender mejor el conjunto de datos escogido, y para sacar las primeras conclusiones de estos. Todo ello para posteriormente utilizarlos de forma óptima en la implementación de los modelos. En referencia al segundo bloque será la parte de implementación de los modelos indicados en el apartado anterior, el cual se utilizará para extraer los resultados y conclusiones del proyecto. A continuación se explicará con más detalle estos dos partes en las que se divide el trabajo.

6.1. Búsqueda y elección de las bases de datos

En esta tarea inicial había que buscar y elegir cuáles iban a ser las bases de datos utilizadas para posteriormente hacer las predicciones. Para buscar estas bases de datos se ha utilizado la plataforma Kaggle. En esta plataforma habían diferentes posibilidades para escoger, pero uno de los factores que tenían que tener estas bases de datos es que tenían que contener datos de agresiones sexuales. Lo primero que se ha hecho ha sido filtrar todas las bases de datos encontradas que tenían relación con crímenes. Es decir que sus datos nos diesen la información sobre el tipo de crimen, la fecha que ocurrió y varios aspectos más que serán importantes para poder entender la información. Una vez se tiene un grupo más reducido, se ha ido una por una mirando cuáles podían ser las más óptimas teniendo en cuenta diferentes factores como el tamaño de los datos y la información de estos. Los factores más importantes han sido que tuvieran datos de agresión sexual, que se indicase la fecha de los crímenes o que nos ofrezcan información de las víctimas.

Finalmente, las bases de datos seleccionadas han sido las siguientes:

1. Chicago Crimes[17]
2. New York Crimes[18]
3. Los Angeles Crimes[19]

La base de datos de Chicago contiene una amplia serie de datos, en concreto 43 mil agresiones de violencia de género reportadas que ocurrieron en la ciudad de Chicago desde el año 2001 hasta el año 2018.
En concreto este dataset muestra datos sobre el tipo de delito que es, la fecha en la que ocurrió, la localización exacta y en que espacio ocurrió, si fue en una casa, apartamento o en la calle. Además, nos ofrece datos más detallados como por ejemplo de sí el delito fue doméstico o no, o si fueron arrestados o no los delincuentes.

Siguiendo con la base de datos de Nueva York, esta contiene un conjunto de datos sobre los delitos reportados en la ciudad de Nueva York desde 1965 al 2018. En relación con los delitos contiene información como el tipo de delito reportado, la fecha en la que ocurrió el delito o en el que fue reportado, en que lugar sucedió y diversos datos sobre la víctima, entre los cuales se pueden destacar el genero, la edad y etnia.
Como se puede observar este no aposta una significativa cantidad de datos sobre todo lo relacionado con los delitos que aparecen en el dataset. Todo esto nos puede servir para poder entender mejor el conjunto de datos y poder tener en cuenta estos aspectos para cuando se tenga que realizar las predicciones.

Por último el tercer conjunto de datos contiene un total de 27 mil agresiones ocurridas en Los Angeles. Al igual que los datasets mencionados anteriormente, este también contiene diversas informaciones sobre los delitos reportados en esta zona durante los años 2010 al 2019. Añadir también que nos da información sobre la víctima. Por último, nos informa sobre la fecha, la localización y si se usó arma o no, entre otras muchas cosas.

6.2. Preparación de los datos

En esta tarea el objetivo principal ha sido el de analizar en profundidad los 3 conjuntos de datos escogidos anteriormente. Además de hacer una limpieza en ellos previa al análisis. Para esta limpieza de datos se han seguido diferentes procedimientos. El primero de todos ha sido el de filtrar solamente los datos que son necesarios para nuestro proyecto, es decir, de todos los datos que nos proporcionaban las bases de datos se ha escogido solo aquellos datos que estaban relacionados con agresiones sexuales. Seguido de eso, se han detectado y eliminado los valores nulos. Finalmente, en relación con la limpieza de datos se han eliminado los valores que no son representativos para entrenar el proyecto.
Una vez listo los dataset, se ha comenzado a analizar los diferentes conjuntos de datos en profundidad para poder entenderlos y lo máximo posible.

6.2.1. Chicago Crimes

Primero de todo se ha comenzado por el de Chicago. En este dataset se ha analizado diferentes aspectos, como el porcentaje de casos que eran domésticos o no por año, y lo se ha podido observar es que en todos los años el porcentaje de crímenes domésticos han sido superiores a los no domésticos, como se puede observar en la figura siguiente:

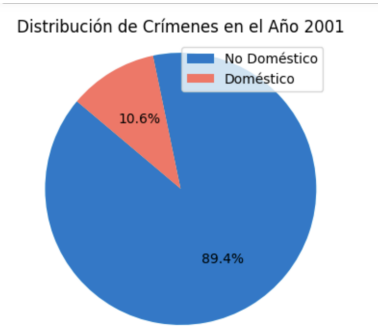

Fig. 4: Ejemplo gráfica 2001

Aparte, también se ha analizado las agresiones que ha habido por año como se muestra en la siguiente figura:

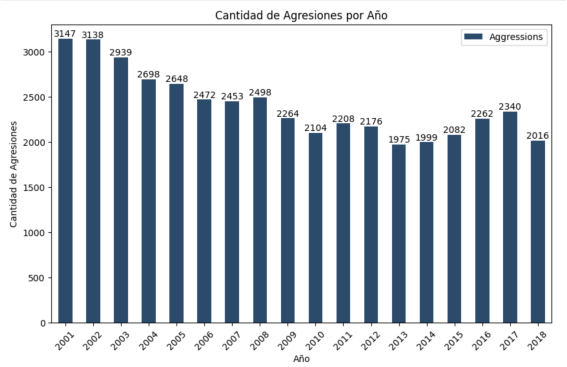

Fig. 5: Cantidad de agresiones por año

Como se puede apreciar en la figura anterior se puede ver que hay un pico de agresiones al inicio de cada año, por lo que podemos entender que el día de fin de año se producen bastantes agresiones en comparación con el resto. Además, también podemos ver que durante los meses de verano las agresiones crecen en comparación a los otros meses.

6.2.2. New York Crimes

Una vez analizado el de Chicago se ha analizado el dataset referente a Nueva York. En este dataset se han dividido en 2 tipos de crímenes, crimen de violación o crimen sexual. Lo primero de todo a analizar ha sido hacer una comparativa entre ellos dos para ver con que frecuencia sucedía y ver el porcentaje por año. Se ha analizado a las víctimas de las agresiones en profundidad, tanto el género más afectado como el rango de edad o la etnia. Además de analizar las víctimas, también se ha hecho un análisis de las agresiones como el lugar donde se produje la agresión, el numero de agresiones por año, entre otros diversos factores. Como se pude apreciar en siguiente mapa:

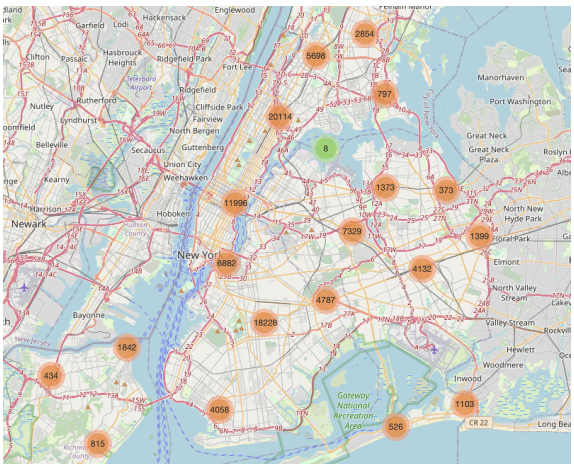

Fig. 6: Mapa de agresiones Nueva York

Se ha podido apreciar que desde 1965 a 1997 no hay muchos registros por lo que se entiendo como que antes no se reportaban estos delitos, ya que como maximo hay 60 a diferencia de años más actuales. Esto se puede apreciar en la gráfica siguiente:

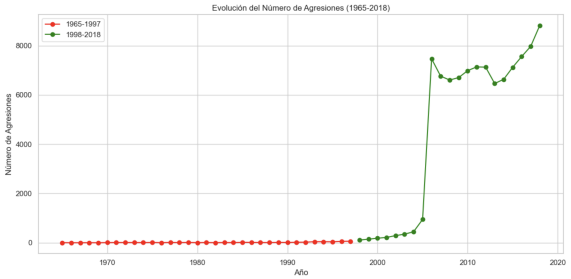

Fig. 7: Evolución de las agresiones

6.2.3. Los Angeles Crimes

Por último se ha analizado el dataset de Los Angeles y al igual que el resto se ha estudiado las agresiones que aparecían en el conjunto de datos y analizar a las víctimas a fondo. Para ello se han extraído gráficas y mapas para poder entender mejor los datos con los que estamos trabajando y que posteriormente se utilizaran en los modelos.Algunas de las gráficas más interesantes son estas:

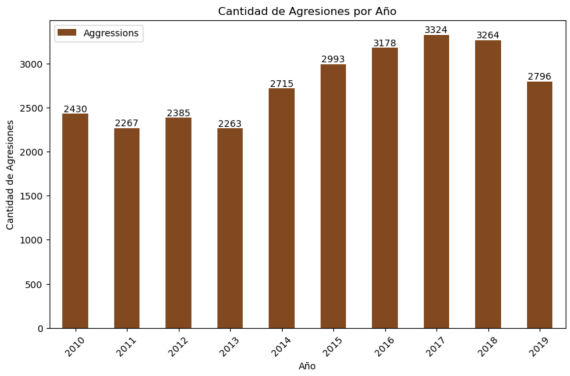

Fig. 8: Número de agresiones en Los Angeles

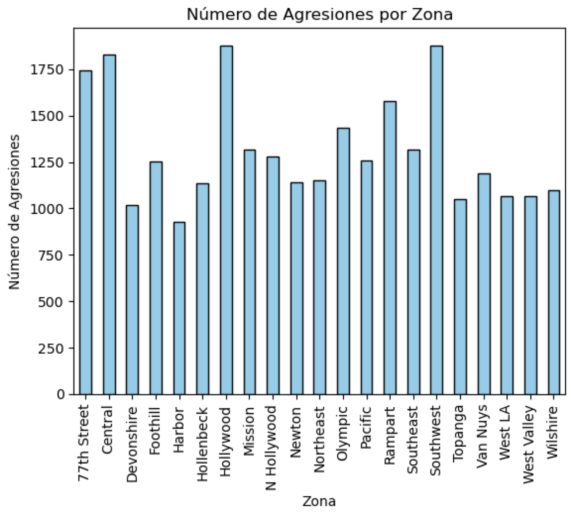

Fig. 9: Número de agresiones en Los Angeles por zona

Además se ha hecho un mapa para visualizar en que zonas de los angeles se han cometido los delitos.

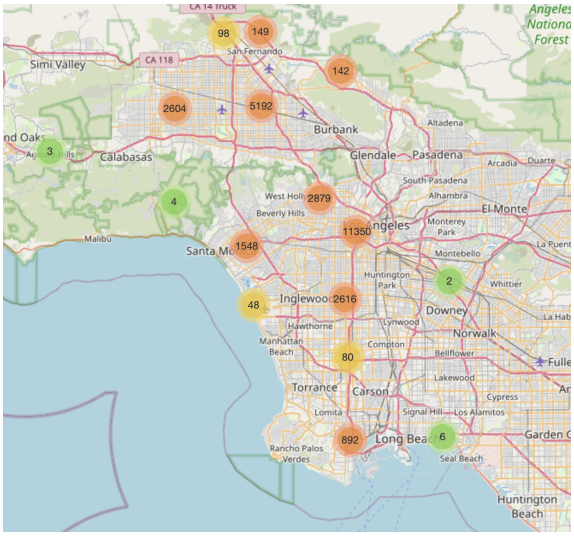

Fig. 10: Mapa de Los Angeles

6.3. Implementación de los modelos

Finalmente, comenzamos la última parte del proyecto y mas importarte que la implementación de los modelos escogidos. Para su implementación primero se ha hecho una investigación para aprender sobre estos y adquirir todo el conocimiento posible para poder sacarles el mejor rendimiento. Primero de todo se ha empezado a implementar el DeepAR, seguido de ARIMA y finalmente XGBoost.

6.3.1. XGBoost

Para la implementación del modelo, primero de todo se han seguido una serie de pasos previos al entrenamiento de esto. Para comenzar se han importado tanto las librerías necesarias como los datasets específicos para la predicción. Unos vez importado todo lo necesario, se ha divido el dataset en train y test, el train para la parte del entrenamiento y el test para la validación. Esta división ha sido posible utilizando la función $train\_test\_split$ de la librería sklearn. Finalizado la configuración previa y con el dataset dividido, se ha comenzado a entrenar el modelo.
Para el entrenamiento del modelo XGBoost se ha hecho uso de la librería $xgboost$ y la función $XGBRegressor()$ para poder realizar el entrenamiento del modelo. Para esta función ha sido necesario emplear una serie de hiperparámetros para poder conseguir el mayor rendimiento posible del modelo. Estos hiperparámetros[20]son los que aparecen en la siguiente tabla:

Nombre	Definición	Valor
n_estimators	Número de árboles que se deben construir.	500
learning_rate	Tasa de aprendizaje	0.01
max_depth	La profundidad máxima de un árbol. Se utiliza para controlar el sobreajuste.	5
subsample	Fracción de observaciones que serán muestras aleatorias para cada árbol.	0.5
colsample_bytree	Fracción de columnas que serán muestras aleatorias para cada árbol.	0.8
colsample_bylevel	Proporción de submuestra de columnas para cada división en cada nivel	1
random_state	Establecer una semilla para la generación de números aleatorios	0

TABLA 1: TABLA DE HIPERPARÁMETROS XGBOOST

Después de aplicar todo lo dicho anteriormente se han podido conseguir resultados como los de la siguiente gráfica:

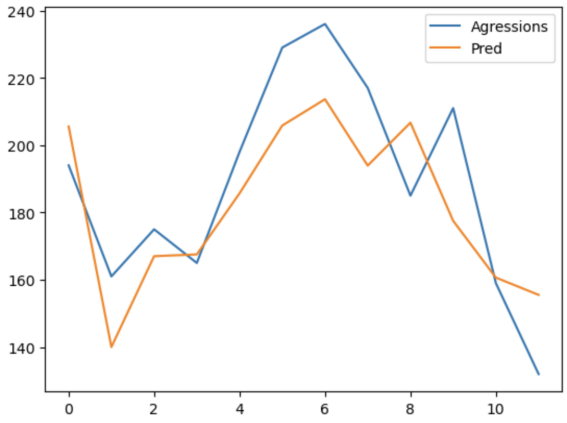

Fig. 11: Predicción año 2016

6.3.2. ARIMA

Para el modelo ARIMA, se han seguido los mismos pasos previos que el modelo anterior XGBoost. Primero de todo se han importado los datasets y librerías necesarias. Una

vez importado todo lo necesario, se ha empezado a realizar el entrenamiento. Previo al entrenamiento se ha separado el conjunto de datos en datos de entrenamiento y datos de test. Una vez separado, el primer paso a seguir para entrenar el modelo ha sido el de ha sido el de crear un historial que contiene las observaciones de la serie temporal de entrenamiento. Esto se utiliza para inicializar el modelo ARIMA con datos históricos. Después se ha definido el modelo ARIMA usando como parámetros el historial y el orden $(1, 1, 0)$. El orden empleado nos indica que tal y como esta explicado en el apartado de metodología en la sección de ARIMA, significa que el modelo tiene un componente autoregresivo de orden 1 (p=1), un componente de diferenciación de orden 1 (d=1), y ningún componente de media móvil (q=0).
Realizado esto se ha procedido a entrenar el modelo, el cual nos ha proporcionado resultados como el siguiente:

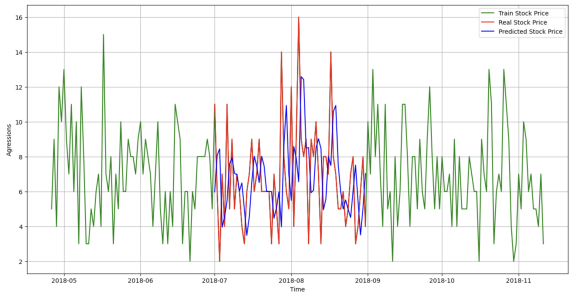

Fig. 12: Predicción verano 2018

6.3.3. DeepAR

Este modelo al igual que el resto de modelo se ha hecho una configuración previa. Primero se han importado las librerías y datasets necesarios para la realización óptima del modelo. Una vez importado todo y siguiendo los mismos pasos que los anteriores modelos se ha separado el conjunto de datos, en datos de entrenamiento y datos de test. Una vez hecho todos estos pasos previos, a diferencia de el resto de modelos en este se ha necesitado diferenciar las columnas, es decir, separar las $keys$ en los grupos de características categóricas dinámicas,categóricas estáticas, reales dinámicas y reales estáticas.
Una vez realizada esta asignación se ha procedido a desarrollar la parte de definición y entrenamiento del modelo.Para definir el modelo se ha hecho uso de los siguientes hiperparámetros[21]:

Nombre	**Definición**	**Valor**
freq	Frecuencia de serie temporal	'd'
context_length	Número de puntos de tiempo que el modelo ve antes de realizar la predicción	45
prediction_length	Número de pasos de tiempo que el modelo está entrenado para predecir	61
num_layers	Número de capas ocultas en el RNN	2
num_cells	Cantidad de celdas que se usarán en cada capa oculta del RNN	40
cell_type	Tipo de celda recurrente	'gru'
epochs	Número máximo de pasadas sobre los datos de entrenamiento	20
cardinality	Matriz que especifica el número de categorías por característica categórica.	lista

TABLA 2: TABLA DE HIPERPARÁMETROS DEEPAR

Una vez definido el modelo con los hiperparámetros más óptimos se ha entrenado y obtenido la predicción junto a los valores de las métricas de evaluación. Algún resultado de los obtenidos es el siguiente:

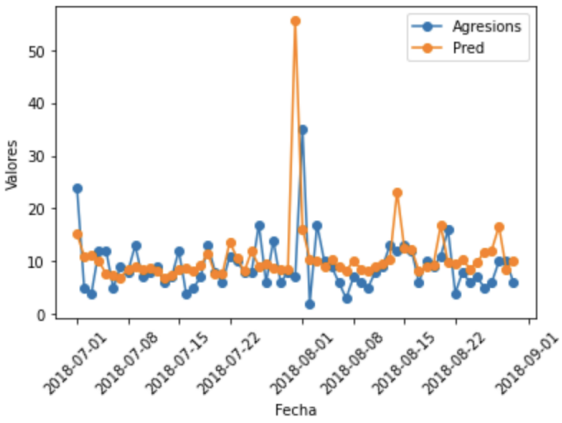

Fig. 13: Predicción verano 2018

7 COMPARACIÓN DE RESULTADOS

Después de haber implementado todos los modelos y realizar los mejores entrenamientos posibles empleando los hiperparámetros correspondientes para cada modelo explicados en la sección anterior, se ha llegado a los siguientes resultados en las diferentes bases de datos.
Para la base de datos de Chicago se ha extraído el siguiente resultado:

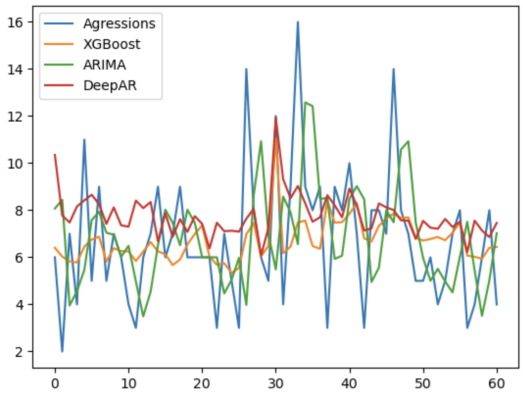

Fig. 14: Resultados Chicago 2018

Modelo	MAE	RMSE	MAPE	sMAPE
XGBoost	**1.84**	**2.50**	**34.14**	**28.24**
DeepAR	2.14	2.82	45.64	32.21
ARIMA	2.72	3.44	47.80	39.88

TABLA 3: RESULTADOS DE LOS MODELOS CHICAGO

Como se puede observar en tanto en la gráfica como en los valores de las diferentes métricas de evaluación de los modelos, el modelo que ha funcionado mejor en el conjunto de datos de Chicago ha sido XGBoost.
Para la base de datos de Nueva York se ha extraído el siguiente resultado:

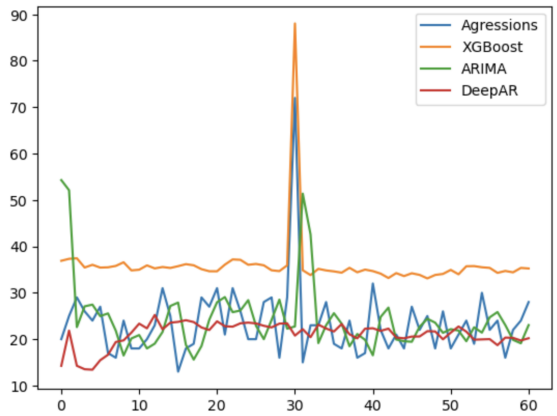

Fig. 15: Resultados Nueva York 2018

Modelo	MAE	RMSE	MAPE	sMAPE
XGBoost	12.57	13.40	61.68	44.48
DeepAR	**5.62**	**8.73**	**22.47**	**23.74**
ARIMA	8.25	14.26	33.03	28.97

TABLA 4: RESULTADOS DE LOS MODELOS NUEVA YORK

Si nos centramos en los resultados obtenidos en el conjunto de datos de Nueva York, observando las métricas se puede extraer que el modelo más óptimo para esta base de datos ha sido el modelo DeepAR. Finalmente para la base de datos de Los Angeles el resultado ha sido el siguiente:

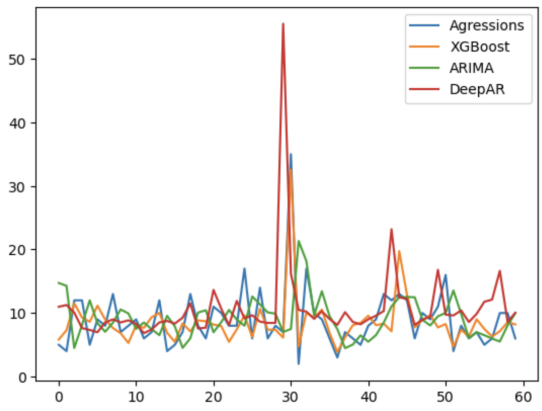

Fig. 16: Resultados Los Angeles 2018

Modelo	MAE	RMSE	MAPE	sMAPE
XGBoost	**2.21**	**2.88**	**27.18**	**25.75**
DeepAR	4.23	7.79	59.86	46.39
ARIMA	4.07	6.28	62.48	41.59

TABLA 5: RESULTADOS DE LOS MODELOS LOS ANGELES

Para acabar si tenemos en cuenta los resultados obtenidos de esta serie de datos temporales de Los Angeles podemos observar e identificar que el modelo que mejor a rendido ha sido el XGBoost.

Si nos fijamos bien en los resultados obtenidos por las tablas podemos sacar las siguientes conclusiones en referencia a las métricas de evaluación:

- SMAPE, nos enfrentamos al problema de las penalizaciones por subestimación. Las predicciones de los modelos muestran una tendencia a subestimar las agresiones, lo cual se refleja en errores muy altos.
- El MSE es una medida más precisa que el MAE, ya que considera la magnitud de los errores, mientras que el MAE es ineficiente para valores extremos. Por eso los errores del MSE y RMSE son más amplios.

8 CONCLUSIONES

Como conclusión a los resultados obtenidos en referencia a los modelos puedo decir que el XGBoost ha sido el qeu mejor rendimiento ha tenido respecto a los otros dos, ya que este suele tener un excelente desempeño en detectar patrones i relaciones no lineales en los diferentes conjuntos de datos.Todo esto se debe a que XGBoost tiene una alta capacidad para gestionar grandes conjuntos de datos y características más elaboradas, como podría ser la base de datos de Chicago.Pese a que en el conjunto de datos de Nueva York no ha salido como se esperaba debido a la baja cantidad de datos. Centrándonos en el modelo DeepAR los resultados no han sido los que se esperaban de este, pese a que DeepAR tiene la capacidad de identificar diferentes patrones a corto plazo. Si es verdad que hay una notable diferencia en este modelo cuando se usan fechas diarias a cuando se han utilizado fechas que agrupan un conjunto de fechas, como podría ser una base de datos con fechas mensuales. La razón de ello es que en los datos diarios haya patrones más pronunciados. Por último, en relación a ARIMA podemos decir que en las diferentes bases de datos no ha llegado a tener el mejor rendimiento de los tres pero tampoco ha sido el peor, esto se atribuye a que es sensible a la calidad de los datos y la selección adecuada de los hiperparámetros. Además, ARIMA tiende a funcionar mejor en series temporales estacionarias y puede enfrentar desafíos en la modelización de tendencias complejas o patrones no lineales. El desempeño de ARIMA puede variar significativamente según el contexto de la aplicación y la calidad de los datos disponibles.
En referencia a trabajos a futuro creo que podrían ser los siguientes:

- Mejorar el DeepAR si es posible conseguir una base de datos de calidad y con un gran tamaño de datos, ya que este modelo puede ofrecer muy buen rendimiento.

- Intentar implementar el modelo SARIMAX. Este es un modelo mucho más flexible que el modelo ARIMA, ya que puede modelar tanto la estacionalidad como los factores externos.

AGRADECIMIENTOS

Para finalizar, me gustaría dedicar unas palabras de agradecimiento a mi tutor Jordi González Sabaté por haberme ayudado durante el desarrollo de todo este bonito proyecto, poniéndome las cosas más fáciles y ayudándome en todos los contratiempos que han podido surgir. Este trabajo no habría sido posible sin él. Por último también agradecer también la ayuda a Jordi Pons que me ha mantenido informado de todas las entregas y plazos, además de resolver alguna duda cuando se necesitaba.

REFERENCIAS

[1] Macias, D. A. (2022, 14 agosto). A brief history on time series analysis forecasting. Medium. https://medium.com/@deniseamacias1/a-brief-history-on-time-series-analysis-forecasting-f5a22bbd0641

[2] Ambika. (2023, 14 agosto). Time Series Analysis and Forecasting: Definition, types, techniques. Medium. https://medium.com/@ambika199820/time-series-analysis-and-forecasting-definition-types-techniques-f8e75192d992

[3] Wirth, S. (2023, 18 abril). XGBOOST: Theory and Application - Hoyalytics - Medium. Medium. https://medium.com/hoyalytics/xgboost-theory-and-application-4801a5dba4fb

[4] Introduction to boosted Trees — XGBoost 2.0.3 documentation. (s. f.). https://xgboost.readthedocs.io/en/stable/tutorials/model.html

[5] How XGBoost Works - Amazon SageMaker. (s. f.). https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html

[6] Hayes, A. (2023, 29 septiembre). Autoregressive Integrated Moving Average (ARIMA) Prediction Model. Investopedia. https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

[7] ARIMA models for time series forecasting . (s. f.) https://people.duke.edu/~rnau/411arim.htm

[8] GeeksforGeeks. (2024, 6 enero). DeepAR Forecasting Algorithm. https://www.geeksforgeeks.org/deepar-forecasting-algorithm/

[9] Maklin, C. (2022, 15 julio). DeepAR Forecasting Algorithm - Cory Maklin - Medium. Medium. https://medium.com/@corymaklin/deepar-forecasting-algorithm-6555efa63444

[10] How the DeepAR algorithm works - Amazon SageMaker. (s. f.). https://docs.aws.amazon.com/sagemaker/latest/dg/deepar_how-it-works.html

[11] Arrigoni, A. (2021, 6 diciembre). Paper review & Code: Amazon DEEPAR - Alberto Arrigoni - Medium. Medium. https://medium.com/@albertoarrigoni/paper-review-code-amazon-deepar-809938a319d9

[12] Agrawal, R. (2023, 9 octubre). Know the best evaluation metrics for your regression model! Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/know-the-best-evaluation-metrics-for-your-regression-model/

[13] GeeksforGeeks. (2023, 12 octubre). Regression metrics. https://geeksforgeeks.org/regression-metrics/

[14] A comprehensive overview of regression evaluation metrics — NVIDIA Technical blog. (2023, 11 julio). NVIDIA Technical Blog. https://developer.nvidia.com/blog/a-comprehensive-overview-of-regression-evaluation-metrics/

[15] M, P. (2023, 30 noviembre). A comprehensive introduction to evaluating regression models. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression models/: :text=Calculate %20Mean %20Absolute %20Percentage %20Error,) %2C %20increases %20linearly %20with %20error.

[16] Goodwin, P., Lawton, R. (1999b). On the asymmetry of the symmetric MAPE. International Journal of Forecasting, 15(4), 405-408. https://doi.org/10.1016/s0169-2070(99)00007-2

[17] Chicago Crimes 2001-2018 (November). (2018, 24 noviembre). Kaggle. https://www.kaggle.com/datasets/spirospolitis/chicago-crimes-20012018-november

[18] New York City Police crime data historic. (2020, 12 julio). Kaggle. https://www.kaggle.com/datasets/mrmorj/new-york-city-police-crime-data-historic

[19] LA Crime data. (2023, 29 abril). Kaggle. https://www.kaggle.com/datasets/chaitanyakck/crime-data-from-2020-to-present

[20] Jain, A. (2024, 7 enero). Mastering XGBoost Parameters Tuning: A complete guide with Python codes. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

[21] DeepAR Hyperparameters - Amazon SageMaker. (s. f.). https://docs.aws.amazon.com/sagemaker/latest/dg/deepar_hyperparameters.html