MÁSTER EN BIG DATA & BUSINESS ANALYTICS PRESENCIAL

TRABAJO DE FIN DE MASTER

PREDICCIÓN DE ACTOS DELICTIVOS EN LA CIUDAD DE CHICAGO

AUTORES:

Carlos Astorga Serrano
Ramón Denia Chávarri
Noelia García García

Tutores:

Prof.D. José Luis Llorente (parte académica)

Madrid, 2020

RESUMEN

El presente trabajo de fin de máster responde a la necesidad de controlar el crimen en la ciudad estadounidense de Chicago. Con la intención, de lograr una ciudad más segura y justa, gracias al procesamiento de datos que proveen las técnicas de ***Big Data***. Como solución, se intentará implementar un modelo de **Machine Learning** que proporcione una predicción del crimen, lo más ajustada posible para los próximos años.

Los datos utilizados para la implementación del proyecto, han sido los registros con los que cuenta la policía de chicago, que se comparte como información pública a través de una API. Esta información, esta contrastada por organismos fiables estadounidenses. Por ello, no se ha requerido ningún acuerdo de confidencialidad.

Para intentar alcanzar el objetivo, se han usado una serie de algoritmos centrados en modelos de aprendizaje supervisados que dependen de una variable objetivo. La variable seleccionada por sus características, es el distrito donde se cometen los crímenes. Con la intención de adelantarnos a sucesos o acciones realizadas en una zona determinada.

ABSTRACT

The present end of master project answers the necessity of criminal control in the city of Chicago, with the intention of achieving a safer and fairer city, and the help of data processing provided by Big Data techniques. The solution given is trying to implement a Machine Learning model that provides a crime prediction, as tight as possible, for the following years.

The data used for the implementation of the Project has been the Chicago police records, which are shared as public information trough an API. This information is verified by reliable USA agencies. Therefore, no confidentiality agreement has been required.

In order to achieve the goal, a series of algorithms focused on supervised learning models which depend on a target variable have been used. The variable selected for its characteristics, is the district where the crimes are committed. The main goal is anticipating events or actions carried out in a specific area.

ÍNDICE

ÍNDICE DE ILUSTRACIONES

ABREVIATURAS

CRISP-DM: Cross Industry Standard Processfor Data Mining

KDD: Knowledge Discovery in Databases

SEMMA: Sample, Explore, Modify, Model and Assess

API: ApplicationProgramming Interface

CLEAR: CitizenLawEnforcementAnalysis and Reporting

MSE: Error Cuadrático de la Media

MEA: Error Absoluto de la Media

R2: Error Cuadrado

DAX:Data AnalysisExpressions

I. PARTE INTRODUCTORIA

1. Introducción

En los últimos años con el nacimiento de las nuevas tecnologías, la inteligencia artificial y el desarrollo exponencial del ***Big Data*** hemos asistido a un cambio significativo a la hora de poder gestionar, tomar decisiones, ser más rápidos y sobre todo lograr objetivos de forma eficaz y con un menor coste. Los ***datos*** se han convertido en el nuevo ***petróleo***, y quien es capaz de disponer de ellos es quien tiene poder.

Esta revolución en el mundo del dato se debe principalmente al ***progreso tecnológico*** que ha traído consigo, un aumento sin precedentes del número de fuentes de origen y por supuesto, el desarrollo de software de las últimas décadas que ha aportado los instrumentos necesarios para poder trabajar con tanto ***volumen***. El Big Data, ha permitido la posibilidad de ser más eficientes y desarrollar cualquier tarea dentro de un sinfín de áreas completamente heterogéneas como pueden ser la salud, las finanzas, el deporte, los transportes, etc.

En este caso, se puede considerar de gran ayuda aplicar estas tecnologías con el fin de aportar nuestro granito de arena al ámbito de la administración, en concreto, cercando el estudio en la creación de un ***modelo*** capaz de predecir los ***delitos*** y ayudar así a una mejora en la gestión y planificación de activos policiales. Un estudio del pasado realizado por la compañía española ***Synergic Partners*** en colaboración con la universidad de Columbia y el ayuntamiento de la ciudad de Nueva York consiguió ***reducir*** el crimen de manera considerable, con la intención de mejorar su puesto en la lista de las ciudades más peligrosas.

El modelo de la investigación logró ***predecir*** el crimen en un 72% de las ocasiones, así como un nivel de acierto del 83% de los asesinatos. Se pudo observar cómo había una ingente cantidad de delitos cometidos en la vía pública que no se tomaban en cuenta y ocurrían a plena luz del día, en concreto un 39,8%. También, determinados factores o variables que se consideraban importantes al final no tenían tanta influencia en el modelo, como es la hora a la que se cometían la mayor parte de los delitos. Estas acciones no eran por la noche si no en la franja horaria de 15:00 a 19:00.

Por todo esto, se piensa que podría ser de gran ayuda construir un modelo similar para la ciudad de ***Chicago,*** no sólo por ayudar a tener una mejor convivencia y lograr

reducir el número de delitos sino porque también ***optimizaría*** la gestión de los recursos policiales, los medios tanto humanos como materiales que son necesarios para lograrlos y poder detectar donde estos recursos son insuficientes o, por el contrario, se está haciendo un uso excesivo de ellos.

El siguiente trabajo, procura predecir los actos delictivos en los distintos distritos de la ciudad de Chicago con el fin conjugar de forma correcta recursos, ser más eficientes y *aumentar* la presencia de ***dispositivos policiales*** en las zonas requieran un mayor control, así como reducirlos donde no se necesiten tanto. Con esto, se lograr una ***sociedad pacífica*** y segura, y un uso eficaz de los presupuestos destinados a esta aplicación.

Y es que en los últimos años la ciudad de Chicago, requiere de una mejora en materia de seguridad ya que como se verá se ha producido un incremento de estos desde los últimos años rompiendo con esa tendencia decreciente desde finales de los años 90. En los últimos tiempos, se registra ***una media de dos asesinatos al día***, cifra que impacta a la hora de comparar los datos. Por ejemplo, la ciudad de Madrid la cual tiene una población similar y presenta un 0.6 de homicidios por cada 100.000 personas frente a las 28[1] *d*e Chicago.

Si esta comparación por diferencias estructurales se considera poco válida simplemente bastaría contrastarla con ciudades de gran protagonismo dentro de los Estados Unidos como son ***Los Ángeles*** o ***Nueva York***, aunque nos encontramos con más de lo mismo, Chicago sigue desgraciadamente a la cabeza. Las claves a estudiar de estos datos, se centran en el plano legislativo del estado, la tenencia de armas y por supuesto la gestión de recursos policiales.

En primer lugar, la legislación ha cambiado pasando a ser más permisiva en cuanto al control de armas, por otro lado, Chicago colinda con los estados de Indiana y Wisconsin quienes tienen las leyes de armas menos estrictas pudiendo ser adquiridas sin ningún tipo de permiso. Un estudio realizado por el departamento de policía de Chicago confirmó que, del total de las armas utilizadas para cometer crímenes, un 60% procedían de los estados

[1] https://magnet.xataka.com/en-diez-minutos/chicago-tiene-casi-la-misma-poblacion-que-madrid-y-40-veces-mas-asesinatos

antes mencionados lo que tiene una relación más que directa con el crimen ya que un 90% de los asesinatos son producidos por ***armas de fuego.***

Y, por último, el que será uno de los objetivos del estudio, como organizar de forma óptima los dispositivos policiales. No hay más que ver el caso de Nueva York quien incidió en este tema y actualmente ha reducido los homicidios en la ciudad de forma considerable.

Al igual que otras aplicaciones recientemente desarrolladas como pueden ser los casos de ***Beware*** (el cual es un software que utiliza los datos publicados en las redes sociales para cruzarlos con otros muchos pertenecientes. Por ejemplo, al registro de propiedades y cámaras de tráfico prediciendo así delitos e incluso localizando individuos que se encuentren en busca y captura) o de ***Predpol*** (quien lidera la vigilancia predictiva identificando los lugares y momentos donde es más probable que ocurran los delitos).Se cree que sería un gran avance para la ciudad de Chicago como para las diferentes ciudades del mundo en el futuro incorporar un modelo que anunciase donde se tienen que estar y en qué momento los activos policiales deben actuar, adelantándose a los sucesos, conociendo que variables afectan y de qué modo, detectando las zonas calientes y logrando destinar de forma coherente recursos a estas tal y como se hizo en la ciudad de Nueva York.

Bien es cierto que en Estados Unidos ya se están aplicando modelos tales como el presente estudio, un claro ejemplo de estos es el implementado en ***Santa Cruz (California)*** donde mediante analítica del dato se han conseguido reducir en un ***27% los robos***[2] en locales públicos y en un ***11% los asaltos***.

Otro caso sería, el antes comentado en la ciudad de Nueva York, cuyo éxito es más que visible o también los primeros pasos en esta materia que tuvieron lugar en la ciudad de Chicago, y es que, desde hace ya unos años nuestra ciudad ya comenzó su andadura en el análisis del crimen mediante el dato. Chicago cuenta con una lista de los criminales ordenados según sus datos personales y una puntuación que indica la probabilidad de cada uno de ellos de volver a reincidir. Todo este tipo de indicadores, muestran como en los Estados Unidos una vez más se está liderando la implantación de estas nuevas tecnologías

[2] https://www.tuataratech.com/2016/08/como-combatir-el-crimen-con-big-data.html

y como muchos otros países deberían tomar nota y no quedarse atrás como es el caso de ***España.***

Por todo esto, se pretende aportar algo de luz a la materia y buscar ***respuestas eficaces***, ¿Qué áreas son más conflictivas?, ¿Qué horarios suponen un mayor peligro?, ¿Qué variables afectan y en qué manera a estos comportamientos de la sociedad?, ¿Es realmente viable lograr modelos predictivos los suficientemente desarrollados para tener un impacto destacado en la criminalidad?

Para todas estas cuestiones, se implementará la tecnología del Big Data. En primer lugar, se realizará una colecta de las variables que pueden tener un impacto en nuestra predicción y se llevara a cabo un análisis exploratorio de esas independientemente de si al final son relevantes o no. La relevancia la indicaran principalmente nuestros modelos, y estas variables pueden ser de distintos tipos o ámbitos.

Posteriormente, se creará nuestro modelo supervisado mediante la prueba y error de los distintos algoritmos que nos proporciona la inteligencia artificial eligiendo finalmente aquel con métricas más precisas.

Por último, para su correcta comprensión y visualización se realizará un informe detallado presentando los resultados, así como las conclusiones obtenidas.

2. Objetivos

El presente trabajo de final de máster tiene como ***objetivo general***, predecir la delincuencia de la ciudad de Chicago para los próximos años en función de los distritos. Con la intención de dotar a los cuerpos de seguridad de más recursos, logrando así generar una ciudad más ***segura*** y justa para sus ciudadanos.

Para ello, es necesario plantear una serie de objetivos específicos:

1. Mostrar la importancia de las técnicas y tecnologías ***Big Data*** en el ámbito de la seguridad ciudadana. Para, controlar y disminuir los actos delictivos de las ciudades estadounidenses, más detalladamente la ciudad metropolitana de Chicago.
2. Utilización de ***técnicas ETL*** para la limpieza, trasformación y gestión de los datos obtenidos mediante una API gestionada por los cuerpos de seguridad de Chicago.
3. Realización de un exhaustivo análisis de los datos extraídos, así como la visualización de una serie de variables claves para entender los millones de datos con los que cuenta el dataset. En este punto, se podrá comprobar la gran importancia de hacer un buen reprocesado de características para mejorar la capacidad de predicción de una serie de algoritmos.
4. Análisis comparativo de los diferentes algoritmos de ***aprendizaje automático*** utilizados durante el curso. Para ello, se deben usar algoritmos de ***entrenamiento*** y de ***test***, para reconocer cual es la opción más óptima a la hora de generar la predicción.
5. Implementación de ***tecnologías punteras en visualización*** expuestas durante el módulo de Inteligencia de Negocio del Máster. Con el propósito, de representar de forma analítica los datos recopilados en los puntos anteriores y facilitar la interpretación de los resultados finales.
6. Por último, se mostrarán las conclusiones obtenidas de las predicciones realizadas durante el trabajo y se llevara a cabo una reflexión sobre las grandes repercusiones que puede llevar a cabo la delincuencia en un área metropolitana como Chicago.

3. Estado del arte

Antes de entrar en materia con el diseño y la implementación, se expondrán una serie de aspectos relevantes relacionados con los estudios realizados para abordar nuestro proyecto fin de máster. En el apartado 3.1 se introduce un pequeño ***contexto*** detallado sobre la situación criminal y judicial que ha vivido la ciudad de chicago. Y, en el apartado 3.2 se hablarán de ***técnicas*** parecidas a la nuestra ya implementadas o en proceso de desarrollo.

3.1. Contexto

EEUU es una de las mayores potencias del mundo y uno de los países con más crímenes registrados. Según un estudio en el año 2005, este país ya contaba 2,25 millones de presos, lo que supone un 40% de la población entre rejas. La mayoría de estos presos, había cometido actos violentos.

A lo largo de la última década, la ciudad de metropolitana de chicago ha tomado mucha relevancia debido a un aumento de crímenes registrados entre la población. Esto se puede deber a una serie de factores como el calor, las aglomeraciones de gente, las tendencias económicas, el permiso de armas de fuego o la diversidad cultural. Por ejemplo, en el año 2015 en un solo fin de semana se registraron ***80 disparos***, de los cuales 15 personas fallecieron.

Además, las tasas de encarcelamiento han ido aumentando a lo largo de los años, mientras que las tasas de la delincuencia se han ido reduciendo. Esto, se debe a que la justicia se ha vuelto mucho más dura, es decir, la nación ha tomado la ***decisión colectiva*** de castigar ***más severamente*** a los delincuentes.

Por ello, este trabajo quiere desarrollar una predicción para seguir manteniendo esa reducción del crimen, anticipándose a los hechos que sucederán en un futuro próximo.

3.2. Revisión de estudios anteriores

Con crecimiento del Big Data y la inteligencia artificial, las fuerzas de seguridad han ido implementando técnicas capaces de predecir las circunstancias de cometer un determinado crimen.

Actualmente, ya existen países que usan la famosa técnica de "análisis predictivo" que consiste en implementar una serie de algoritmos de inteligencia artificial para identificar focos que requieren la intervención de la policía, con la intención de prevenir delitos o incluso resolver crímenes pasados.

El país precursor de este tipo de técnicas es EEUU, como era de esperar, debido a la gran cantidad de dólares que destina a la tecnología. Sin embargo, el primer país en la Unión Europea, fue nuestro país vecino Francia. Actualmente, un ***60%*** de los países utilizan ***algoritmos de predicción*** a día de hoy***.***

Algunos de las técnicas más utilizadas en nuestro país:[3]

- ***EurocopPred-crime***: Desarrollado en la empresa EuroCop Security Systems en colaboración con la Universidad Jaume I de Castellón en el año 2011. Se trata de un modelo que utiliza diferentes fuentes de origen, como datos socioeconómicos, urbanísticos y geográficos. Con toda esta información, genera mapas de calor con el fin de identificar las zonas más problemáticas.
- ***PredictivePolicePatroling(P3-DSS):*** desarrollado por la policía nacional de Madrid. Se trata un algoritmo que coge los datos estadísticos de la policía sobre crímenes para aprender y generar en un mapa que indica las ruta que deben hacer las patrullas de policía. Actualmente, esta iniciativa está en estudio.
- ***Sistema de análisis y explotación estadístico (SAEX):*** la guardia civil ha desarrollado internamente un sistema que permite recopilar un repositorio de los crímenes cometidos. Más que un programa de predicción es de recopilación.

Algunas técnicas usadas en otros países:

- ***Nacional AnalyticsSolution(NDAS)[4]:*** es un sistema implementado por la policía de Reino Unido que pretende combinar la inteligencia artificial y la estadística para evaluar el riesgo de que una persona cometa un crimen.
- ***PredPol***:[5] una startup californiana que usa algoritmos de Big Data para predecir cómo se va comportar la delincuencia. Es decir, predecir cuando y donde se va a cometer. Con este software predictivo, se consiguió en solo 14 meses una reducción del crimen y del cual se ha mencionado anteriormente.

[3] https://blog.realinstitutoelcano.org/prevencion-del-crimen-y-prediccion-de-delitos-en-que-punto-esta-espana/
[4] https://www.lavanguardia.com/tecnologia/20181202/453268636098/policia-britanica-uso-inteligencia-artificial-delitos-crimenes-delincuencia.html
[5] https://elpais.com/tecnologia/2017/03/09/actualidad/1489078250_691655.html

La mayoría de las técnicas explicadas tienen un punto en común. Dicho punto, es que la mayoría de ellas están en desarrollo o en estado de prueba. La única más desarrollada podría ser ***PreedPool*** que se usa en algunos puntos de Estado Unidos, pero no todos.

La intención de este proyecto es implementar un algoritmo capaz de usarse en la ciudad de Chicago y posteriormente expandirse por otros puntos de EEUU. Generando en un futuro que los países o naciones implementen este algoritmo para mejorar la seguridad de sus ciudades.

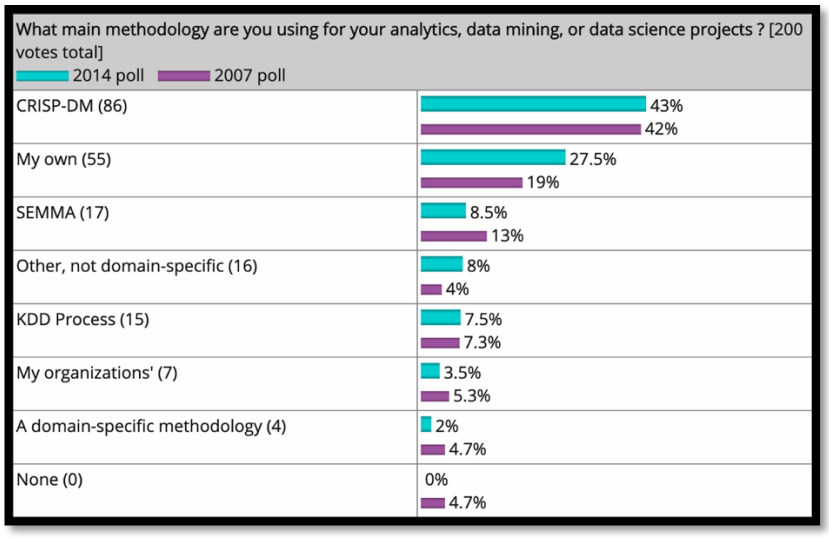
4. Metodología

La metodología utilizada a lo largo del proyecto es aquella con mayor reconocimiento y recorrido en los proyectos de ***data mining***, es decir, la metodología **CRISP-DM**.[6]

Este método no es más que la evolución y mejora en el tiempo del conocido termino ***KDD*** utilizado desde finales de los años 90 y al que se hace referencia cuando se habla de la búsqueda de conocimiento en los datos. Es a finales de esos años, concretamente en 1999 cuando un consorcio formado por empresas europeas protagonistas en estos proyectos se pusieron de acuerdo en la necesidad de crear una guía modelo de libre distribución, naciendo así CRISP-DM.

Dentro de la industria, han aparecido otras metodologías similares como ***SEMMA*** pero sin embargo una encuesta realizada por ***KDNuggets***[7], considerado uno de los principales ***portales*** de información en el ámbito de la inteligencia artificial y Big Data. Muestran en el año 2007 que los resultados de la encuesta, se posicionan claramente a favor de CRISP-DM, siendo esta la principal y la "4 veces" más implementada que SEMMA.

Ilustración 1. Metodologías utilizadas en proyectos de minería de datos



Fuente: kdnuggets.com

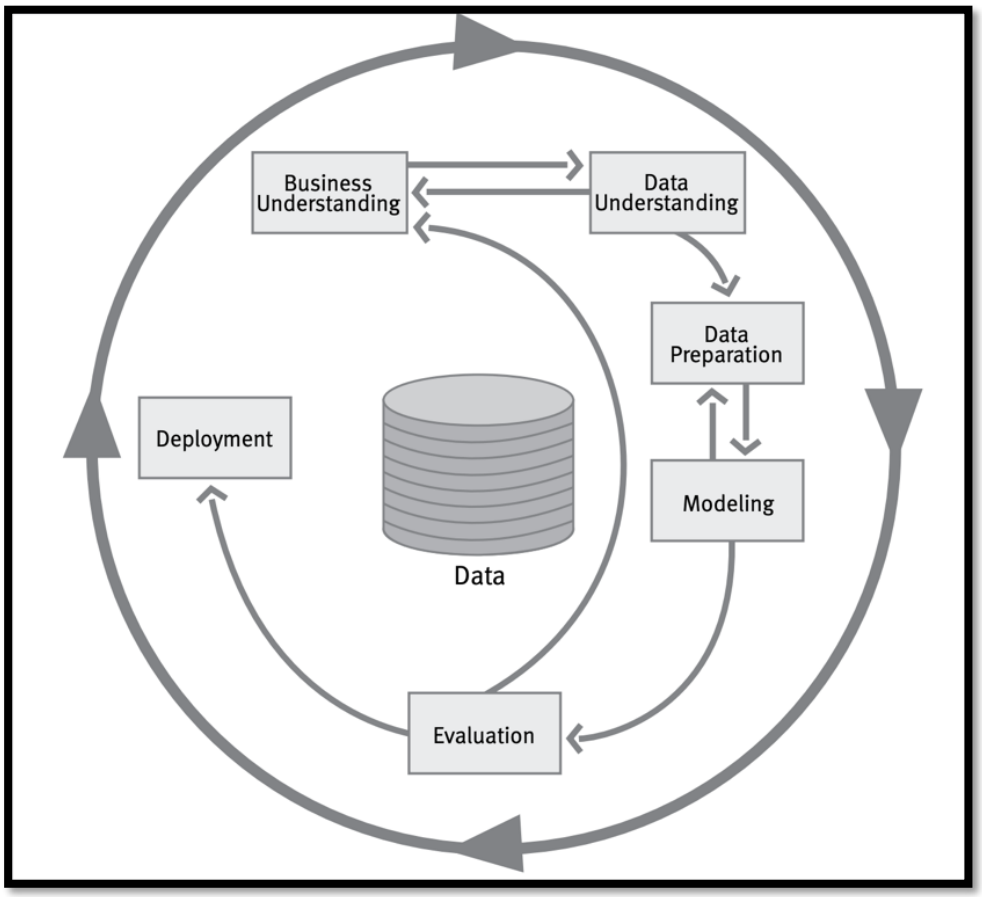
[6] https://www.the-modeling-agency.com/crisp-dm.pdf,
[7] https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

CRISP-DM aporta un cuadro normalizado del ciclo de vida de un proyecto basado en el análisis de datos. Dentro de cada una de sus fases, se encuentran las tareas que se han realizado, su contexto y lo más relevante las diferentes relaciones entre estas.

Uno de los puntos a desatacar es que esta técnica no da por concluido ningún proyecto, ya que supone que una vez alcanzado el modelo ideal este, se encuentra relacionado con otros proyectos. Por todo esto, es necesario documentarlo de forma exhaustiva y mantenerlo para que otros proyectos se puedan nutrir de él.

CRISP-DM está compuesto por un total de seis fases. Las cuales no siguen un orden completamente rígido, como se observa en la ilustración 2.Se puede, destacar enormemente su naturaleza cíclica que no da por finalizado en ningún momento un proyecto.

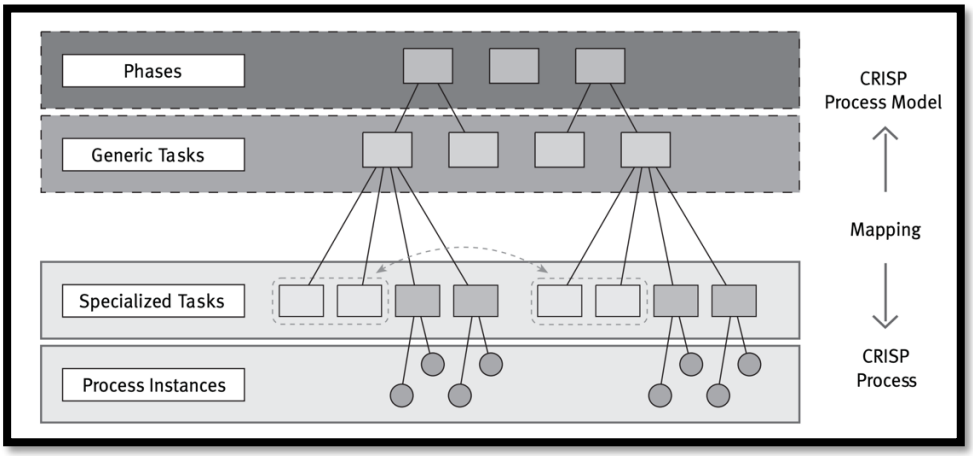
Ilustración 2. Flujo metodología CRISP-DM



Fuente:Chapman, P.; Clinton, J., Keber y otros (2000). "CRISP-DM 1.0 Step by step guide"

Dentro de cada una de estas fases, se encuentran 4 niveles de abstracción que siguen una organización jerárquica como son las **fases**, ***tareas generales***, ***tareas especializadas*** e ***instancias de un proceso***.

Ilustración 3. Jerarquía de tareas dentro de cada fase CRISP-DM



Fuente: Chapman, P.; Clinton, J., Keber y otros (2000). "CRISP-DM 1.0 Step by step guide"

Por todo ello, se presentan las diferentes fases de nuestro proyecto con las correspondientes aportaciones realizadas:

I. Business Understanding

La fase inicial es la más importante del proyecto porque sin unos objetivos claros y bien definidos no es posible elegir ni entrenar un algoritmo de forma correcta. Gracias a este punto en la investigación, se conseguirá comprender cual es el problema a resolver y esto ayudará en el futuro a recolectar los datos necesarios e interpretar eficazmente los resultados.

Nuestra investigación, se centra en poder realizar una predicción de los delitos cometidos en la ciudad de Chicago. Con esto, se pretende mejorar la calidad de vida en la ciudad, consiguiendo así una reducción del crimen. Se cuenta con una gran cantidad y variedad de datos recogidos en un *dataset* sólido. Mediante la minería de datos, se buscarán patrones de comportamiento creando así un modelo de predicción del crimen, capaz de proporcionar la información necesaria para adelantarnos y atajar el problema.

II. Data Understanding

La comprensión de los datos pasa por su recolección, su comprensión y acaba en un análisis superficial de estos descubriendo posibles patrones entre ellos. Como, por ejemplo, correlaciones, calidad de los datos, etc. Se dice que cuando se sitúan los datos recopilados frente a frente y se conocen con plenitud, se puede tener una visión global sobre las cartas con las que jugamos.

En nuestro caso tal y como se ha comentado anteriormente, los datos se han recogido gracias a una **API** proporcionada por el ayuntamiento de la ciudad de Chicago donde se han obtenido muchas variables de gran relevancia. Entre estos datos, se encuentra por ejemplo con las coordenadas de los delitos cometidos que ayudan a generar zonas calientes con predisposición a conflictos y los distritos policiales que atienden los actos violentos que ayudaran a rastrear como están repartidos esos activos policiales.

Tras este primer análisis, se ha comprobado de que tipo eran nuestros datos y en qué proporción estaban para poder saber cuáles eran más persistentes y cuales menos. Posteriormente gracias a un ***"describes"*** sacamos la información general de nuestros datos como puede ser la media, los máximos y mínimos, los distintos cuartiles etc.

Hasta aquí llega siempre el primer análisis simplista de cualquier trabajo de minería de datos.

III. Data Preparation

Como se viene comentando en esta fase, se pondrá a punto nuestros datos para comenzar con su implementación en los modelos. Es ahora, cuando se tienen que pulirlo para que estén perfectos y permitan trabajar a los algoritmos de forma eficiente. Si esta etapa del trabajo, no se realiza correctamente producirá problemas en el futuro con toda seguridad por lo que se considera una parte fundamental de cualquier trabajo de minería de datos.

Dentro de nuestra investigación, se ha comenzado por un clásico, es decir, la eliminación de los valores nulos. Al comprobar que estos se correspondían con un 8% del total de los datos recogidos.

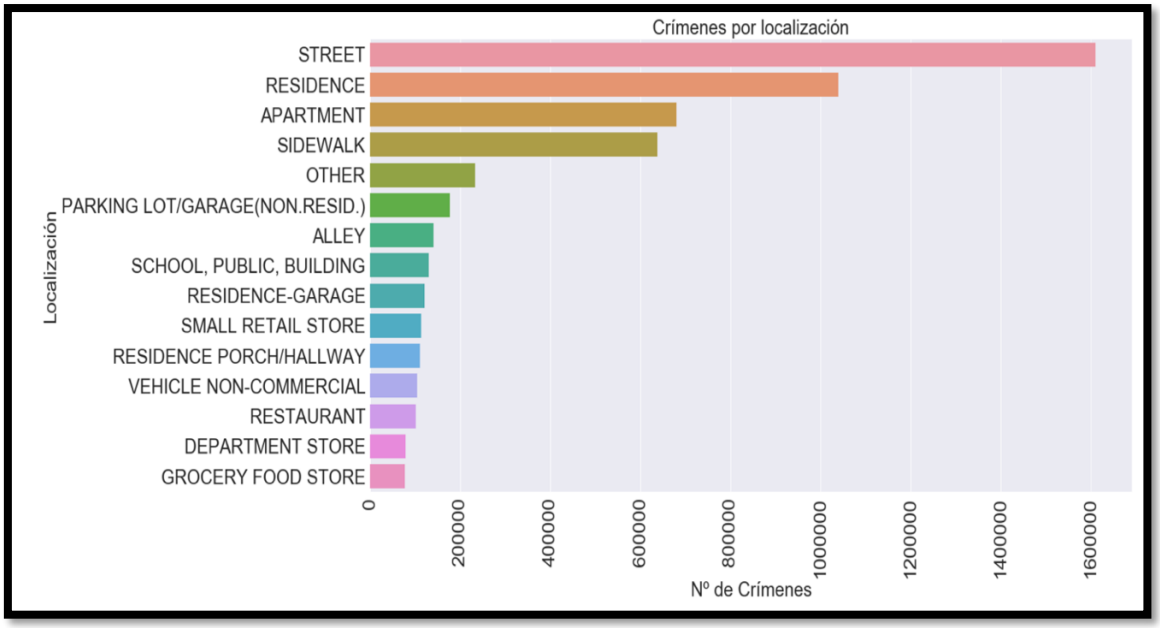
En segundo lugar, al tratarse de un análisis en el tiempo del crimen para su posterior predicción se ha utilizado la fecha de los acontecimientos como *id* de nuestros datos generando una separación entre el año mes y día. Al mismo tiempo, se ha prescindido de aquellas variables que no suponían un valor añadido para la investigación como **FBI code**. Además, se ha tenido que agrupar la variable **PrimaryType** para usar valores más genéricos. Por ejemplo, todos los que tienen que ver con delitos sexuales han sido agrupados bajo **SEX ASSAULT**.

Dentro de esta etapa de la metodología, se ha incluido lo que se denomina ***análisis exploratorio***. Es decir, todas aquellas variables fundamentales visualizadas en gráficos mediante los que se puede tener una primera impresión. El motivo por el que se ha decidido anexarlo en esta fase, es porque la siguiente ya se centra en la predicción del crimen. Para nosotros, este trabajo es previo al modelo y está más próximo a la preparación de los datos que al propio modelo en sí.

Aquí aparece la eliminación de columnas no significativas, la reestructuración del dataset, la creación de un nuevo datase centrado en los delitos relacionados con las drogas, la conversión de variables categóricas a variables numéricas para su posterior utilización en el modelo.

Se han representado gráficamente algunas variables relevantes como por ejemplo el porcentaje de crímenes según su tipología, la división de los crímenes relacionados con drogas en función del tipo de sustancia, el número de crímenes y su evolución a lo largo de nuestra serie temporal, así como su tendencia, el lugar donde tienen una mayor presencia, o el número de crímenes por localización.

Ilustración 3. Crímenes por localización



Fuente: Elaboración propia

IV. Modeling

Esta fase, se considera el centro de nuestro proyecto, la generación de un modelo que sea capaz de satisfacer la necesidad inicial del proyecto. Aquí, se seleccionan los algoritmos a implementar, así como los hiperparámetros.

También se determina, cuáles serán los métodos de evaluación de nuestros modelos y aquellas métricas que permiten concluir si el algoritmo aplicado es bueno o no. En conjunto, se abarca desde la selección del modelo hasta su posterior evaluación técnica.

En nuestro caso, se han escogido unos algoritmos de clasificación, concretamente un "***Gradient boosting***" y un ***"Random forest***" y para su posterior evaluación se ha optado la ***matriz de correlación***. Esta matriz es una de las métricas más utilizadas a la hora de medir la eficiencia de los algoritmos de clasificación. Gracia, a esta matriz se podrá determinar que también funciona nuestro modelo en función de la relación que muestren las variables.

Tanto para el "**Random Forest**" como para el "**Gradient Boosting**" se han seguido los mismos pasos. En primer lugar, se ha realizado una división del dataset, y se han ajustado los modelos a unos estimadores bases. Posteriormente, se ha decido llevar a cabo la

técnica de ***"Stepwise"*** con el fin de optimizar más el modelo y así alcanzar el más eficiente con el número óptimo de variables a considerar.

En cuanto al problema de sobreajuste del modelo y la correcta selección de los hiperparámetros se ha implementado el algoritmo de "***GridSearch***". Esta técnica de la búsqueda de cuadrículas permite mejorar el rendimiento del algoritmo probando las diferentes combinaciones de hiperparámetros y ofreciendo la mejor solución. Una vez realizados todos estos pasos pasamos a su posterior evaluación.

V. Evaluation

Es en esta fase donde se evalúa los resultados obtenidos por nuestro modelo mediante una técnica de validación, llamada matriz de correlación que muestra los resultados de una forma visual sencilla. Para, poner en relieve la fiabilidad obtenida por nuestro modelo. También en este paso, es donde se revisan posibles errores, se repiten procesos anteriores y se realizan distintas pruebas.

Es en este punto del proyecto donde se tiene una investigación tan avanzada se puede dar marcha atrás o continuar. Si los resultados obtenidos no fuesen los esperados se plantearía volver a la fase de preparación de los datos para buscar nuevas alternativas e incluso volver a empezar el proyecto desde cero porque se haya realizado un mal planteamiento desde el inicio.

VI. Deployment

Una vez llegados a este punto, se elige cómo será la implementación del modelo, es decir, como materializar el trabajo hecho para conseguir nuestro objetivo. En nuestro caso, lo teníamos muy claro, mediante un informe final se pretende que los agentes participen en el proyecto, para conocer de forma actualizada en que sitios se van a cometer los crímenes y se puedan así gestionar recursos de forma anticipada. La idea es sencilla, los futuros datos se irán cargando al modelo que irá prediciendo y mostrando los resultados finales. El usuario simplemente tendrá que darle a actualizar y el informe se refrescará con la nueva información.

En nuestro caso a diferencia de muchos otros no se contará con un informe final ya que la idea es mantenerlo en el tiempo y en todo caso, se vaya ampliando el modelo con

más atributos que sean relevantes. También, la idea es que cada cierto tiempo se evalúe la eficacia del modelo con los datos que han ido entrando en los últimos meses y así ver donde se necesita un mayor desarrollo.

II. PARTE PRINCIPAL

En el siguiente punto, se desmenuzarán todos y cada uno de los pasos seguidos e implementados en nuestro análisis con el fin de clarificar y facilitar la compresión.

5. Obtención de datos

Los datos con los que cuenta el presente trabajo, se obtienen a través de una API llamada ***CHICAGO DATA PORTAL*** que contiene una multitud de datos abiertos relacionados con la ciudad de Chicago. Se entiendo como API, aquella plataforma software que permite la comunicación de información de un punto a otro. Esta API, es un sistema de reporting capaz recoger los datos publicados por la policía de chicago en el sistema ***CLEAR*** y generarlos en distintos formatos. En este caso, se ha elegido .csv porque es más rápido y fácil a la hora de llevar a cabo las cargas de datos.

Además, es necesario citar que la API proporciona los datos de los crímenes de la ciudad de chicago desde el 2001 hasta el momento. Pero, en el presente trabajo solo se usan los datos hasta el 2019 para que la evaluación sea lo más veraz posible. Lo que lleva a tener un total de 6.954.967 observaciones, distribuidas entre 30 columnas.

A continuación, se mostrarán las columnas con las que cuenta el dataset y se introducirá una breve explicación para entender los datos mucho mejor.

- **ID** - Identificador único para el registro.
- **Case Number**- Numero del departamento de policía de chicago. Es diferente para delito.
- **Date** - Fecha cuando ocurre el incidente.
- **Block** - La dirección donde ocurre el incidente.
- **IUCR** - Código del crimen. Relacionado con PrimaryType.
- **PrimaryType** - Descripción del código del crimen.

- **Description**– Descripción más detalla sobre PrimaryType.
- **LocationDescription** - Localización donde ocurre el incidente.
- **Arrest** - Indica si se llevó a cabo un arresto por la policía.
- **Domestic** - Indica si el incidente estuvo relacionado con el hogar.
- **Beat** – Indica el sector donde se llevó a cabo el incidente. Hay 22.
- **District** - Distrito donde ocurre el incidente.
- **Ward** - El barrio donde ocurre el incidente.
- **Community Area** - Indica el área donde ocurre el incidente. Chicago tiene aproximadamente 77 áreas.
- **FBI Code** - Indica la clasificación del delito como se describe en (NIBRS) del FBI.
- **X Coordinate** - Coordenadas X, donde ocurre el incidente.
- **Y Coordinate** - Coordenadas Y, donde ocurre el incidente.
- **Year** - El año donde ocurre el incidente.
- **UpdatedOn** - La últimaactualización de la información.
- **Latitude** - La latitud donde ocurre el incidente.
- **Longitude** - La longitud donde ocurre el incidente.
- **Location** - Localización donde ocurre los hechos para representación de mapas.
- **Historical-Wards 2003-2015** - Los barrios históricos entre el 2003 y 2015.

- **Zip Codes** - Códigos Postales.
- **CommunityAreas** - Áreas por comunidad dentro de los distritos de Chicago.
- **CesnsusTracts** - Registros del Censo de la población.
- **Wards** - Barrios donde se encuentra el crimen (repetido en el dataset).
- **Boundaries ZIP Codes** - Códigos Postales fronterizos de las áreas colindantes.
- **PoliceDistricts** - Distritos Policiales.
- **PoliceBeats** - Sectores Policiales.

6. Arquitectura

El siguiente paso a implementar, es la explicación detallada de las arquitecturas usadas durante todo el proyecto para lograr nuestro objetivo final. Se explican detalladamente en dos grupos principalmente:

6.1 Tecnologías implementadas

Se citarán y explicaran, las técnicas en auge que se han ido usando e implementando durante la elaboración del trabajo. A continuación, se muestran una breve explicación:

- ANACONDA[8]: Anaconda es una suite de código abierto que abarca una serie de librerías y conceptos diseñados para el desarrollo de la ciencia de datos con Python y otros tipos de lenguaje de programación. En líneas generales, es una distribución de varios lenguajes que funciona como un gestor de entorno y de paquetes principalmente.

- COLAB: Mas conocido como Google Colab es un entorno gratuito de Google que no requiere ningún tipo de configuración y se ejecuta todo en la

[8] https://blog.desdelinux.net/ciencia-de-datos-con-python/

nube. Varias personas del grupo han usado esta plataforma para implementar código debido a problemas con Jupyter.

- JUPYTER[9]: Jupyter Notebook es una aplicación de servidor cliente que permite la elaboración de documentos que mezclan código con varios lenguajes de programación y texto.
- POWERBI[10]: Power BI es una solución destinada principalmente a la inteligencia empresarial que permite unir varias fuentes de datos, modelizar y finalmente analizar datos mediante paneles o informes de manera muy fácil y atractiva.

6.2. Diagrama de la arquitectura del flujo

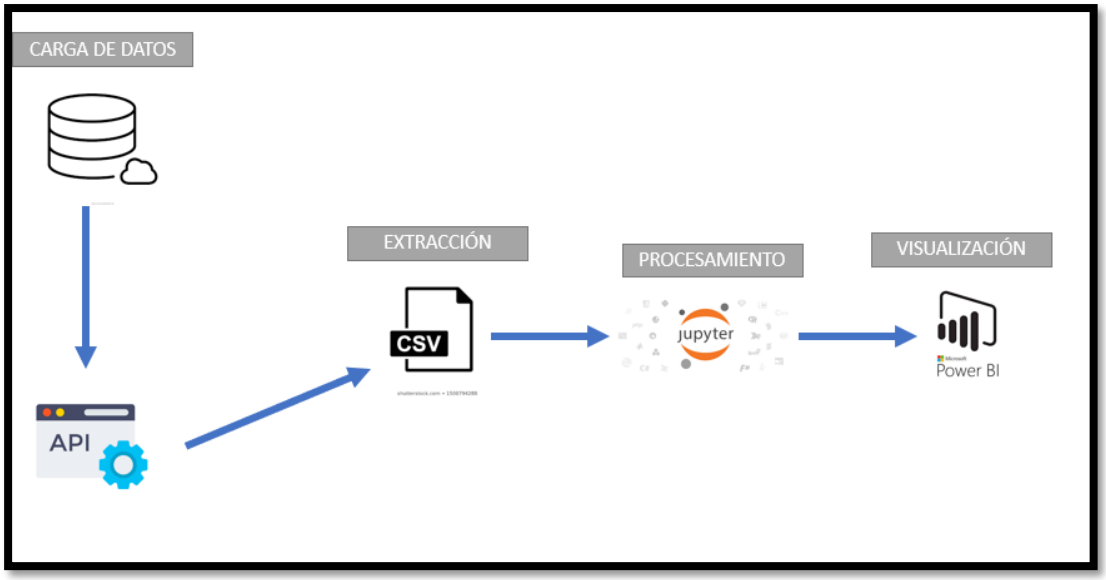
Además, se ha querido mostrar las tecnologías asociadas a los flujos de nuestra arquitectura. Es decir, en cada fase de la arquitectura del proyecto, se implementa una tecnología, como se puede observar en la imagen inferior.

Las fases que se han considerado, son las siguientes:

1. Carga de datos
2. Extracción de datos
3. Procesamiento de datos
4. Visualización de cuadros de mando e informes dinámicos.

[9] https://www.ionos.com/digitalguide/websites/web-development/jupyter-notebook/
[10] https://www.makesoft.es/powr-bi-que-es-power-bi/

Ilustración 4.Diagrama de la arquitectura implementada



Fuente: Elaboración propia

Si se observa en la imagen superior, los datos se extraen en formato .csv de una API que tira "***históricos***" de la policía de chicago, como se ha mencionado en los puntos anteriores. Una vez cargados dichos datos, se procesarán una serie de algoritmos en la herramienta de ***jupyter*** que permite elaborar código de forma documentada en el lenguaje de programación ***Python***.

Para finalizar, se usará una herramienta muy conocida llamada ***Power Bi*** para visualizar y representar los datos obtenidos a través de los cuadros de mando.

7. Desarrollo

Una vez explicado de donde se obtienen los datos y las herramientas utilizadas, se describirán las fases principales del desarrollo que han ido aconteciendo en nuestro proyecto. Y, cabe indicar que este apartado se explica también brevemente en el notebook adjuntado con una estructura similar.

7.1. Carga de información

En las primeras instancias de código, han de implementarse siempre las librerías de Python. Estas, son necesarias para ejecutar el código de todo nuestro proyecto y se van nutriendo durante todo el desarrollo. En función de lo que se quiera representar, se implementara un tipo de librería u otras.

Aunque, siempre existen unas librerías básicas que se implementan, como pueden ser:

- Pandas: La librería panda es una de las más relevantes porque hace posible un análisis de los datos, para su posterior limpieza.
- Numpy: La librería de numpy se usada para poder llevar a cabo cualquier cálculo numérico.
- Matplotlib.pyplot: La librería Matplotlib hace posible generar gráficos sobre una serie de datos. En los puntos siguientes, se muestran gráficos elaborados con esta librería.
- Seaborn: La librería seaborn es bastante avanzada y sirve para generar gráficos más elegantes, partiendo de la base de Matplotlib.

Una vez implementadas todas las librerías, se cargara el archivo adquirido en pasos anteriores mediante el comando "***pd.read***". A continuación, se muestra una imagen del código implementado y la tabla que se imprime por pantalla.

Ilustración 5. Comando para cargar dataset

```
dataset = pd.read_csv ('crimen.csv', low_memory = False, encoding="ISO-8859-1")
#Hemos hecho un low_memory=False, porque nos daba problemas a la hora de interpretar el tipo de datos de algunas filas
```

Fuente: Elaboración Propia

Importante fijarse en las palabras "low_memory" y "encoding" que promueven que los datos se ejecuten e impriman sin problema.

Ilustración 6. Tablas cargadas

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District	Ward
0	11807717	JC408714	08/26/2019 11:58:00 PM	060XX S JUSTINE ST	0520	ASSAULT	AGGRAVATED:KNIFE/CUTTING INSTR	RESIDENCE	True	True	713	7.0	16.0
1	11807826	JC408716	08/26/2019 11:57:00 PM	012XX N LA SALLE DR	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	True	True	1821	18.0	2.0
2	11807746	JC408370	08/26/2019 11:56:00 PM	065XX S PROMONTORY DR	3731	INTERFERENCE WITH PUBLIC OFFICER	OBSTRUCTING IDENTIFICATION	PARK PROPERTY	True	False	331	3.0	5.0
3	11807718	JC408708	08/26/2019 11:55:00 PM	079XX S ELLIS AVE	1310	CRIMINAL DAMAGE	TO PROPERTY	RESIDENCE	False	False	624	6.0	8.0
4	11807777	JC408706	08/26/2019 11:45:00 PM	062XX W 64TH PL	0420	BATTERY	AGGRAVATED:KNIFE/CUTTING INSTR	APARTMENT	False	False	812	8.0	23.0

Community Area	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	Location	Wards 2003-2015	Zip Codes	Community Areas	Census Tracts	Wards	Boundaries - ZIP Codes
67.0	04A	1167036.0	1864704.0	2019	09/02/2019 04:03:45 PM	41.784306	-87.663123	(41.784305722, -87.663123342)	44.0	22257.0	65.0	277.0	2.0	23.0
8.0	08B	1174908.0	1908624.0	2019	09/02/2019 04:03:45 PM	41.904654	-87.632948	(41.904653619, -87.6329484)	51.0	14926.0	37.0	17.0	11.0	54.0
42.0	24	1192337.0	1862034.0	2019	09/02/2019 04:03:45 PM	41.776400	-87.570448	(41.776400433, -87.570448042)	32.0	22538.0	9.0	134.0	33.0	24.0
44.0	14	1184271.0	1852522.0	2019	09/02/2019 04:03:45 PM	41.750491	-87.600314	(41.750491135, -87.600314257)	9.0	21546.0	40.0	247.0	35.0	61.0
64.0	04B	1136032.0	1861102.0	2019	09/02/2019 04:03:45 PM	41.775028	-87.776884	(41.775028096, -87.776883531)	23.0	22268.0	62.0	266.0	6.0	7.0

Fuente: Elaboración propia

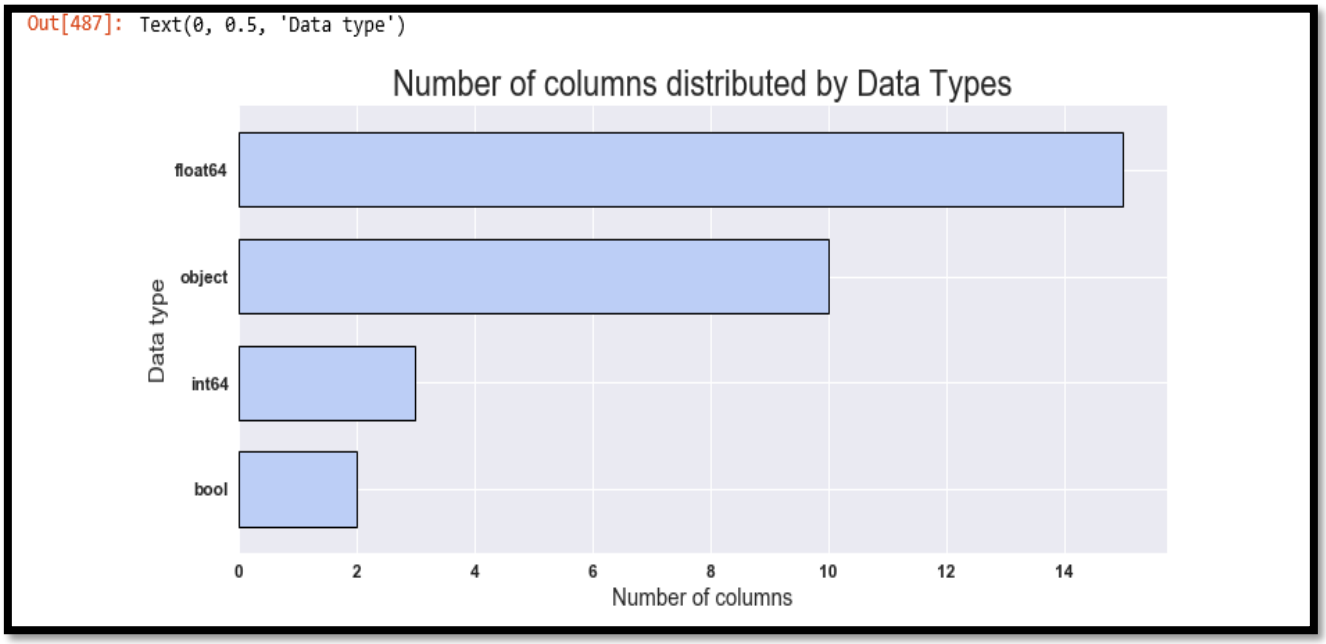
7.2. Análisis exploratorio

Como paso previo a la limpieza, se hará un análisis exploratorio de los datos con los que se cuenta para entender y optimizar la información relevante.

Este apartado o punto, se considera imprescindible a la hora de llevar a cabo cualquier modelo de Machine Learning. Porque, en función de las características y las variables de los datos, se planteará un algoritmo u otro. No implementar este paso en el desarrollo, puede suponer la elaboración de un modelo sobreajustado o erróneo por el tipo de datos o variables.

Por ello, un ejemplo de análisis descriptivo implementado en la memoria seria, los ***tipos de datos*** que contienen nuestro conjunto.

Ilustración 7.Tipos de datos



Fuente: Elaboración propia

En la ilustración 7, se muestra simple vista que variables son numéricas y cuales son categóricas. La mayoría de los datos son tipo ***float***y ***object,*** lo que indica que el formato tipo ***object*** seguramente será trasformado, eliminado o agrupado a la hora de evaluar el modelado.

Otro tipo de análisis, implementado en el proyecto es el ***análisis estadístico*** para ver la funcionalidad de las variables o la ***matriz de correlación*** para observar que variables están más relacionadas entre ellas.

7.3. Limpieza de variables

En este punto, se limpiarán dichos datos en función de los errores o patrones detectados en el paso anterior. De ahí, la importancia de las acciones anteriores.

Como era obvio, existen una serie de datos y variables que hay limpiar o trasformar para la futura predicción. Por ejemplo, los cambios más frecuentes son; limpieza de valores nulos o el formateo de fechas en función de los datos.

A simple vista, se observan que existen variables que no aportan valor o incluso alguna que contienen valores nulos. En la ilustración 8, se representan las variables que contienen datos nulos.

Ilustración 8.Número total de Naan

```
ID : 0
Case Number : 4
Date : 0
Block : 0
IUCR : 0
Primary Type : 0
Description : 0
Location Description : 5451
Arrest : 0
Domestic : 0
Beat : 0
District : 47
Ward : 614826
Community Area : 613495
FBI Code : 0
X Coordinate : 65971
Y Coordinate : 65971
Year : 0
Updated On : 0
Latitude : 65971
Longitude : 65971
Location : 65971
Historical Wards 2003-2015 : 85976
Zip Codes : 65971
Community Areas : 83228
Census Tracts : 81088
Wards : 83116
Boundaries - ZIP Codes : 83181
Police Districts : 82149
Police Beats : 82126
```

Fuente: Elaboración propia

El total de valores vacíos no alcanza el 8% de la muestra. Por lo que, se ha tomado la decisión de eliminar esos valores. Esta acción, no falsearía los datos en ningún momento porque son una cantidad bastante reducida. Esto, se logra mediante la línea de comando "***.dropna***" que se muestra en la siguiente ilustración.

Ilustración 9. Eliminación de valores nulos

```
#Se eliminan los valores nulos
dataset.dropna(axis=0,inplace=True)
```

Fuente: Elaboración propia

Otra acción relevante, será trasformar las fechas en índices porque el proyecto de predicción es a lo largo del tiempo. En la ilustración 10, se muestra como quedaría los datos una vez se ha establecido las fechas como índice. Además, esta trasformación de

fechas permite desglosarlas fácilmente, permitiendo sacar los días, meses y años de forma individual.

Ilustración 10. Tabla con índices

Date	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District
2019-08-26 23:58:00	11807717	JC408714	2019-08-26 23:58:00	060XX S JUSTINE ST	0520	ASSAULT	AGGRAVATED:KNIFE/CUTTING INSTR	RESIDENCE	True	True	713	7.0
2019-08-26 23:57:00	11807826	JC408716	2019-08-26 23:57:00	012XX N LA SALLE DR	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	True	True	1821	18.0
2019-08-26 23:56:00	11807746	JC408370	2019-08-26 23:56:00	065XX S PROMONTORY DR	3731	INTERFERENCE WITH PUBLIC OFFICER	OBSTRUCTING IDENTIFICATION	PARK PROPERTY	True	False	331	3.0
2019-08-26 23:55:00	11807718	JC408708	2019-08-26 23:55:00	079XX S ELLIS AVE	1310	CRIMINAL DAMAGE	TO PROPERTY	RESIDENCE	False	False	624	6.0
2019-08-26 23:45:00	11807777	JC408706	2019-08-26 23:45:00	062XX W 64TH PL	0420	BATTERY	AGGRAVATED:KNIFE/CUTTING INSTR	APARTMENT	False	False	812	8.0

Fuente: Elaboración propia

En último lugar, se agruparán valores de una serie de variables para facilitar el futuro modelado. A continuación, se muestran y explican las variables que han experimentado una variación:

a. ***PrimaryType***: Esta variable contiene millones de tipos de crímenes, por lo que, se han agrupado principalmente en 6 categorías. En la ilustración 11, se muestra la distribución final de los datos.

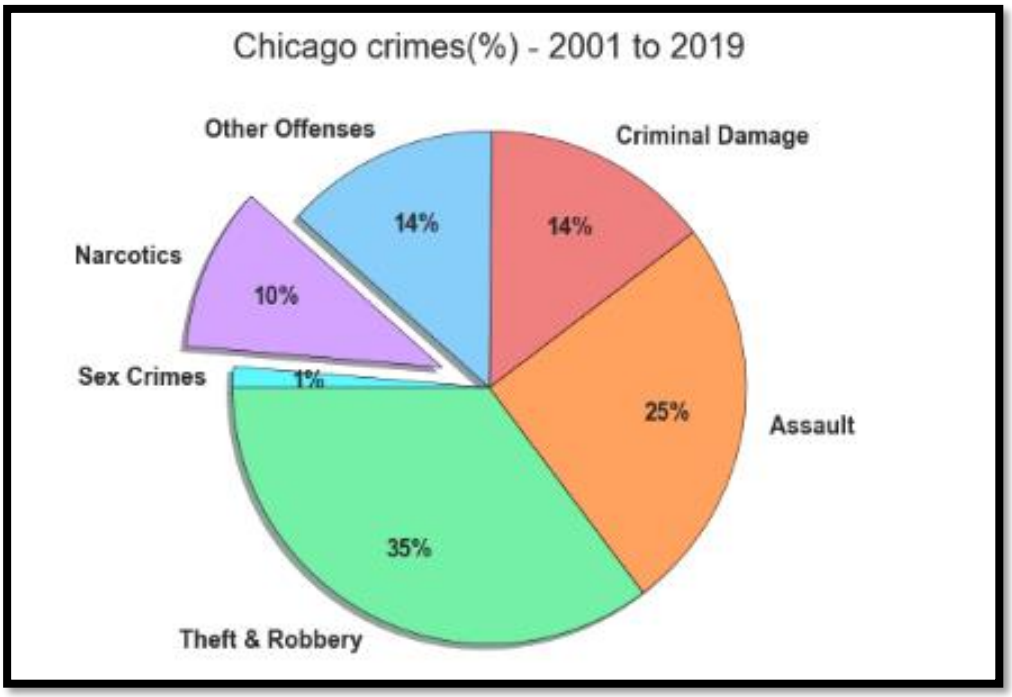
Ilustración 11. Agrupación de crímenes

```
THEFT & ROBBERY     2204043
ASSAULT             1571313
CRIMINAL DAMAGE      905329
OTHER OFFENSES       845339
NARCOTICS            645110
SEX CRIMES            82528
Name: Primary Type, dtype: int64
```

Fuente: Elaboración propia

Además, en la ilustración 12se muestra de forma más visual el porcentaje que toma cada grupo de crimen. Una de las categorías relevantes, será la relacionada con drogas que más adelante se explicará su relevancia y relación directa con el crimen.

Ilustración 12.Representación de la agrupación PrimaryType



Fuente: Elaboración propia

b. ***Location Descripción***: Existen múltiples escenarios donde se puede cometer un crimen, por lo que esta variable contiene múltiples valores distintos, lo que supone un problema a la hora de trasformar la variable a numérica. Por ello, se han agrupado las localizaciones. En la ilustración 12, se muestran los datos agrupados de la forma más precisa.

Ilustración 13.Porcentaje de crímenes cometidos en una determinada zona (agrupados)

	Location Description	Crime_count	Percentage(%)
12	STREET	475000	73.630854
9	RESIDENCE	56842	8.811210
8	PUBLIC BUILDING/GROUNDS	49093	7.610020
13	VEHICLE	19475	3.018865
5	OTHER	13393	2.076080
11	SCHOOL	11505	1.783417
10	RETAIL OUTLET	9389	1.455411
6	POLICE BUILDING	4882	0.756770
1	BUSINESS	2766	0.428764

Fuente: Elaboración propia

7.4. Análisis exploratorio de las variables

A continuación, se implementará una investigación más exhaustiva de las variables consideradas clave a la hora de generar la predicción. Se ha decidido que la forma más simple y rápida de estudiar esas variables, es representándolas de forma visual, para ver cómo evolucionan y se desarrollan. Por ello, se han formulado una serie de preguntas que todo analista de datos o persona interesada en el crimen se formularía.

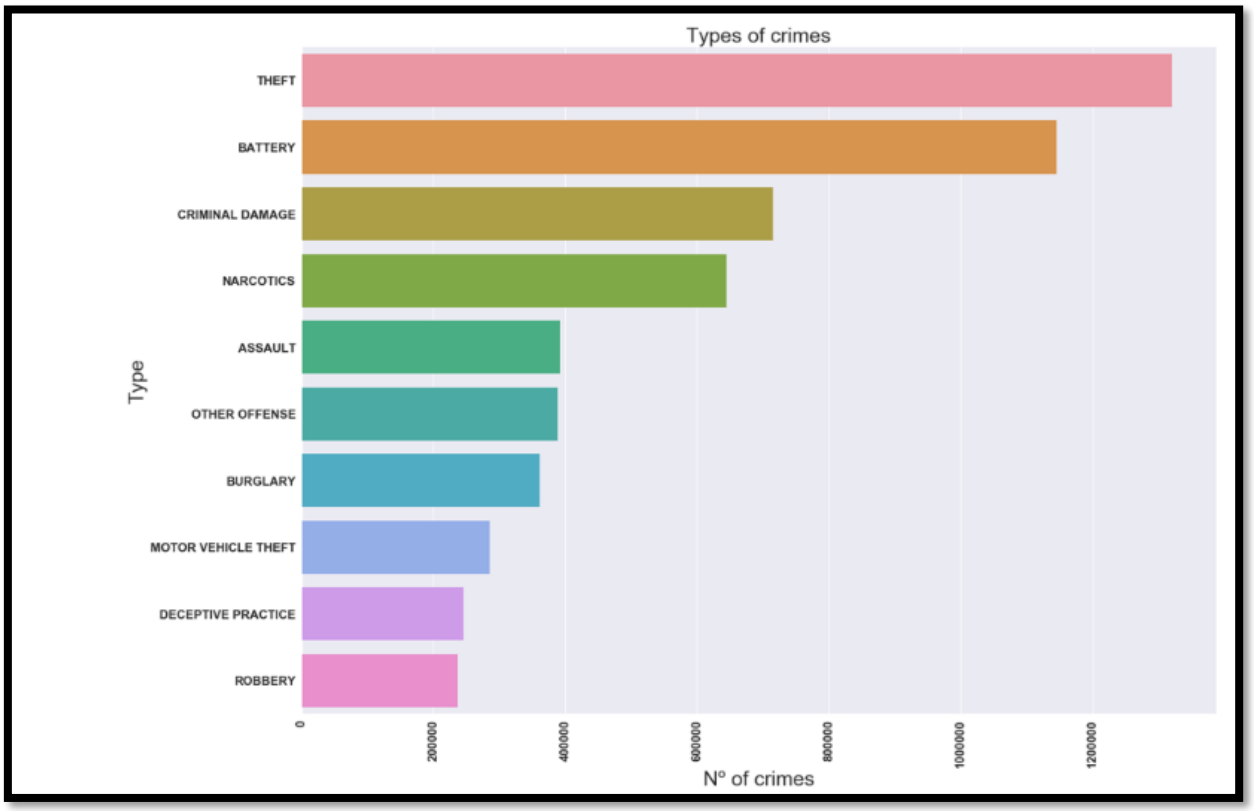
Además, será necesario citar que gracias a este estudio exploratorio se han detectado trasformaciones de datos relevantes que se han detallado en el apartado de limpieza.

Pregunta 1. ¿Cuáles son los delitos o crímenes más frecuentes en Chicago?

En la imagen inferior, se puede observar que los delitos que toman más fuerza durante el año 2001-2019 son aquellos relacionados con el robo y las peleas violentas. Esto, se debe a que Chicago cuenta con una legislación diferente a las de otros países, como los de la Unión Europea.

En este territorio estadounidense y en muchos otros, la tendencia de armas es algo usual. Esta legislación sumada con la diversidad cultural del país, genera constantemente conflictos sobre todo en los barrios donde la renta anual es reducida.

Ilustración 14.Tipos de crímenes

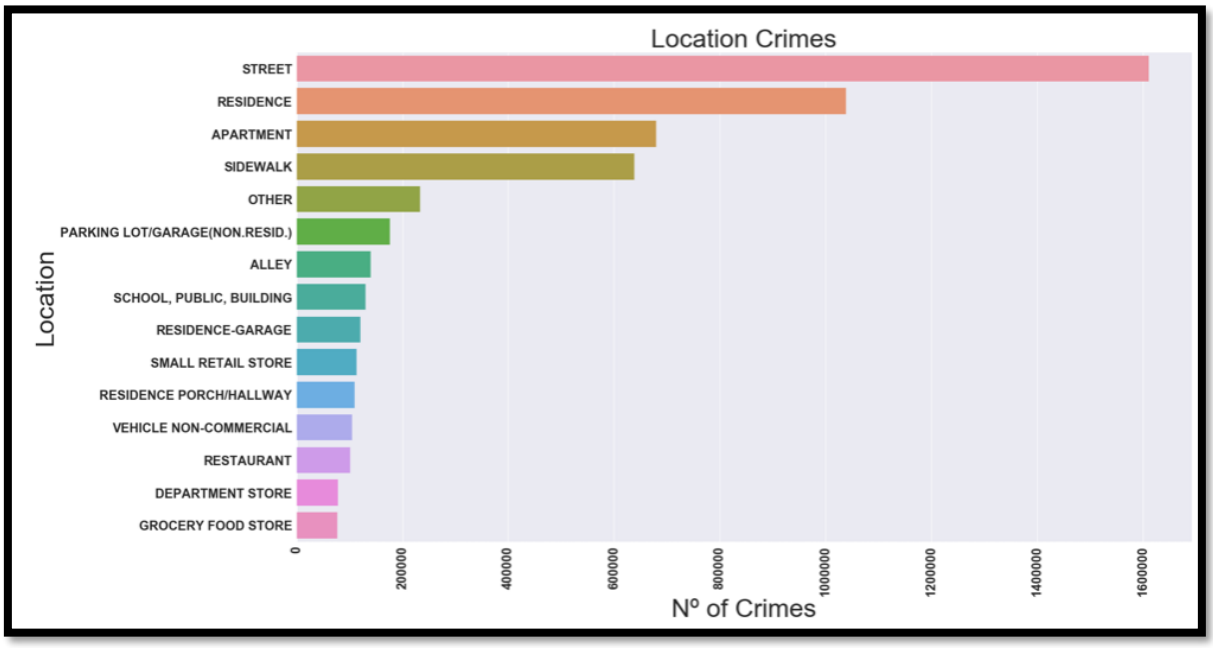


Fuente: Elaboración propia

Pregunta 2. ¿Dónde se produce la mayoría de delitos o crímenes?

En la siguiente fotografía, se observa un punto de inflexión bastante relevante si lo se compara con otros países o continentes. Y, es que el segundo y tercer lugar donde se cometen más delitos son las residencias y apartamentos. Esto, tiene una explicación bastante importante y es que en algunas zonas de EEUU existe el concepto de propiedad privada, por lo tanto, en el momento que se invada la propiedad de una persona, cualquier acto estará totalmente justificado. Por ello, se registran la mayor cantidad de actos violentos o delictivos en estos lugares. Aunque existe otro punto, que está cobrando bastante fuerza en los últimos años y es la violencia de género que sufren una gran cantidad de mujeres.

Ilustración 15. Localización de crímenes

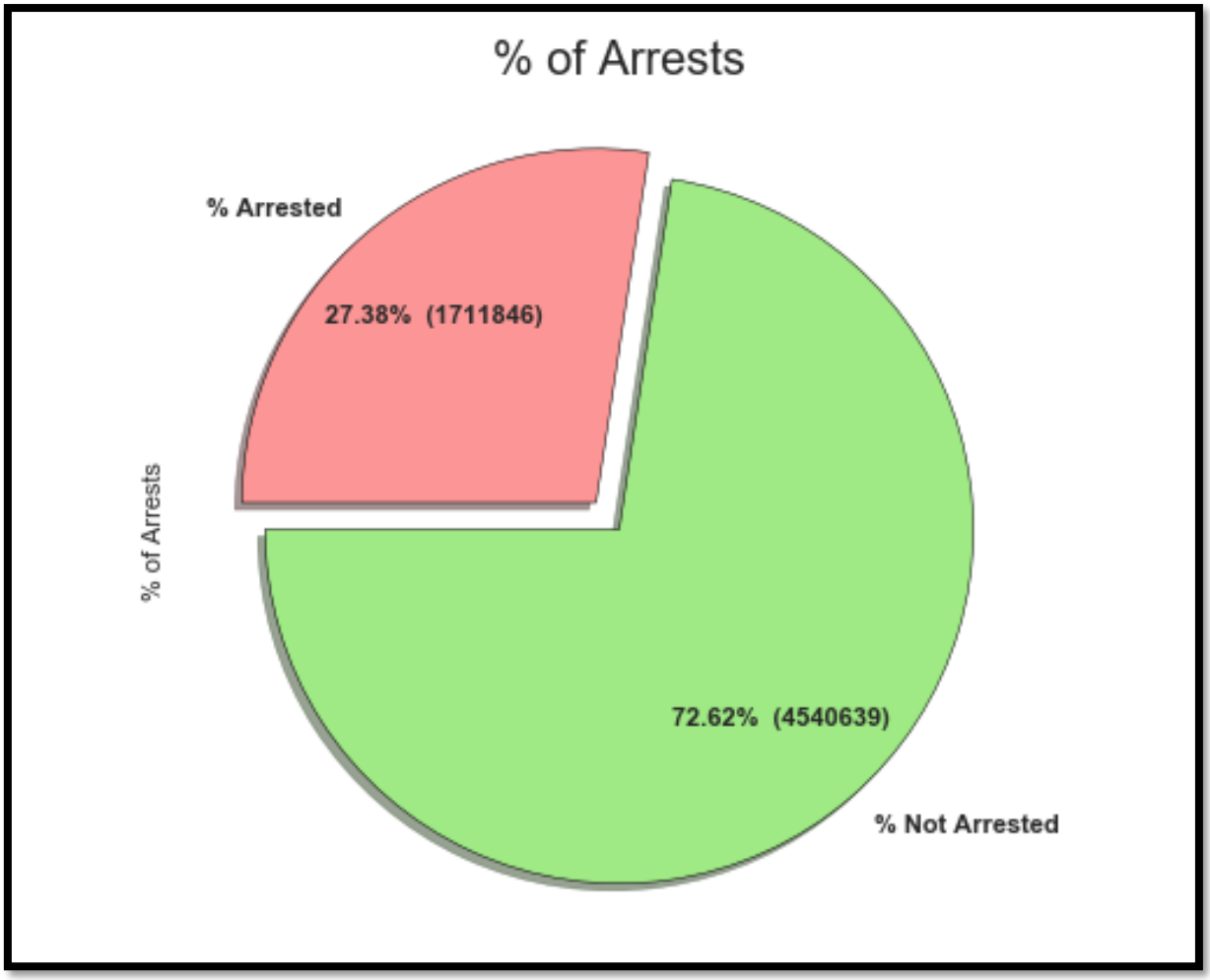


Fuente: Elaboración propia

Pregunta 3. ¿Qué porcentaje de criminales consigue arrestar la policía de Chicago?

A simple vista, se puede observar que la policía de chicago consigue arrestar un pequeño porcentaje de las personas que cometen crímenes. Esta, es una de las causas principales por las que se quiere implementar este proyecto. Con la intención de ayudar a la policía mediante las nuevas tecnologías Big Data que están aconteciendo en la nueva era digital, para gestionar recursos de forma más rápida y anticiparse a los hechos.

Ilustración 16. Porcentaje de detecciones realizadas por la policía



Fuente: Elaboración propia

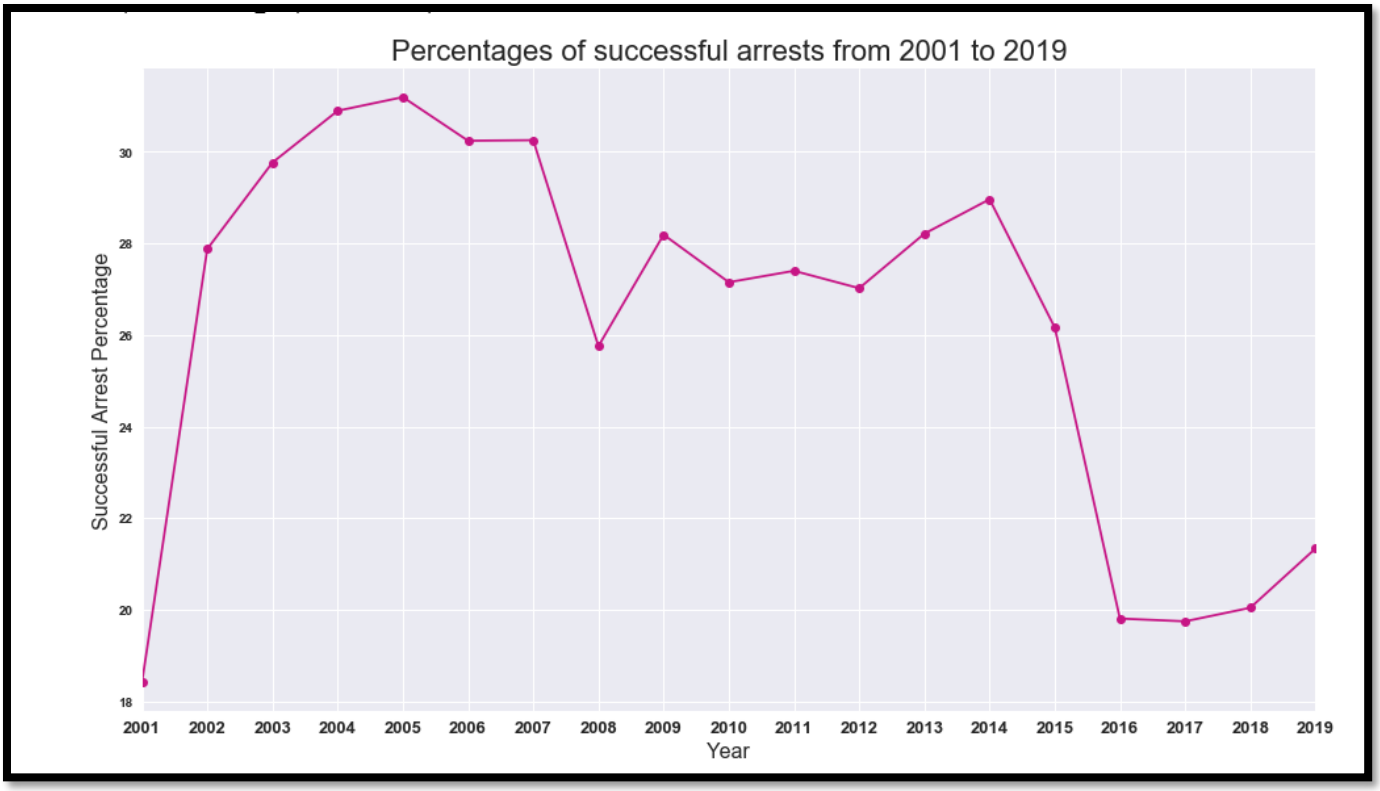
Pregunta 4. ¿Cómo ha ido evolucionando el crimen a lo largo de los años en la ciudad de chicago?

En la ciudad de Chicago, el crimen ha ido variando en función de la situación que ha ido viviendo la ciudad metropolitana. A simple vista, se observa una tendencia elevada en los primeros años, aproximadamente entre 2001-2007, a partir de ahí, la tendencia se empieza a estabilizarse. Esto, coincide con el crecimiento y la inversión que se depositan en los cuerpos de seguridad de la ciudad de Chicago, generando un control del crimen en los siguientes años.

Además, se estima que los últimos años se experimenta un crecimiento del crimen que coincide con el nombramiento del presidente **Donald Trump**. Presidente que promueve la desigualdad racial y fomenta los actos violentos en los ciudadanos estadounidenses.

Nuestra intención, es conseguir predecir cómo se ira desenvolviendo esta tendencia o esta línea en los años futuros.

Ilustración 17. Evolución del crimen



Fuente: Elaboración propia

Pregunta 5. ¿Cuáles son los distritos policiales más perjudicados por la delincuencia?

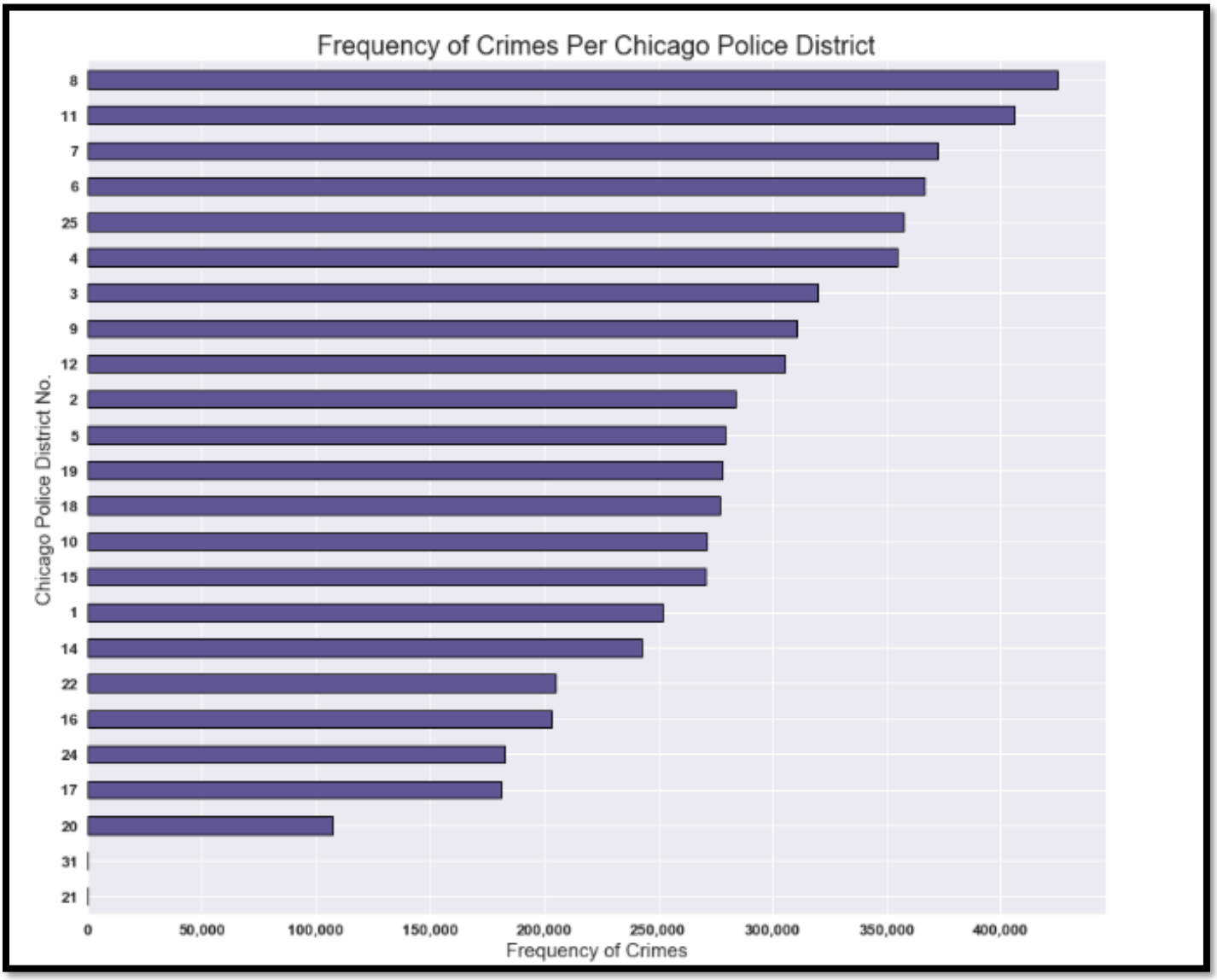
En la imagen posterior, se puede detectar que existen distritos que se encuentran gravemente afectados por la delincuencia y otros en los que apenas se registra ningún acto violento. A continuación, se muestran los distritos más afectados y viceversa:

- Las zonas que registran *mayor* nivel de crímenes hacen referencia a los distritos 8 y 11 que engloban zonas como South Shore, Chatham, Calumet Heights, Pullman,

Avalon Park, Burnside, South Chicago, Bridgeport, Canaryville, ArmourSquare, Pilsen, UniversityVillage.

- Las zonas ***que apenas registran crímenes*** hacen referencia a los distritos número 31 y 21. Estos distritos engloban zonas como Auburn Gresham, Washington Heights, Gresham, Chatham, Roseland, Hermosa, Belmont Cragin, Logan Square

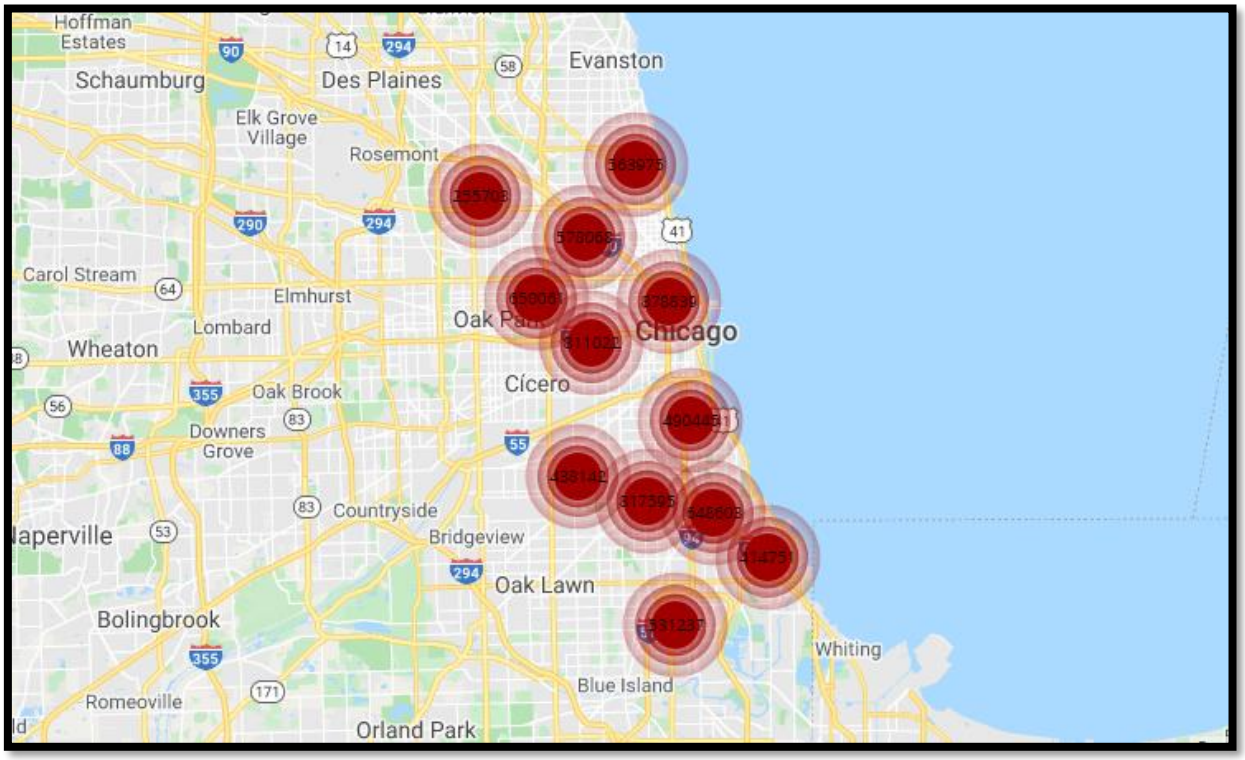
Ilustración 18. Crímenes por distritos



Fuente: Elaboración propia

Además, para tener una idea de cómo están distribuidos los distritos con mayor tasa de crímenes, se ha usado un mapa con puntos de calor para representarlos de forma más visual, en la ilustración 19 se puede observar.

Ilustración 19. Mapa representativo de las zonas con más crimen



Fuente: https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present-Map/c4ep-ee5m

En la imagen superior, se observa que los crímenes principalmente se centran zonas de costa. Dentro de estas zonas existen una serie de factores que influyen como el nivel de renta, las zonas donde predominan clubs nocturnos, abundante cumulo de gente y diversidad cultural.

Todos estos factores, se encuentran entre las causas más influyentes a la hora de evaluar porque en un distrito existe más crimen que en otro. Un caso particular, es que los robos predominan en zonas ricas donde el nivel de vida es mucho más elevado.

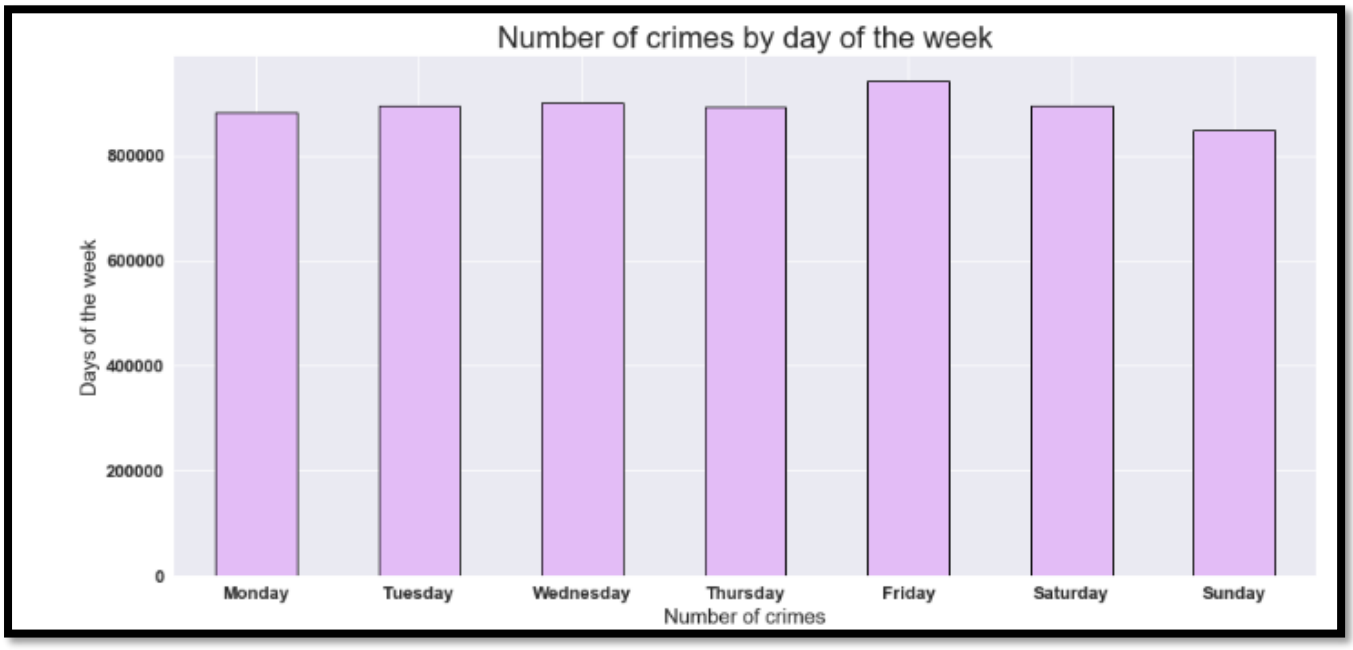
Pregunta 6. ¿Cuáles son los periodos más propensos a producirse crímenes?

Esta pregunta, se consigue visualizar gracias a una trasformación elaborada del formato fecha.

Se ha realizado esta acción porque según un estudio **llevado a cabo por Simon A. Levin**. "Descubrimos que la mayoría de los delitos en Chicago tenían **patrones muy distintos según la época del año**, hora del día, día de la semana"[11]

A simple vista se observa en la ilustración 20, que la mayoría de crímenes se producen los fines de semana más detalladamente entre el viernes y el sábado que es cuando más gente sale y suele cobrar.

Ilustración 20. Día de la semana donde se registran más crímenes

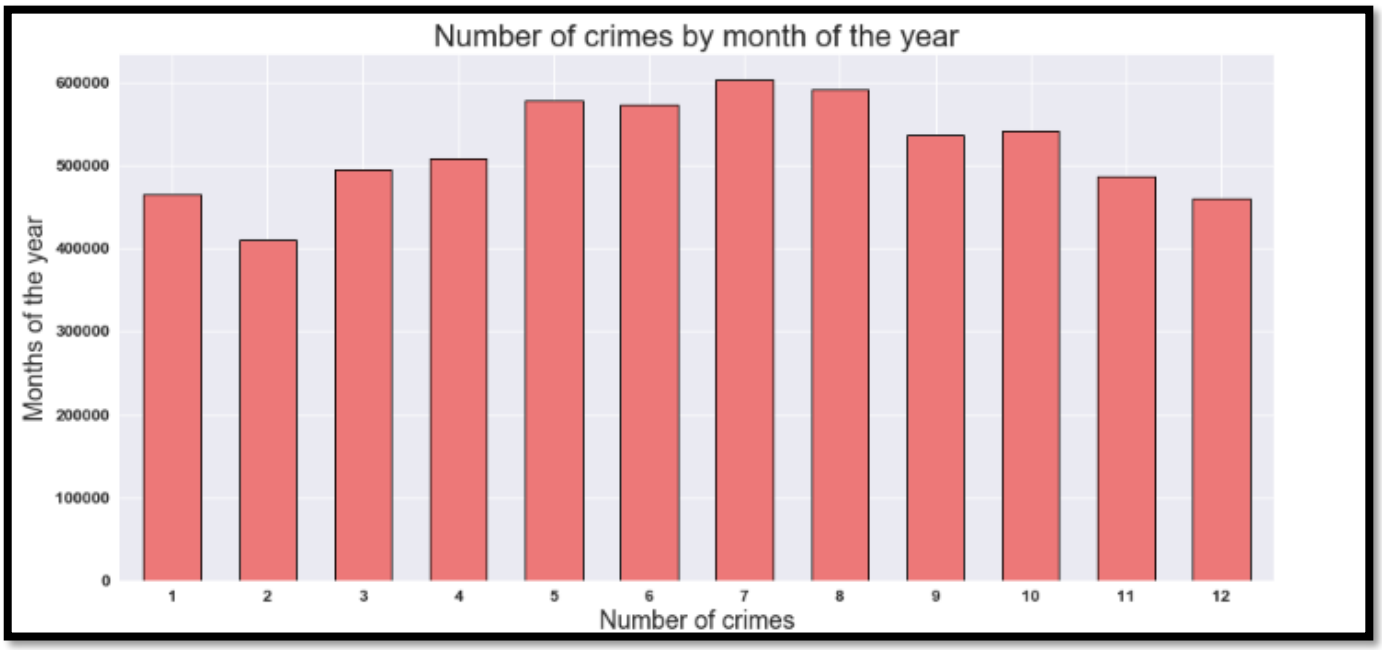


Fuente: Elaboración propia

Además, también se ha comprobado que las altas temperaturas incitan a que se produzcan actos violentos, esto coincide con la representación de la imagen inferior. Los meses donde se registran mayores oleadas de crímenes son los calurosos.

[11] https://www.elespanol.com/ciencia/investigacion/20181218/dias-calendario-crimenes-violentos/352715069_0.html

Ilustración 21.Dia del mes donde se registran más crímenes



Fuente: Elaboración propia

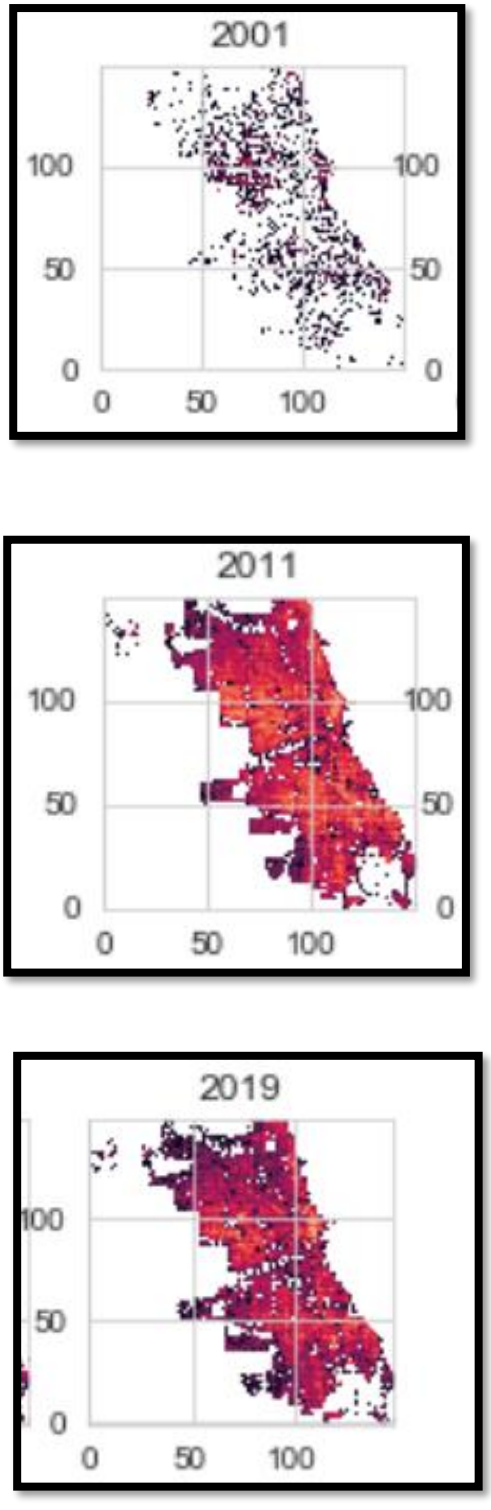
Por lo tanto, si mezclamos altas temperaturas con zonas de playa más los fines de semana se produce una tormenta perfecta para que se desencadenen una gran cantidad de actos delictivos.

Pregunta 7. ¿Cómo han ido evolucionando los crímenes a lo largo de los años en función de las zonas geográficas?

Gracias a esta representación elaborada con una librería de Python, se puede representar como los crímenes han ido variando a lo largo de los años de una zona a otra. En el notebook, se muestran todos los años, pero aquí, se han querido mostrar los más relevantes en el tiempo que son los años 2001, 2015, 2019.

Existe una breve apreciación, la cual es la poca cantidad de crímenes que existían en el año 2001 esto se puede achacar a dos acciones: todavía no habían surgido las oleadas de crímenes o los registros no eran de todo fiables en ese año, debido a que empezaban a implantar las herramientas de control.

Ilustración 22. Mapa de calor del año 2001,2011,2018



Fuente: Elaboración propia

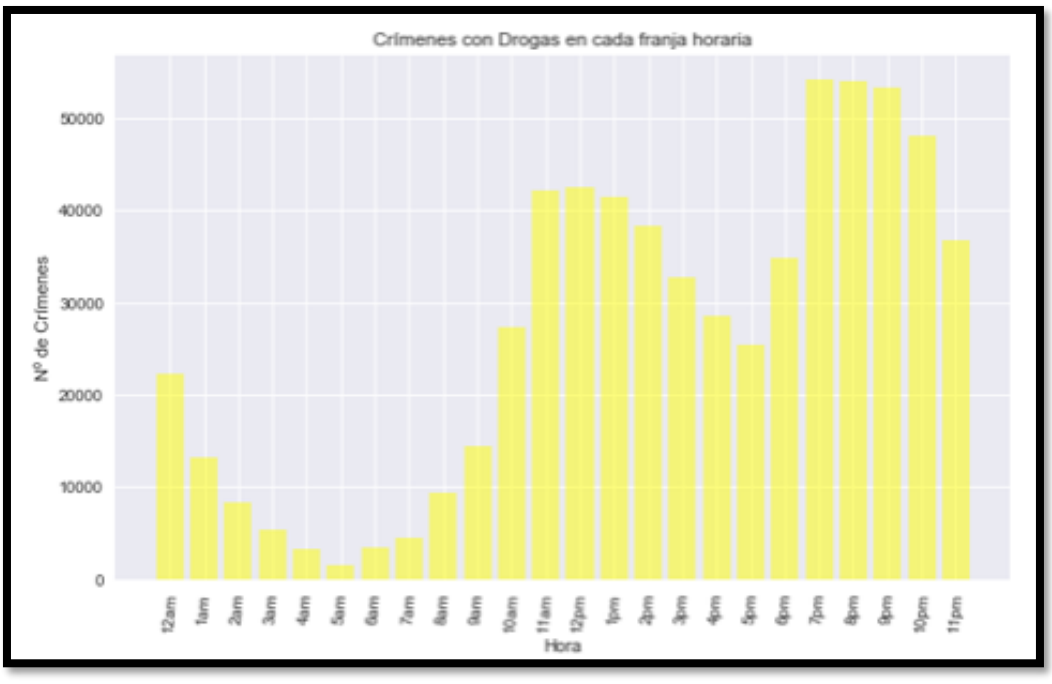
Para concluir con las visualizaciones, se quiere tocar el crimen desde una variable que se encuentra directamente relacionada que es "***consumo de drogas***". Según estudios, el consumo de drogas incrementa los actos violentos de las personas, debido a que se inhibe la capacidad de raciocinio de aquellas que consume esta sustancia. Además, el trasporte de este tipo de estupefacientes trae malas y peligrosas influencias a largo plazo.

Por ello, se ha generado un dataframe que recoge todos los crímenes de la PrimaryType que contengan la palabra "droga" o sustantivos de ella. Este dataframe, se va a representar para sacar conclusiones a través de una serie de preguntas.

Pregunta 8. ¿A qué hora se producen mayor cantidad de crímenes relacionados con las drogas?

Como es usual, los crímenes relacionados con drogas se promueven por las noches acompañado fiestas y se incrementa sobre todo a altas horas de la madrugada. Esto muestra una relación dispersa con los crímenes que se comenten. La mayoría son por las noches, cuando la ciudad de chicago está dormida, como se representa en la ilustración 23.

Ilustración 23. Crímenes por drogas según la hora

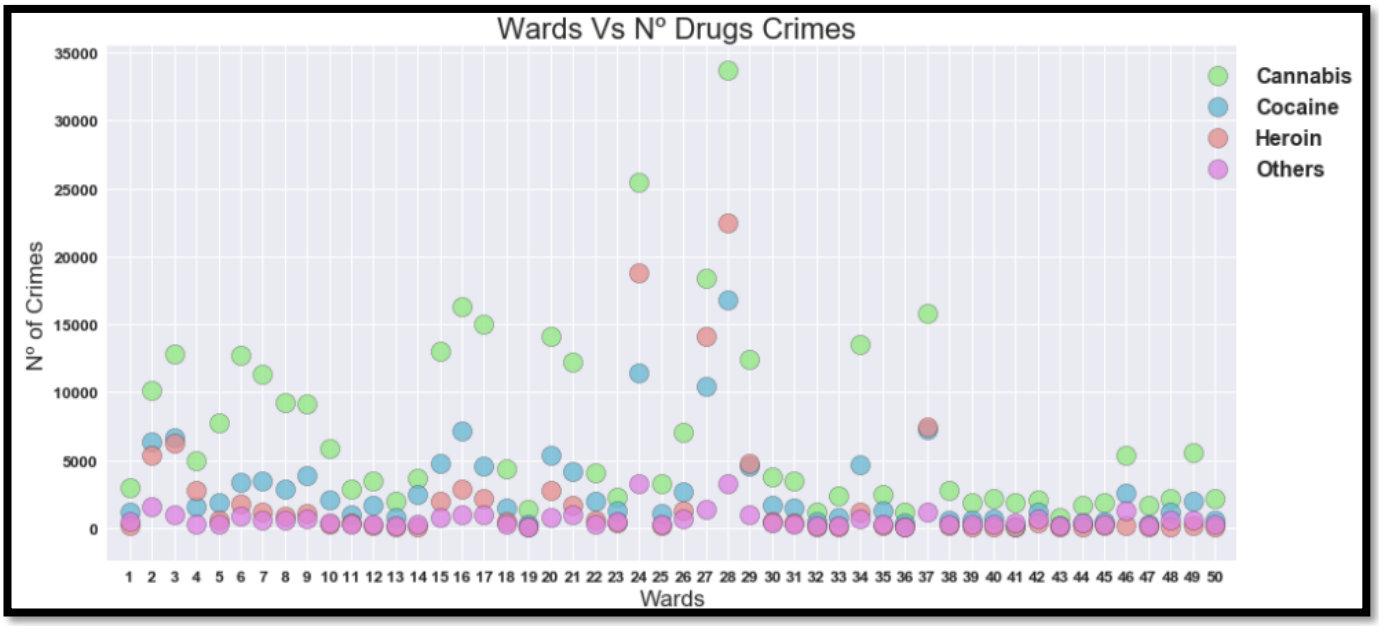


Fuente: Elaboración propia

Pregunta 9. ¿Los crímenes relacionados con drogas donde se producen?

El consumo o trasporte de drogas van variando de zonas a lo largo del tiempo, por ello se quiere representar una visualización que nos muestre cuales son las drogas que más se consumen y los distritos donde se produce más crímenes relacionados con esta variable o fenómeno.

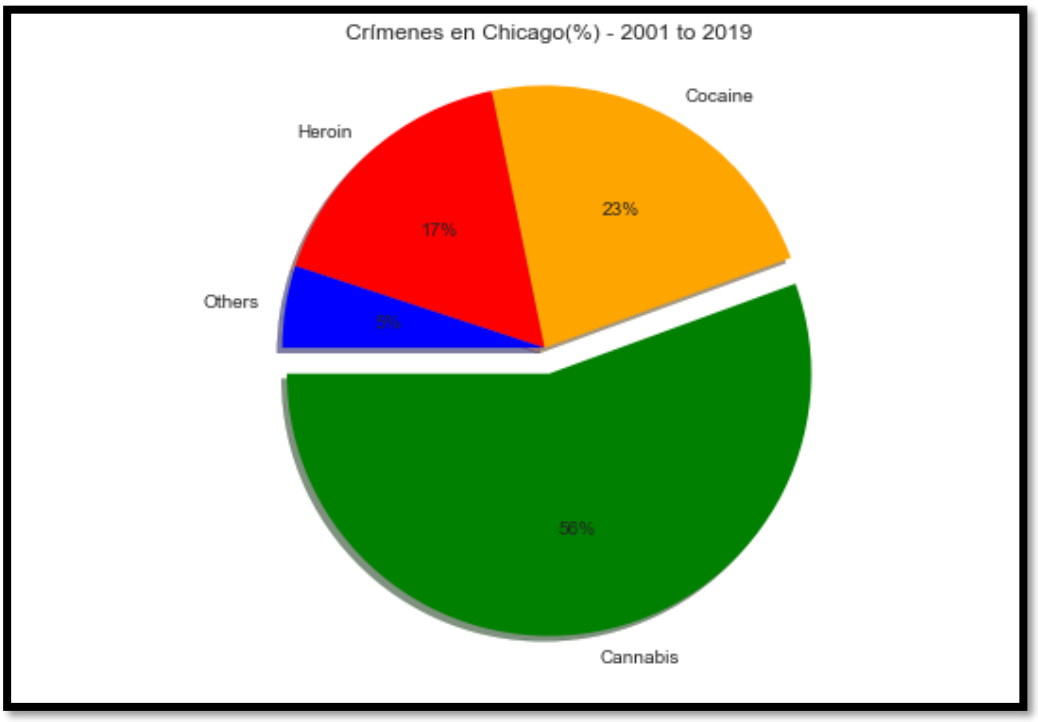
Ilustración 24. Tipos de drogas por distritos



Fuente: Elaboración propia

En la ilustración 25, se muestra que las drogas con mayor tendencia a consumirse en la ciudad de chicago. Encontrando, el cannabis como principal droga consumida por la sociedad estadounidense, seguido de la cocaína debido a su fácil acceso en la sociedad en la que vivimos.

Ilustración 25. Tipos de drogas comunes en los crímenes.



Fuente: Elaboración propia

7.5. Predicción de crimen

Para finalizar con el desarrollo, se hablará de las variables elegidas y los modelos de Machine Learning utilizados para intentar alcanzar una predicción del crimen en Chicago.

7.5.1. Selección de variables

A lo largo del estudio, se han ido representando las diferentes variables con las que se contaba, un total aproximado de 30. De todas ellas, solo se han seleccionado las que se consideran más relevantes, según algunos estudios exploratorios previos con la intención de generar un modelo u algoritmo preciso y lógico.

Estas variables seleccionadas se agrupan generalmente en dos grupos; las variables categóricas y las variables numéricas. Considerando que NUM_FEATURES hace referencia a las variables continuas o numéricas y CAT_FEATURES hace referencia a las variables categóricas. En la ilustración 26, se puede ver más detalladamente.

Ilustración 26. Selección de variables categóricas y numéricas

```
# Seleccion de variables categoricas y numericas
NUM_FEATURES = ["District", "Ward", "Census Tracts","Police Beats"]
CAT_FEATURES = ["Primary Type", "Location Description", "Arrest"]
```

Fuente: Elaboración propia

Para entender un poco más la división entre variables categóricas y numéricas, se van a mostrar los tipos de datos que contiene cada uno de estos grupos. Teniendo en cuenta que las variables categóricas, son aquellas que posteriormente van a necesitar una trasformación mediante alguna "técnica" de las aprendidas en clase para generar valores numéricos y conseguir una implementación óptima en el modelo.

Ilustración 27.Tipos de datos de las variables numéricas y categóricas

```
dataset[NUM_FEATURES].dtypes

# Tipos de datos de las variables numericas

District         float64
Ward             float64
Census Tracts    float64
Police Beats     float64
dtype: object
```

```
dataset[CAT_FEATURES].dtypes

#Tipos de datos de las variables categoricas

Primary Type           object
Location Description   object
Arrest                   bool
dtype: object
```

Fuente: Elaboración propia

Previo a introducirnos dentro de los algoritmos de predicción, tenemos que preparar el modelo eficientemente para su interpretación, y para ello necesitamos limpiar 2 variables categóricas, las cuales consideramos que poseen información muy relevante para el modelo. Estas variables son **Primary Type** y **Location Description**.

Nosotros al obtener los datos a través de la API, nos encontramos con que estas variables poseen multiples registros para un mismo significado. Con lo cual, nosotros queremos unificar esos registros y dejar la variable fácilmente interpretable para nuestro modelo.

Nuestra variable **Primary Type** tiene los siguientes registros:

```
#Estas son todos los resultados que tenemos como valores unicos en los distintos tipos de crimenes.
dataset['Primary Type'].value_counts()
```

```
THEFT                                1321165
BATTERY                              1144756
CRIMINAL DAMAGE                       715245
NARCOTICS                             644940
ASSAULT                               391916
OTHER OFFENSE                         388135
BURGLARY                              360567
MOTOR VEHICLE THEFT                   285123
DECEPTIVE PRACTICE                    245009
ROBBERY                               236745
CRIMINAL TRESPASS                     179862
WEAPONS VIOLATION                      69535
PROSTITUTION                           60262
PUBLIC PEACE VIOLATION                 45259
OFFENSE INVOLVING CHILDREN             42193
CRIM SEXUAL ASSAULT                    24849
SEX OFFENSE                            22259
INTERFERENCE WITH PUBLIC OFFICER       15747
GAMBLING                               13291
LIQUOR LAW VIOLATION                   12095
ARSON                                  10031
HOMICIDE                                9452
KIDNAPPING                              5583
INTIMIDATION                            3662
STALKING                                3242
OBSCENITY                                573
CONCEALED CARRY LICENSE VIOLATION        438
NON-CRIMINAL                             159
PUBLIC INDECENCY                         156
OTHER NARCOTIC VIOLATION                 118
HUMAN TRAFFICKING                         57
NON - CRIMINAL                            38
RITUALISM                                 14
NON-CRIMINAL (SUBJECT SPECIFIED)           9
Name: Primary Type, dtype: int64
```

Fuente: Elaboración propia

Nosotros no podemos trabajar con una variable tan importante con tantísimos registros, por lo tanto realizamos una transformación de los datos, agrupándolos en 6 diferentes grupos: THEFT & ROBBERY, ASSAULT, DAMAGE, OTHER OFFENSES, NARCOTICS y SEX CRIMES.

```
#Aquí vemos como la agrupación llevada a cabo ha sido correcta y tenemos solo nuestras 6 categorías.
dataset['Primary Type'].value_counts()
```

```
THEFT & ROBBERY    2203600
ASSAULT            1570973
CRIMINAL DAMAGE     905138
OTHER OFFENSES      845195
NARCOTICS           645058
SEX CRIMES           82521
Name: Primary Type, dtype: int64
```

Ahora vamos a ver como es nuestra variable **Location Description**:

```
# Aquí obtenemos el numero total de delitos en función de su localización.

dataset["Location Description"].value_counts()
```

```
STREET                    1611534
RESIDENCE                 1039117
APARTMENT                  679996
SIDEWALK                   638461
OTHER                      232847
                           ...
POOLROOM                        1
EXPRESSWAY EMBANKMENT           1
LAGOON                          1
LIVERY AUTO                     1
FUNERAL PARLOR                  1
Name: Location Description, Length: 179, dtype: int64
```

Fuente: Elaboración propia

Como podemos apreciar en esta ilustración, llegamos a tener 179 tipos de registros diferentes en nuestra variable. Es un tedioso proceso, porque tenemos que cambiar todas las variables y agruparlas de forma que no variemos los valores originales.

Finalmente agrupamos las variables en los siguientes 12 grupos:

```
# Comprobamos que se ha realizado correctamente.

dataset["Location Description"].value_counts()
```

```
STREET                     2391741
RESIDENCE                  2090360
PUBLIC BUILDING/GROUNDS     408717
RETAIL OUTLET               407859
BUSINESS                    290704
OTHER                       249178
SCHOOL                      181988
VEHICLE                     142755
HOSPITAL                     32395
POLICE BUILDING              16378
CHURCH                       13910
FEDERAL BUILDING             12691
AIRPORT                      11815
POLICE/FIRE STATION           1994
Name: Location Description, dtype: int64
```

Fuente: Elaboración propia

En la ilustración 28, se mostrarán los datos finales que se van a usar para generar la predicción. Estas constantes, se han elegido tras un elaborado trabajo porque solo existía relación entre las variables que abarcaban la "ubicación" del crimen.

Ilustración 28. Datos finales para generar el algoritmo

	District	Ward	Census Tracts	Police Beats	Primary Type	Location Description	Arrest
0	7.0	16.0	277.0	266.0	ASSAULT	RESIDENCE	True
1	18.0	2.0	17.0	198.0	ASSAULT	RESIDENCE	True
2	3.0	5.0	134.0	262.0	OTHER OFFENSES	PUBLIC BUILDING/GROUNDS	True
3	6.0	8.0	247.0	227.0	CRIMINAL DAMAGE	RESIDENCE	False
4	8.0	23.0	266.0	272.0	ASSAULT	RESIDENCE	False

Fuente: Elaboración propia

Finalmente, para la trasformación de las variables categórica Arrest se ha decidido usar la función o técnica ***"one-hot-encoding.".***Esta transformación asigna únicamente 0 y 1 a las variables con diferentes registros, indicando 1 si la muestra se corresponde con el nivel y 0 en caso contrario.

Es importante explicar porque las variables categóricas "***PrimaryType"*** y ***"LocationDescription"*** no han sido transformadas con el método anteriormente mencionado. Las variables "***PrimaryType"*** y ***"LocationDescription"*** no se han trasformado porque aumentaba en exceso la dimensionalidad del modelo y, por ello, se les asigno un valor sobre 100 en función del número de veces que se repite dicho valor en los datos.

En caso de haber incluido estas dos variables en el método "one-hot-encoding" habríamos convertido un modelo de 7 variables en 24 variables y esto habría ralentizado la obtención de resultados, pues a medida que aumenta la dimensionalidad, y no mejoramos la capacidad de procesamiento de la máquina, habríamos tardado 10 veces más tiempo.

A continuación, en la ilustración 29 se muestra cómo quedarían los datos una vez aplicadas las trasformaciones necesarias.

Ilustración 29. Variables finales después de la trasformación

	District	Ward	Census Tracts	Police Beats	Primary Type	Location Description	Arrest_True
0	7.0	16.0	277.0	266.0	25.13	33.43	1
1	18.0	2.0	17.0	198.0	25.13	33.43	1
2	3.0	5.0	134.0	262.0	13.52	6.54	1
3	6.0	8.0	247.0	227.0	14.48	33.43	0
4	8.0	23.0	266.0	272.0	25.13	33.43	0

Fuente: Elaboración propia

Para finalizar, de las 30 variables que se tenían solo 7 se van a usar en la elaboración de la predicción.

Ilustración 30. Variables finales

```
Index(['District', 'Ward', 'Census Tracts', 'Police Beats', 'Primary Type',
       'Location Description', 'Arrest_True'],
      dtype='object')
```

Fuente: Elaboración propia

7.5.2. Selección de algoritmos y modelos

Una vez seleccionada las variables, será necesario dividir los datos para generar una muestra de entrenamiento y test. Además, será necesario indicar nuestra variable objetivo en este caso es ”District”. Normalmente, la muestra en entrenamiento es mucho mayor a la de test. Por ello, en este caso se ha usado un 75% de test y un 25% de entrenamiento, como se observa en la ilustración 32.

Ilustración 31. División de test y train

```
X = crimen_df.drop("District", axis = 1)
variables = X.columns
X = X.values

y = crimen_df["District"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, random_state=0, test_size=0.25, stratify=y)
```

Fuente: Elaboración propia

Una vez separado los datos por la variable objetivo se ha decidido hacer un análisis de los dos algoritmos a emplear y, tras tener en cuenta lo ya aplicado y los conocimientos obtenidos en el máster se ha considerado emplear el ***Gradient Boosting*** y ***Ramdom Forest*** porque nuestra intención no es comparar variables sino clasificar los valores con los que contamos para predecir cuales son los distritos donde se cometen más crímenes.

Finalmente se ha usado un modelo de **regresión lineal** para mostrar la diferencia que existe de forma visual entre algoritmos clasificadores y comparativos.

Antes de empezar con los algoritmos, se debe señalar la implementación de técnicas o métodos para generar un modelo más óptimo o con menos fallos. Estas técnicas que se han usado reciben nombre de *Stepwise* y *optimización de hiperparámetros*.

Entendiendo ***Stepwise*** como un método que selecciona características de forma interactiva. Esto, consiste en que, si se elimina una de las variables, de forma rápida se comprueba la calidad de los modelos y se selecciona el mejor. En caso, de que el proceso sea el original termina, en caso de que no se de esa condición, el proceso volverá a empezar.

Y, la ***optimización de hiperparámetros*** *se usará el modelo* ***GridSearch*** y se importará la función ***extract_results*** *para ver los resultados estadísticos, como la media o la varianza de la muestra, tanto el set de entrenamiento como de test.*

En cuanto a los algoritmos implementados, se destacan los siguientes:

a) ***Regresión Lineal***

La regresión lineal es un modelo supervisado que representa básicamente las relaciones. En función del número de atributos se tendrá un plano o una recta. En este caso, se tiene un plano porque tenemos más de un atributo como se ha visto en los pasos anteriores. Este algoritmo, en el proceso de entrenamiento busca parámetros que reducen el error total cometido por el conjunto de variables que contiene el modelo, por ello es necesario definir una función que mida el error.

Ilustración 32. Regresión lineal

```
# Entrenamos y obtenemos las predicciones del modelo:

lr = LinearRegression(normalize = True)

lr.fit(Xrl_train, yrl_train)

y_hat_train = lr.predict(Xrl_train)

y_hat_test = lr.predict(Xrl_test)
```

Fuente: Elaboración propia

b) Gradient Boosting

Gradient Boosting es un algoritmo automático muy potente porque combina aprendizaje débil para crear un modelo predictivo sólido. Este tipo de modelos, está tomando mucha relevancia debido a su eficacia a la hora de clasificar conjuntos de datos complejos.

Ilustración 33. Algoritmo de Gradiente Boosting

```
from sklearn.ensemble import GradientBoostingClassifier

Xgb_train=X_train
ygb_train=y_train
Xgb_test=X_test
ygb_test=y_test
variables_gb = variables

list_estimators = (10, 20, 30)

for n_estimators in list_estimators:
  gb = GradientBoostingClassifier(
      n_estimators=n_estimators,
      random_state = 0
  )
  gb.fit(Xgb_train, ygb_train)
```

```
for name, importance in zip(variables_gb, gb.feature_importances_):
  print(f"{name}: {importance:.2f}")
```

Fuente: Elaboración propia

c) ***Ramdom Forest***.

Ramdom Forest como su propio nombre indica consiste en una cantidad de árboles individuales que funcionan como un grupo o conjunto. Cada uno de estos árboles saca las predicciones aleatorias con más votos convirtiéndola en la nuestro modelo.

Ilustración 34. Algoritmo de Ramdom Forest

```
Xrf_train=X_train
yrf_train=y_train
Xrf_test=X_test
yrf_test=y_test
variables_rf = variables

list_estimators = (1, 50, 100)

for n_estimators in list_estimators:
  rf = RandomForestClassifier(
      n_estimators=n_estimators,
      max_depth=50,
      random_state = 0
  )
  rf.fit(Xrf_train, yrf_train)
```

Fuente: Elaboración propia

Todos los resultados obtenidos de los siguientes modelos, se reflejarán en el punto de soluciones y en el notebook adjunto a esta memoria.

8. Visualización de Información

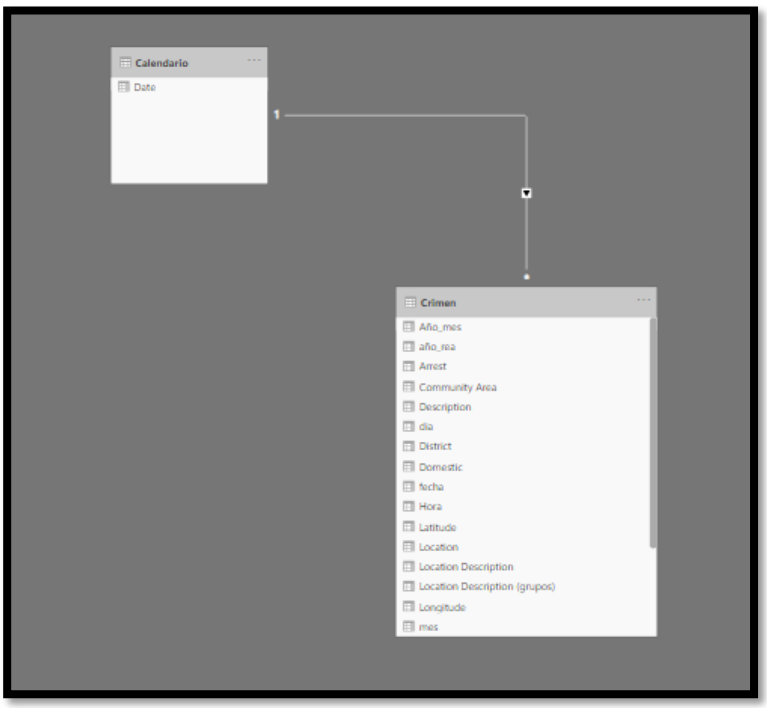
Para concluir con el apartado de desarrollo, se mostrarán los datos en una plataforma visual llamada ***Power BI***, con la intención de generar paneles personalizados que ayuden a entender los datos de forma más dinámica mediante una serie de filtros. Estos filtros, pretenden adaptar la información en función del rol que desempeñe cada usuario que consuma el panel.

8.1 Cuadro de mandos

Esta herramienta se conecta a la carpeta donde se encuentran los datos. Además, esta tecnología cuenta con una opción de trasformación que adapta los datos de forma genérica, como ejemplo establecer encabezados de las columnas. En este caso, se ha realizado la trasformación manual porque contamos con millones de datos y sabemos exactamente los cambios que hay que realizar.

Además, se genera un modelo lógico simple en forma de estrella donde se mostrarán las tablas y la carpeta de métricas elaborada por nosotros. Entre las tablas, se cuenta con dos principalmente. La tabla ***"crímenes***" que contiene los datos y la tabla ***"calendario"*** generada por la métrica de ***CALENDARAUTO()*** para contar con una tabla de tiempo. Todo esto, se muestra en la ilustración 33.

Ilustración 35. Modelo Lógico Power BI



Fuente: Elaboración propia

Una vez relacionado los datos, se han elaborado una serie de métricas como “Total de crímenes” o “fecha máxima o mínima de los datos” para mostrarla en nuestro cuadro de mandos, gracias a las fórmulas DAX.

A continuación, se muestra un formula DAX implementada en nuestro cuadro de mandos para sacar el total de crímenes:

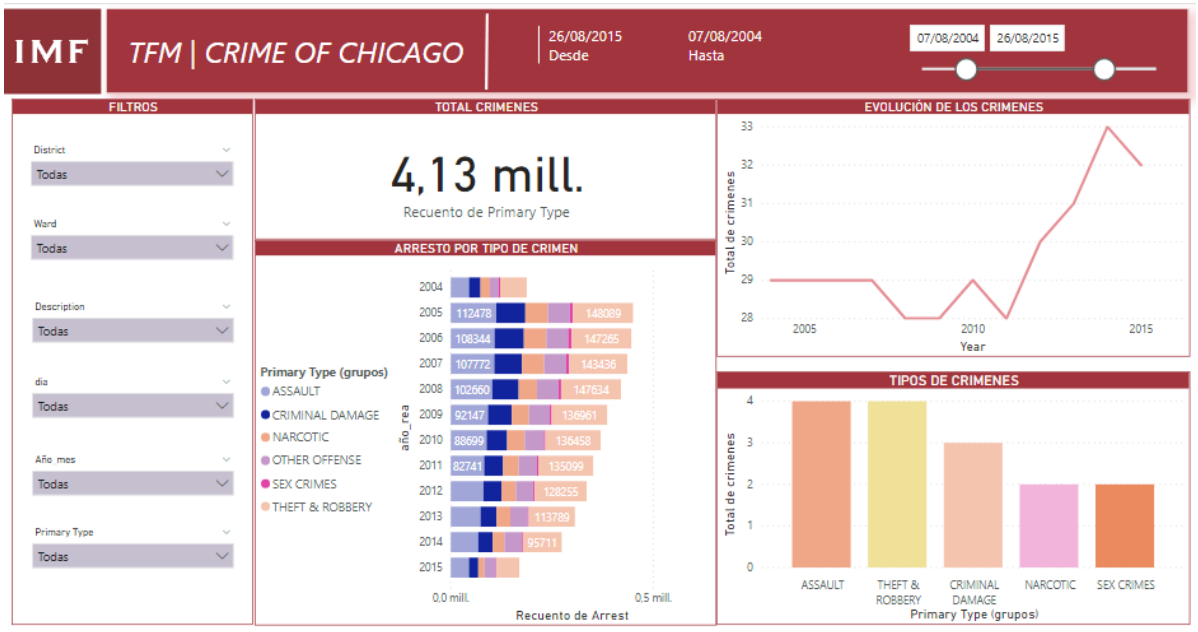
Ilustración 36.Formula DAX para total de crímenes

```
Total de crimenes = DISTINCTCOUNT('Crimen'[Primary Type])
```

Fuente: Elaboración propia

En la ilustración 35, se muestra el cuadro de mandos final. En este cuadro, se puede observar el total de crímenes que han acontecido en un periodo determinados, su evolución y los más frecuentes. A demás de contar en la parte izquierda de múltiples filtros en función de lo que interese, en cada momento.

Ilustración 37. Cuadro de Mandos



Fuente: Elaboración propia

Este cuadro de mandos, podría representarse de forma online o en una aplicación web para ayudar a los departamentos de policía a organizarse en función del periodo del año, el lugar e incluso los tipos de crímenes.

III. PARTE DE RESULTADOS

Los resultados que se han obtenido en los modelos de la predicción han sido relevantes y reveladores en cuanto a información del dataset se refiere.

Se ha tenido que analizar muchas veces el dataset con diferentes tipos de variables para encontrar un significado a la hora de interpretar los datos obtenidos. Como primera alternativa se han tomado las variables que se consideraban más relevantes para el dataset tras el análisis exhaustivo llevado a cabo. El dataset final se compone de las siguientes variables mencionadas en el apartado de desarrollo.

Con las variables seleccionadas se ha llevado a cabo el siguiente modelo:

6.1 Resultados de la regresión lineal

Los resultados de las siguientes métricas sobre el modelo de Regresión Lineal entre el set de entrenamiento y el set de test, son muy similares entre sí. Tanto el error cuadrático de la media (MSE), como el error absoluto de la media (MEA), y el error al cuadrado (R2) son prácticamente iguales, revelando que el set de test mejora de manera insignificante el set de entrenamiento, con lo cual, el propio set de entrenamiento serviría para obtener un resultado válido para nuestra predicción.

Ilustración 38. Métricas de la regresión lineal

```
# Calculamos las tres métricas para el set de entrenamiento y test y las mostramos en pantalla

mse_train = mean_squared_error(yrl_train, y_hat_train)
mse_test = mean_squared_error(yrl_test, y_hat_test)

r2_train = r2_score(yrl_train, y_hat_train)
r2_test = r2_score(yrl_test, y_hat_test)

mae_train = mean_absolute_error(yrl_train, y_hat_train)
mae_test = mean_absolute_error(yrl_test, y_hat_test)

print(f"MSE train: {mse_train:.4f} test: {mse_test:.4f}")
print(f"R2 train: {r2_train:.4f} test: {r2_test:.4f}")
print(f"MAE train: {mae_train:.4f} test: {mae_test:.4f}")
```

```
MSE train: 25.3297 test: 25.3156
R2 train: 0.4736 test: 0.4740
MAE train: 3.4401 test: 3.4420
```

Fuente: Elaboración propia

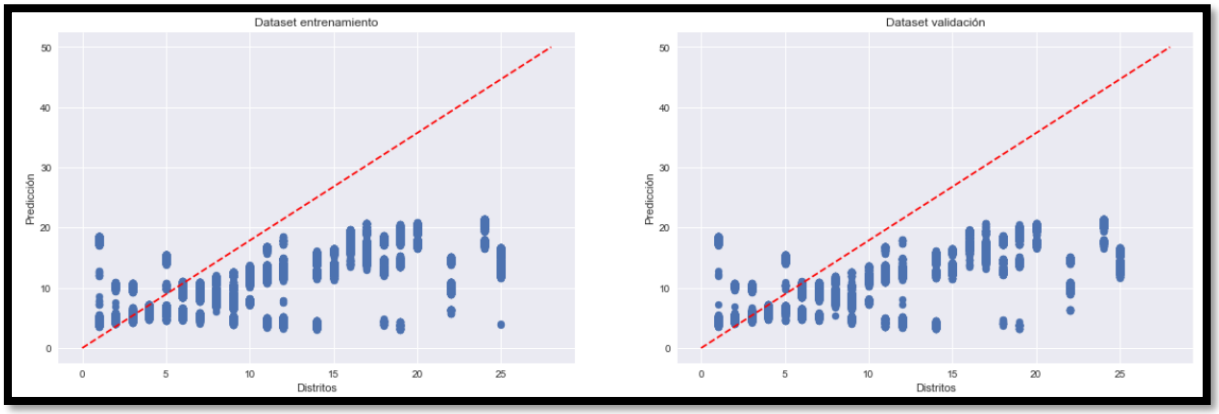
Como se observa en los resultados de las siguientes gráficas, apenas existen cambios entre la predicción del set de entrenamiento y la predicción del set de test. Todo esto, se observa en la ilustración 40.

Ilustración 39. Comandos para generar un gráfico de dispersión

```
# Generamos el gráfico de dispersión y veremos si el set de entrenamiento lo hace mejor que el set de test

fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (20, 6))

ax1.set_title("Dataset entrenamiento")
ax2.set_title("Dataset validación")
for ax in (ax1, ax2):
    ax.set_ylabel("Predicción")
    ax.set_xlabel("Distritos")
    ax.plot(np.linspace(0, 28), np.linspace(0, 50), "r--")

ax1.scatter(y_train, y_hat_train)
ax2.scatter(y_test, y_hat_test)

plt.show()
```

Fuente: Elaboración propia

Ilustración 40. Gráfico de dispersión



Fuente: Elaboración propia

Si se comparan los resultados obtenidos en la predicción de ambos sets, y los resultados que se obtuvieron en los puntos anteriores sobre los distritos más castigados por el crimen. Se puede observar que para distritos como el 8, parece que podría ***descender la criminalidad*** en los próximos años, o distritos como el 1, el cual no se encontraba entre los distritos con mayor crimen, parece que la tendencia que llevara es de un ***aumento en el porcentaje de la criminalidad.***

6.2 Resultados del Gradient Boosting

Este modelo, funciona de forma que a mayor número de árboles individuales menor regularización, tendiendo al overfit cuantos más árboles se añaden. Este modelo ha sido el primero en mostrar que la relevancia que se había intuido en un principio en cuento a las variables categóricas no era la esperada. En la ilustración 41, se puede observar el peso que toma cada variable en función de su relación.

Ilustración 41. Resultados de la relevancia entre variables

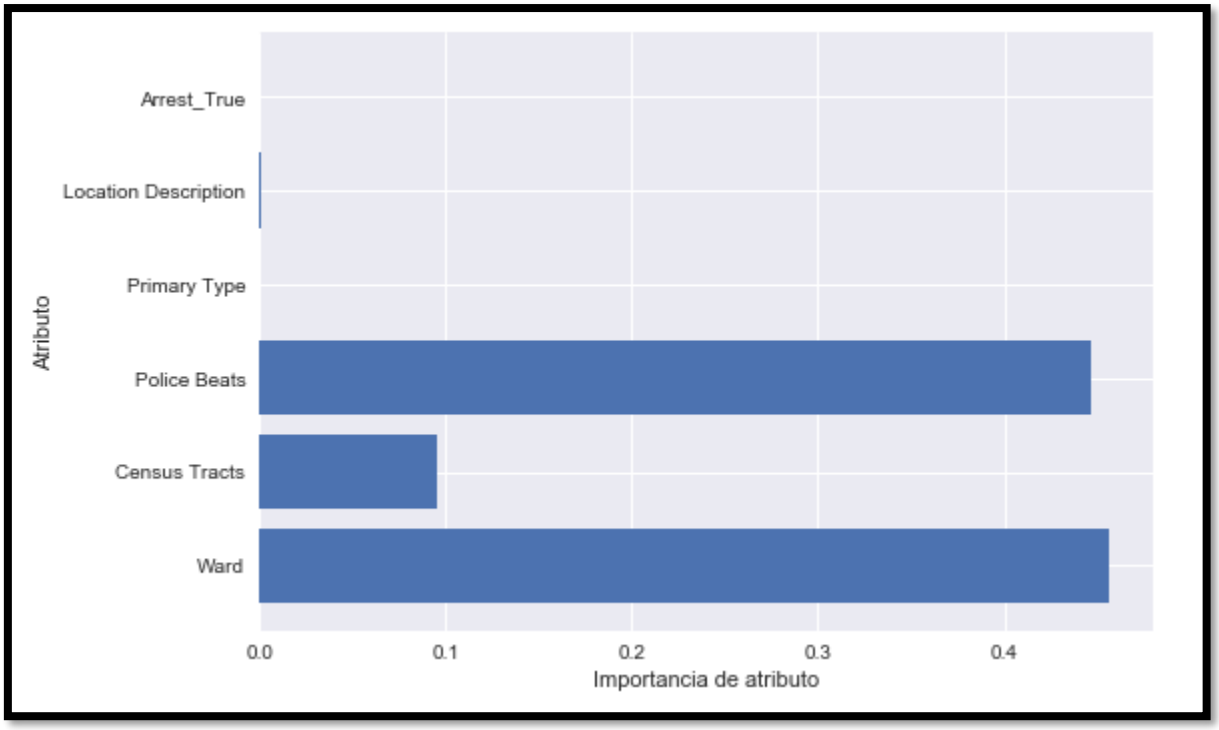
```
for name, importance in zip(variables_gb, gb.feature_importances_):
  print(f"{name}: {importance:.2f}")
```

```
Ward: 0.46
Census Tracts: 0.10
Police Beats: 0.45
Primary Type: 0.00
Location Description: 0.00
Arrest_True: 0.00
```

Fuente: Elaboración propia

Con estos resultados, se observa que las variables “Ward” y “PoliceBeats” son excesivamente importantes para nuestro modelo de clasificación.

Ilustración 42. Representación de la importancia de las variables del Gradient Boosting



Fuente: Elaboración propia

Con estos resultados se obtiene un modelo con una precisión muy alta. Lo más probable es que estos resultados se vean afectados por la alta correlación que tenemos en nuestras variables, y debido a esto, tengamos tanta precisión en nuestro modelo.

Ilustración 43. Resultado final del Gradient Boosting

```
print('Precisión Gradient Boosting train/test  {0:.3f}/{1:.3f}'
      .format(gb.score(Xgb_train, ygb_train), gb.score(Xgb_test, ygb_test)))
```

```
Precisión Gradient Boosting train/test  0.969/0.967
```

Fuente: Elaboración propia

Aquí, se vuelve a mostrar algo parecido a lo que vimos en nuestro modelo de Regresión Lineal, y es que el set de entrenamiento y el set de test nos dan resultados prácticamente idénticos.

No contentos con ello, se realizó una optimización de las variables con el método: ***Backward Elimination*** que implica comenzar con todas las posibles variables y comprobar el efecto de eliminar una característica utilizando un criterio para comparar el nuevo modelo con el anterior.

De esta forma, eliminas en cada paso una característica con la que se mejore el modelo y repetir este proceso hasta que eliminar una característica no mejore el modelo.

Ilustración 44. Método Backward Elimination

```
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFECV

rfe_feature_selector = RFECV(
    LogisticRegression(solver="liblinear"),
    cv = 5,
    scoring = "neg_log_loss",
    n_jobs=-1,
).fit(Xgb_train, ygb_train)
```

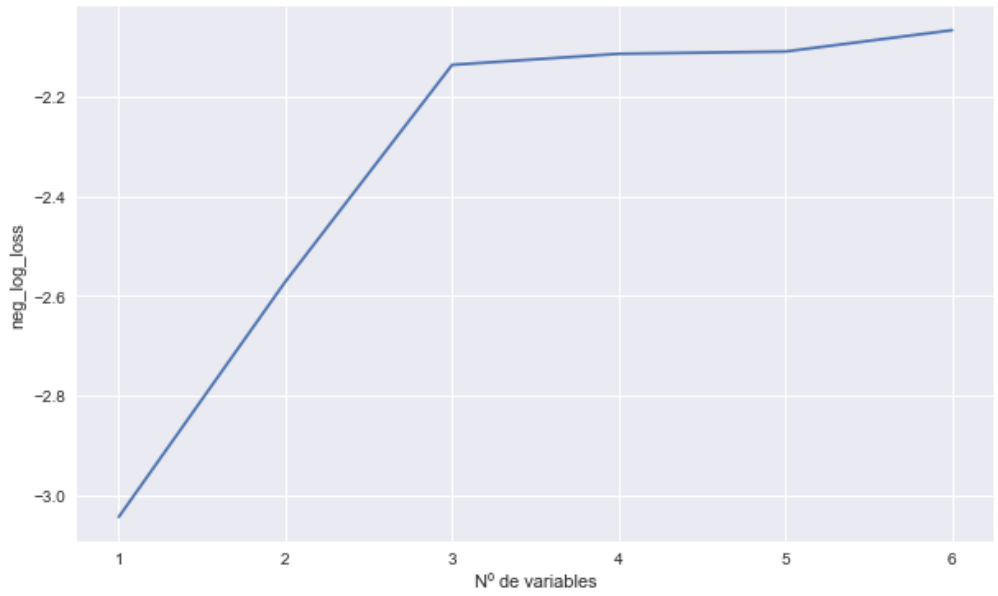
Fuente: Elaboración propia

Los resultados de este modelo se muestran en un gráfico. Indicándole al modelo que muestre las variables que se deberían eliminar para la optimización de nuestro modelo.

Aquí la métrica utilizada ha sido **neg_log_loss**. Este valor, se encarga, de medir el ruido que el modelo Gradient Boosting o cualquier modelo que se esté evaluando. Lo que hace es introducir unos valores a la hora de predecir, los cuales no conoce su etiqueta. Por los tanto, se encarga de medir cual es la desviación que tendría nuestro clasificador para nuevos valores proporcionados que no hayan sido vistos.

Ilustración 45. Representación del Backward elimination

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.set_ylabel("neg_log_loss")
ax.set_xlabel("Nº de variables")
ax.plot(
    [i + 1 for i in range(Xgb_train.shape[1])],
    rfe_feature_selector.grid_scores_
)
plt.show()

print("Variables a eliminar:")
for feature in variables_gb[~rfe_feature_selector.support_]:
  print(f"\t{feature}")
```




```
Variables a eliminar:
```

Fuente: Elaboración propia

Como se ve en la ilustración 45, para las seis variables, se obtiene un resultado de la métrica más próxima a cero y de esta forma, nos indica, que no debemos eliminar ninguna variable.

El otro modelo que se ha implementado para la optimización de las variables, ha sido el GridSearch Validator. Este modelo usa la validación cruzada.

La validación cruzada es una técnica para detectar la existencia de sobreajuste en los modelos. Su utilización en la selección de los modelos garantiza que los resultados sean más estables.

En la validación cruzada, el conjunto de datos es dividido en una cantidad de grupos y se utilizan todos menos uno para la construcción de un modelo y los restantes para su validación. Se repite el proceso tantas veces como grupos se hubiesen definido.

Al repetir el proceso con diferentes datos se puede obtener un promedio de los modelos construidos e identificar si estos son estables o no, debido a la existencia de sobreajuste.

Ilustración 46. Método de Validación Cruzada

```
from sklearn.model_selection import GridSearchCV

np.random.seed(0)

param_grid = {
    "n_estimators": [10, 20, 30],
    "min_samples_split": [2, 30, 60],
    "subsample": [1.0, 0.8, 0.6]
}

grid_search = GridSearchCV(
    GradientBoostingClassifier(),
    param_grid = param_grid,
    cv = 5,
    verbose = 1,
    return_train_score = True,
    scoring = "neg_log_loss",
    n_jobs=-1
).fit(Xgb_train, ygb_train)
```

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits
```

Fuente: Elaboración propia

El resultado de este modelo, se ha conseguido mediante una función, la cual extrae los datos sobre la base de los estimadores y de los tamaños de la muestra. El resultado - que se ha obtenido del modelo, se muestra en la ilustración 47.

Ilustración 47. Resultados de la validación cruzada

```
print(f"Mejores parámetros: {grid_search.best_params_}")
print(f"Mejor resultado: {grid_search.best_score_:.3f}")
```

```
Mejores parámetros: {'min_samples_split': 30, 'n_estimators': 30, 'subsample': 0.8}
Mejor resultado: -0.235
```

Por lo tanto, se obtiene para una muestra de 30 estimadores, y de un tamaño mínimo de la muestra de 30, se tiene que ***el mejor*** resultado es ***-0.235***. Es un resultado óptimo

para nuestro modelo, porque cuanto más nos acerquemos a cero, mejor es nuestro modelo y mayor es nuestra precisión. Este resultado se extra de la siguiente ilustración.

Ilustración 48. Resultados del modelo GridSearch.

			train_score		test_score	
			mean	var	mean	var
min_samples_split	n_estimators	subsample				
2	10	0.6	-0.772360	2.129152e-05	-0.774950	0.000049
		0.8	-0.775900	5.411376e-05	-0.778356	0.000116
		1.0	-0.774650	3.129466e-05	-0.777314	0.000080
	20	0.6	-0.386429	1.523551e-05	-0.388928	0.000053
		0.8	-0.389512	1.399792e-05	-0.392656	0.000016
		1.0	-0.387846	9.863214e-06	-0.390426	0.000045
	30	0.6	-0.235459	4.927588e-07	-0.237924	0.000032
		0.8	-0.234921	8.609832e-06	-0.237788	0.000019
		1.0	-0.234942	5.915417e-06	-0.237674	0.000007
30	10	0.6	-0.776555	5.591724e-05	-0.779671	0.000316
		0.8	-0.773907	1.664504e-05	-0.775598	0.000090
		1.0	-0.774653	3.124873e-05	-0.777316	0.000080
	20	0.6	-0.386278	1.073070e-05	-0.389521	0.000037
		0.8	-0.387920	1.314582e-05	-0.390717	0.000042
		1.0	-0.387867	9.872712e-06	-0.390461	0.000045
	30	0.6	-0.233449	7.024816e-06	-0.236491	0.000017
		0.8	-0.232745	9.748165e-06	-0.235458	0.000005
		1.0	-0.234665	3.311134e-06	-0.237385	0.000012
60	10	0.6	-0.774349	1.387322e-05	-0.777079	0.000172
		0.8	-0.777528	7.521670e-05	-0.779979	0.000083
		1.0	-0.774653	3.124873e-05	-0.777316	0.000080
	20	0.6	-0.386877	1.643288e-05	-0.389611	0.000124
		0.8	-0.392893	1.152277e-05	-0.395765	0.000090
		1.0	-0.387867	9.874445e-06	-0.390461	0.000045
	30	0.6	-0.234730	2.518064e-06	-0.237941	0.000019
		0.8	-0.233251	3.647664e-06	-0.235821	0.000022
		1.0	-0.234717	3.278500e-06	-0.237439	0.000012

Fuente: Elaboración propia

Tras la evaluación del modelo, y su posterior estudio de optimización, el resultado final del modelo Gradient Boosting, se muestra en la ilustración 49.

Ilustración 49. Resultado final del Gradient Boosting

```
from sklearn.pipeline import Pipeline
from sklearn.metrics import log_loss

pipeline = Pipeline([
  ("rfe", rfe_feature_selector),
  ("estimator", grid_search)
])
```

```
ygb_test_pred = pipeline.predict_proba(Xgb_test)
print(f"Log loss test: {log_loss(ygb_test, ygb_test_pred):.3f}")
```

```
Log loss test: 0.239
```

Fuente: Elaboración propia

El resultado final de ***0.239***, refleja un buen modelo, con una métrica próxima a cero y con una precisión elevada entre los sets de entrenamiento y test. Minimizando este valor quiere decir que maximizamos la precisión del clasificador.

6.3 Resultados del Random Forest

Un problema fundamental de los árboles de decisión es que, si no se tiene mucho cuidado con los hiperparámetros tienden a sobreajustar el dataset. Este modelo tiende a solucionar este problema. Es un conjunto de árboles de decisión que trabajan de manera paralela. Este modelo, al igual que el Gradient Boosting, es un modelo de clasificación, que a mayor número de estimadores, mejores resultados tenemos. Los resultados que hemos obtenido con este modelo, se pueden observar en la ilustración 50.

Ilustración 50. Resultados del Random Forest

```
for name, importance in zip(variables_rf, rf.feature_importances_):
  print(f"{name}: {importance:.2f}")
```

```
Ward: 0.35
Census Tracts: 0.22
Police Beats: 0.40
Primary Type: 0.01
Location Description: 0.01
Arrest_True: 0.00
```

Fuente: Elaboración propia

En este caso, se muestra como las variables categóricas pasan a tener un poco de relevancia y las variables numéricas, se distribuye forma más homogénea la relevancia. La gran diferencia de este modelo con el anterior, es la precisión que obtenemos en el set de entrenamiento con respecto al set de test:

Ilustración 51. Precisión del Random Forest

```
print('Precisión Random Forest train/test  {0:.3f}/{1:.3f}'
      .format(rf.score(Xrf_train, yrf_train), rf.score(Xrf_test, yrf_test)))
```

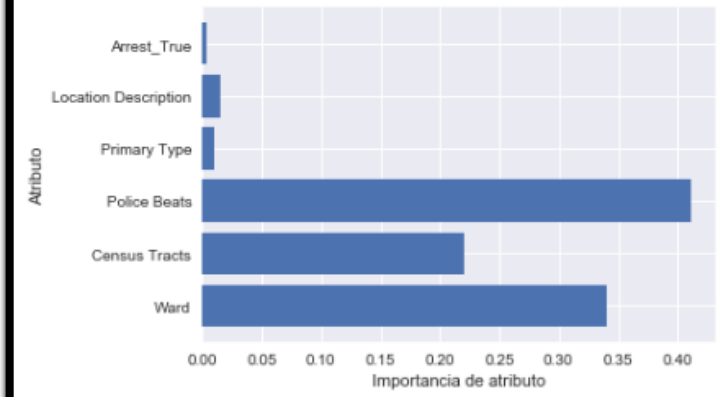
```
Precisión Random Forest train/test  0.994/0.983
```

Fuente: Elaboración propia

La precisión en este caso, nos indica que el modelo es prácticamente perfecto en el set de entrenamiento. Si en este caso mostramos el grafico donde vemos la importancia de las variables de este modelo, veremos diferencias con respecto al del Random Forest.

Ilustración 522. Importancia de las Variables del Random Forest

```
# Mostramos la importancia de cada atributo
n_atributos = 6
plt.barh(range(n_atributos), rf.feature_importances_, align='center')
plt.yticks(np.arange(n_atributos), variables_rf)
plt.xlabel("Importancia de atributo")
plt.ylabel("Atributo")
plt.show();
```



Fuente: Elaboración propia

Se ha aplicado un Backward Elimination, como en modelos anteriores. De esta forma, se ven que variables no aportan información al modelo, y deben ser eliminadas. Pero en este caso, nos muestra que no se elimine ningún variable

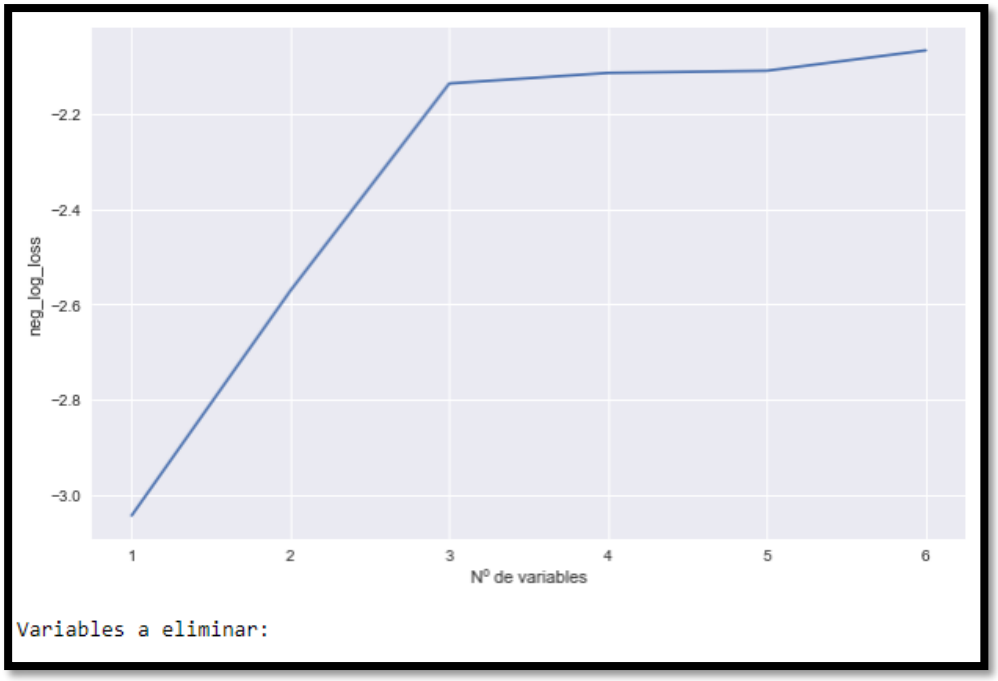
Ilustración 53. MétodoBackward Elimination para Ramdom Forest

```
rfe_feature_selector = RFECV(
    LogisticRegression(solver="liblinear"),
    cv = 5,
    scoring = "neg_log_loss",
    n_jobs=-1,
).fit(Xrf_train, yrf_train)
```

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.set_ylabel("neg_log_loss")
ax.set_xlabel("Nº de variables")
ax.plot(
    [i + 1 for i in range(X_train.shape[1])],
    rfe_feature_selector.grid_scores_
)
plt.show()

print("Variables a eliminar:")
for feature in variables[~rfe_feature_selector.support_]:
  print(f"\t{feature}")
```

Fuente: Elaboración propia

Ilustración 54. Visualización Backward Elimination para Ramdom Forest



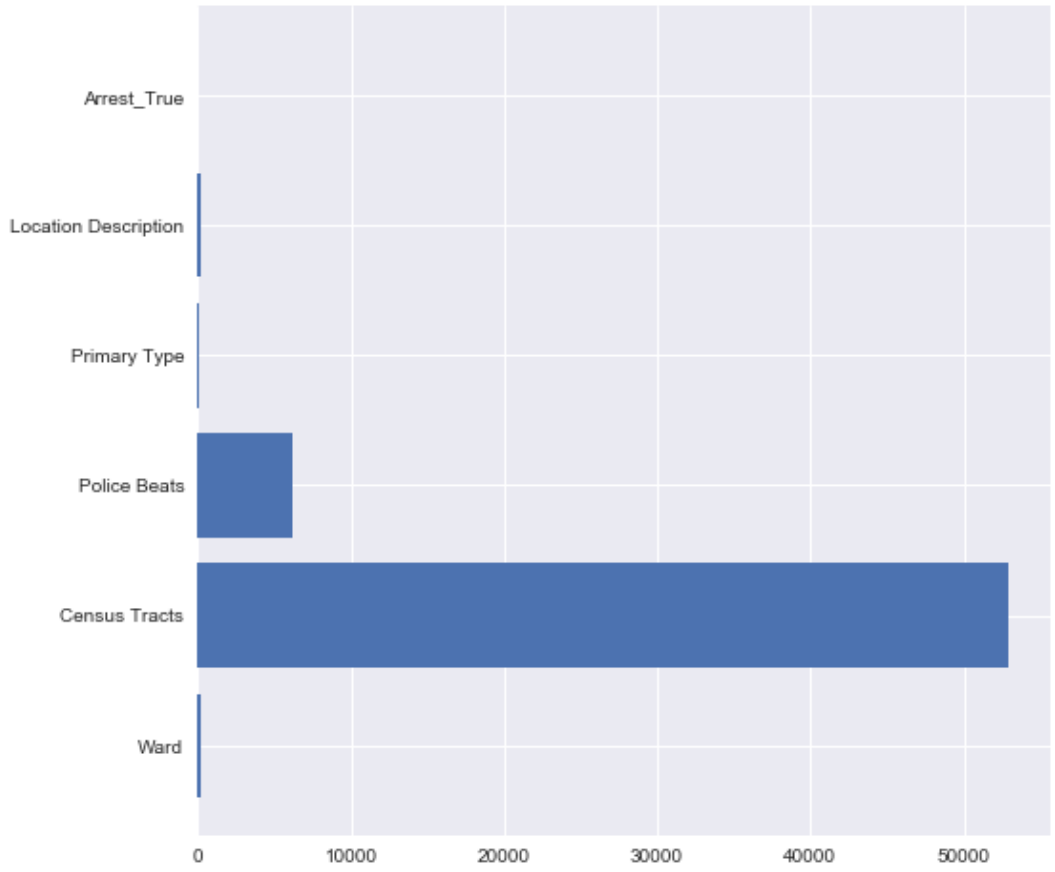
Fuente: Elaboración propia

En este modelo, se ha querido hacer un análisis de los atributos, como si de un modelo no supervisado se tratase. Es decir, se ha utilizado el transformador **Variance Threshold** para ver la importancia de las variables que tenemos en el modelo, sin importar la relación que tienen entre ellas.

En general, se eliminan las variables con ***varianza = 0***, es decir, variables constantes, ya que no aportan información al problema.

Ilustración 55. Representación del método Variance Threshold

```
from sklearn.feature_selection import VarianceThreshold

vt_feature_selector = VarianceThreshold(threshold=0.0)
vt_feature_selector.fit(Xrf_train)
fig, ax = plt.subplots(figsize = (8, 8))
ax.barh(variables_rf, vt_feature_selector.variances_)
plt.show()
```



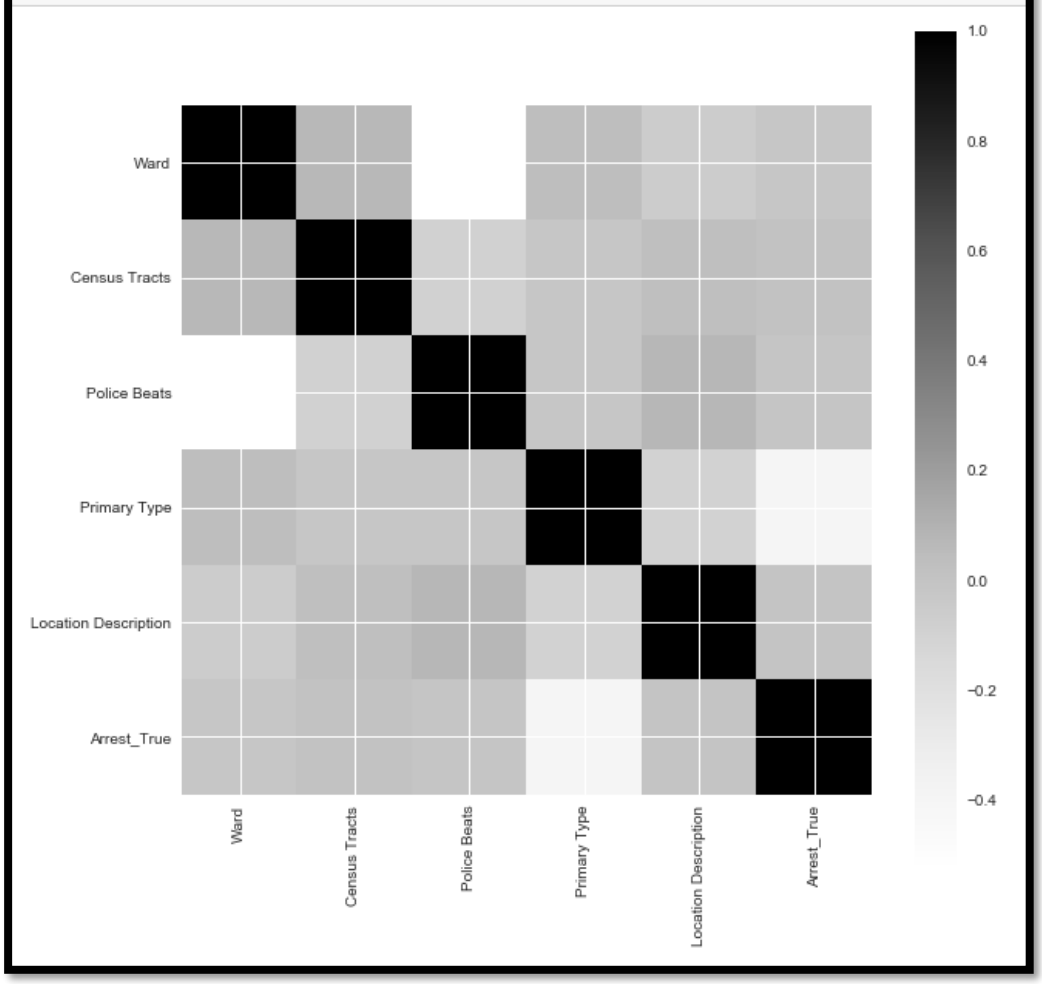
Fuente: Elaboración propia

La varianza de la variable Arrest no debe ser igual a cero, pero si debe estar muy próxima, como se observa en el gráfico de la representación de las varianzas. Además, las variables que están muy correlacionadas entre sí pueden dañar la efectividad de los modelos, hacer nuestro modelo menos eficiente computacionalmente y dañar la interpretabilidad de los mismos.

Para determinar la correlación entre variables hemos empleado un método que calcula la correlación uno a uno de cada variable, generando una llamada **matriz de correlación**.

Ilustración 56. Matriz de correlación

```
fig, ax = plt.subplots(figsize = (10, 10))
cr_matrix = pd.DataFrame(columns = variables, data = Xrf_train).corr()
im = ax.imshow(cr_matrix)
ax.figure.colorbar(im)
plt.xticks(range(len(variables_rf)), variables_rf, fontsize=10, rotation=90)
plt.yticks(range(len(variables_rf)), variables_rf, fontsize=10, rotation=0)
plt.show()
```



Fuente: Elaboración propia

Se puede apreciar que existe cierta correlación entre algunas variables, pero el modelo no nos pide eliminar ninguna de ellas a la hora de mostrar cuales se deben eliminar:

Ilustración 57. Variables a eliminar del Método Variance Threshold

```
cl_feature_selector = ColinearityFeatureSelector(threshold = 0.9)
cl_feature_selector.fit(Xrf_train)
print("Variables a eliminar:")
for feature in variables_rf[~cl_feature_selector.support_]:
  print(f"\t{feature}")
```

```
Variables a eliminar:
```

Fuente: Elaboración propia

Tras haber hecho un estudio de nuestras variables sin fijarnos en nuestra variable objetivo, continuamos con la optimización de los hiperparámetros aplicando el método **Grid Search Validator**, tal y como hemos hecho con el modelo Gradient Boosting y así de esta forma, podemos ver las diferencias entre los modelos y sus resultados.

Ilustración 587. MétodoGrid Search Validator

```
from sklearn.model_selection import GridSearchCV

np.random.seed(0)

param_grid = {
    "n_estimators": [1, 50, 100],
    "min_samples_split": [2, 30, 60],
    "subsample": [1.0, 0.8, 0.6]
}

grid_search_rf = GridSearchCV(
    GradientBoostingClassifier(),
    param_grid = param_grid,
    cv = 5,
    verbose = 1,
    return_train_score = True,
    scoring = "neg_log_loss",
    n_jobs=-1
).fit(Xrf_train, yrf_train)
```

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed: 19.5min
[Parallel(n_jobs=-1)]: Done 135 out of 135 | elapsed: 81.4min finished
```

Fuente: Elaboración propia

Ilustración 598. Resultados Grid Search Validator,

			train_score		test_score	
			mean	var	mean	var
min_samples_split	n_estimators	subsample				
2	1	0.6	-2.076816	1.660484e-05	-2.077233	0.000012
		0.8	-2.073783	1.375726e-05	-2.075137	0.000046
		1.0	-2.076489	3.592023e-06	-2.077917	0.000047
	50	0.6	-0.122954	3.296561e-06	-0.127552	0.000029
		0.8	-0.123111	2.290099e-06	-0.126862	0.000008
		1.0	-0.123747	2.966999e-06	-0.128159	0.000007
	100	0.6	-0.106263	2.766821e-03	-0.116058	0.003129
		0.8	-0.068153	3.293751e-04	-0.079862	0.000591
		1.0	-0.076356	4.355390e-04	-0.084876	0.000360
30	1	0.6	-2.076939	7.483359e-06	-2.078902	0.000035
		0.8	-2.076072	1.597086e-05	-2.076213	0.000028
		1.0	-2.076493	3.578650e-06	-2.077922	0.000047
	50	0.6	-0.123559	2.148885e-06	-0.127973	0.000009
		0.8	-0.122979	3.619049e-07	-0.127576	0.000010
		1.0	-0.123842	1.382580e-06	-0.127860	0.000010
	100	0.6	-0.111500	7.825002e-03	-0.114872	0.005397
		0.8	-0.082953	5.150474e-04	-0.091204	0.000463
		1.0	-0.071072	4.485656e-04	-0.079536	0.000180
60	1	0.6	-2.076354	3.874062e-05	-2.077885	0.000020
		0.8	-2.075898	2.466080e-06	-2.077367	0.000048
		1.0	-2.076493	3.578650e-06	-2.077922	0.000047
	50	0.6	-0.122631	2.196633e-06	-0.126528	0.000007
		0.8	-0.123709	6.584940e-07	-0.127156	0.000018
		1.0	-0.123851	9.618161e-07	-0.127768	0.000010
	100	0.6	-0.067735	9.393371e-05	-0.075906	0.000054
		0.8	-0.074753	4.358500e-04	-0.081469	0.000281
		1.0	-0.084792	5.160493e-04	-0.090401	0.000323

```
print(f"Mejores parámetros: {grid_search_rf.best_params_}")
print(f"Mejor resultado: {grid_search_rf.best_score_:.3f}")
```

```
Mejores parámetros: {'min_samples_split': 60, 'n_estimators': 100, 'subsample': 0.6}
Mejor resultado: -0.076
```

Fuente: Elaboración propia

El resultado obtenido es muy bueno, se tiene que para un número de 100 estimadores y 60 muestras, con una submuestra de 0.6 se tiene un resultado de -0.076. Como se ha comentado antes, el ***Grid Search Validator*** cuanto más se acerque el resultado a cero, mejor es el modelo. De esta forma, el resultado obtenido es buenísimo, dado que prácticamente es cero. La evaluación final del modelo Random Forest ha sido la siguiente:

Ilustración 609. Resultado final del modelo Random Forest

```
from sklearn.pipeline import Pipeline
from sklearn.metrics import log_loss

pipeline = Pipeline([
    ("variance", vt_feature_selector),
  ("colinear", cl_feature_selector),
  ("rfe", rfe_feature_selector),
  ("estimator", grid_search)
])
```

```
ygb_test_pred = pipeline.predict_proba(Xrf_test)
print(f"Log loss test: {log_loss(yrf_test, yrf_test_pred):.3f}")
```

```
Log loss test: 0.258
```

Fuente: Elaboración propia

Obtenemos un modelo una incertidumbre próxima a cero con un ***Log Loss = 0.258***, una precisión altísima y con una evaluación de los hiperparámetros excelente, quedando como el mejor modelo de todos los que se han utilizado para la predicción.

IV. CONCLUSIONES

9. Conclusiones generales

En el presente trabajo de fin de máster, nos ha abierto los ojos sobre la importancia y la complejidad de las técnicas Big Data en el ámbito de la predicción. Debido, a la dificultad que se ha experimentado para lograr un algoritmo optimo con las variables tan poco relevantes con las que se contaba. El modelo seleccionado, será el ***Random Forest***, debido a los buenos resultados que ha mostrado a lo largo de la predicción. De este modelo, se pueden concluir una serie de aspectos relevantes:

- Para la muestra de datos elegida, siempre es posible encontrar una combinación de parámetros de funcionamiento de Random Forest que hagan que este algoritmo muestre resultados precisos que otros modelos de árboles de decisión.
- Los árboles de decisión del modelo tienen una ventaja importante en cuanto a la varianza de sus predicciones.
- Las predicciones de los árboles de inferencia han sido muy sesgadas, mostrándonos valores de la función de respuesta cercanos a cero.
- El número de árboles que contiene el modelo Random Forest no influye de manera significativa en el sesgo de las predicciones, pero si es muy importante para disminuir la varianza de las mismas.
- Por otro lado, la magnitud de influencia de la profundidad de los árboles en la varianza es muy baja, y solo se da en bosques con una profundidad de un centenar de estimadores.

Por lo tanto, este modelo se encarga de mejorar los resultados que nos han dado, no solo en cuanto a precisión se refiere, sino que en la evaluación de los hiperparámetros y en la interpretación de las variables, es el que mejor se ajusta.

Finalmente, este algoritmo será el que se implementara principalmente, con la intención de pasarle una serie de datos históricos, y en función, de dicha información se generara una predicción que indique como va a evolucionar el crimen en los distintos distritos que conforman la ciudad de chicago. Con la intención de reducir el crimen en las zonas más afectadas por estos delitos, produciendo un seguridad y tranquilidad en las calles de chicago gracias al análisis predictivo de los datos y a las nuevas tecnologías.

10. Futuras líneas de investigación

Todo trabajo de investigación supone algún avance en la materia a estudiar y a su vez abre nuevas vías de investigación futuras. Dudas sobre cómo seguir mejorando en dicho campo y como se puede funcionar como complemento a muchos otros. A continuación, se muestran algunos de los diferentes caminos a explorar en este tema y que contienen un alto valor para seguir mejorando como sociedad.

En primer lugar, se considera que sería de vital importancia la ***unificación de un modelo*** general posteriormente configurable en función del área y localización común a todos los países. En nuestro caso, nos hemos centrado en una ciudad de un país, sin embargo, en los últimos años se han podido ver como la coordinación entre las fuerzas y cuerpos de seguridad de los distintos estados es fundamental si se quiere contener y frenar los crímenes a nivel internacional como puede ser el terrorismo.

Otra de las aplicaciones que se tienen en cuenta son aquellas relacionadas con la calidad. Es decir, conseguir un acceso a los datos de forma más actualizada a tiempo real. Nuestros datos han sido obtenidos a través de una API perteneciente a la policía de la ciudad de Chicago la cual es actualizada diariamente, pero tiene un retraso de 7 días.

Sería muy interesante conseguir un flujo de datos constante que fuese actualizando el modelo de forma continuada cada pocos segundos. Algo similar a una ingesta de datos a ***tiempo real o streaming*** en los que según fuesen entrando los datos estos se procesaran rápidamente y se llevaran a las bases de datos policiales actualizándose de forma continua. Con esto, se lograría que los activos policiales estuviesen en cada momento donde más se les va a necesitar y así obviamente también lograr un modelo mucho más eficaz.

Al final, se está consiguiendo una predicción para gestionar más activos policiales a zonas denominadas calientes, sin embargo, con esto conseguiríamos casi adelantarnos al delito que se va a cometer y lograríamos reducirlos en gran cantidad.

Nuestra investigación, se centra en la predicción de crímenes con el objetivo principal de conseguir reducirlos. En todo momento, se ha relacionado ese descenso con una mejora de la gestión de los activos policiales en diferentes áreas, pero, sin embargo, se ha

reducido el ámbito de aplicación, es decir, estos modelos de reducción del crimen pueden también ayudar a gestionar hospitales, centros sociales, centros educativos, etc.

Sería fantástico ***lograr implementar un modelo*** con todos los ***agentes sociales relacionados*** con el fin de alcanzar una sociedad que pueda proveer los recursos de la manera más óptima. Un modelo capaz de que cuando un crimen se cometa permita actuar a la policía con la mayor rapidez, pero a su vez el hospital tenga constancia para que esté preparado el más cercano, si fuese necesario o incluso que se evalúan las áreas más calientes en cuento a crimen y se lleven a cabo actividades deportivas para relajar el ambiente.

V. REFERENCIAS BIBLIOGRÁFICAS

- https://magnet.xataka.com/en-diez-minutos/chicago-tiene-casi-la-misma-poblacion-que-madrid-y-40-veces-mas-asesinatos, consultado el 18/03/2020.
- https://www.tuataratech.com/2016/08/como-combatir-el-crimen-con-big-data.html, consultado el 22/03/2020.
- https://www.kaggle.com/chicago/chicago-crime/kernels consultado el 23/03/2020.
- https://data.cityofchicago.org/Health-Human-Services/Census-Data-Selected-socioeconomic-indicators-in-C/kn9c-c2s2, consultado el 28/03/2020.
- https://data.cityofchicago.org/Public-Safety/Crimes-Map/dfnk-7re6 consultado el 29/03/2020.
- https://blog.realinstitutoelcano.org/prevencion-del-crimen-y-prediccion-de-delitos-en-que-punto-esta-espana/, consultado el 18/04/2020.
- https://www.lavanguardia.com/tecnologia/20181202/453268636098/policia-britanica-uso-inteligencia-artificial-delitos-crimenes-delincuencia.html, consultado el 20/04/2020.
- https://elpais.com/tecnologia/2017/03/09/actualidad/1489078250_691655.html, consultado el 24/04/2020.
- https://blog.desdelinux.net/ciencia-de-datos-con-python/, consultado el 25/04/2020.
- https://www.ionos.com/digitalguide/websites/web-development/jupyter-notebook/, consultado el 18/04/2020.
- https://www.makesoft.es/powr-bi-que-es-power-bi/, consultado el 08/05/2020.
- https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present-Map/c4ep-ee5m/, consultado el 10/05/2020.

- https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

- https://www.the-modeling-agency.com/crisp-dm.pdf, consultado el 01/06/2020.

- https://medium.com/datos-y-ciencia/gesti%C3%B3n-de-flujos-de-trabajo-de-aprendizaje-autom%C3%A1tico-con-pipelines-de-scikit-learn-parte-2-eeecab194d83, consultado el 18/06/2020.

VI. ANEXOS

Anexo 1. Distritos policiales de chicago

Numero Distrito Policial	Distrito Policial
12º	Near West
1º	Central
2º	Wentworth
3º	Grand Crossing
4º	South Chicago
5º	Calumet
6º	Gresham
7º	Englewood
8º	Chicago Lawn
9º	Deering
10º	Ogen
11º	Harrison
14º	Shakespeare
15º	Austin
16º	Jefferson Park
17º	Albany Park
18º	Near North
19º	Town Hall
20º	Foster
22º	Morgan Park
24º	Rogers Park
25º	Grand Central

Anexo 2. Distritos en los que se divide chicago

Numero Distrito	Nombre Distrito
Distrito 1	Palmer Square y West Town
Distrito 2	Near North Side,Old Town, Wicker Park, UkrainianVillage
Distrito 3	Bronzeville, Washington Park, Fuller Park, South Loop
Distrito 4	Hyde Park, Grand Boulevard, Douglas, Bronzeville, PrintersRow, South Loop, Kenwood
Distrito 5	IndianVillage, Hyde Park, Jackson Park, South Shore
Distrito 6	Chatham, Englewood

Distrito 7	Calumet Heights, Pill Hill, South Chicago, South Deering, South Shore
Distrito 8	South Shore, Chatham, Calumet Heights, Pullman, Avalon Park, Burnside, South Chicago
Distrito 9	Chatham, Pullman, Riverdale, Roseland, Washington Heights
Distrito 10	Hegewisch, East Side, South Deering, JefferyManor
Distrito 11	Bridgeport, Canaryville, ArmourSquare, Pilsen, UniversityVillage
Distrito 12	Brighton Park, McKinley Park, Little Village
Distrito 13	West Lawn, Clearing, West Elsdon, Garfield Ridge
Distrito 14	ArcherHeights, Glendale, Garfield Manor, Gage Park
Distrito 15	Brighton Park, Gage Park, Canaryville, West Englewood, Back oftheYards
Distrito 16	Englewood, Gage Park, West Englewood, Chicago Lawn
Distrito 17	Chicago Lawn, Marquette Park, Gresham, Auburn Gresham, West Englewood
Distrito 18	Ashburn, Marquette Park, Auburn Gresham
Distrito 19	Beverly, Mount Greenwood, Morgan Park, Washington Heights
Distrito 20	Back oftheYards, Canaryville, Washington Park, Englewood
Distrito 21	Auburn Gresham, Washington Heights, Gresham, Chatham, Roseland

Distrito 22	Auburn Gresham, Washington Heights, Gresham, Chatham, Roseland
Distrito 23	Far West Side, West Elsdon, West Lawn, Garfield Ridge, Clearing
Distrito 24	North Lawndale, Douglas Park, Little Village
Distrito 25	Lower West Side, Pilsen, Greek Town, China Town, UniversityVillage
Distrito 26	Humboldt Park, Hermosa, UkrainianVillage, Logan Square
Distrito 27	East Garfield Park, Humboldt Park, Near West Side, Greektown, United Center Park, Near North Side
Distrito 28	West Garfield Park, East Garfield Park, Austin, Douglas Park, UniversityVillage
Distrito 29	Elmwood Park, Galewood, Austin
Distrito 30	Belmont Cragin, Portage Park, Irving Park
Distrito 31	Hermosa, Belmont Cragin, Logan Square
Distrito 32	Bucktown, East Village, Goose Island, Hamlin Park, Lakeview, Lincoln Park, Pulaski, RoscoeVillage
Distrito 33	Albany Park, Avondale, Irving Park, RavenswoodManor, North Park, North Center
Distrito 34	Far South Side, West Pullman, Washington Heights, Morgan Park, Roseland
Distrito 35	Hermosa, Logan Square, Avondale, Irving Park, Albany Park

Distrito 36	Montclare, Portage Park, Belmont Cragin, Hermosa
Distrito 37	Austin, Humboldt Park
Distrito 38	Portage Park, Dunning
Distrito 39	North Park, Jefferson Park, Albany Park
Distrito 40	Lincoln Square, Edgewater, West Ridge
Distrito 41	Edison Park, Edgebrook, Norwood Park, O'Hare
Distrito 42	Loop, Streeterville, Near North Side, Greektown
Distrito 43	Lincoln Park, Gold Coast, Old Town
Distrito 44	Lakeview
Distrito 45	Old Irving Park, Portage Park, Jefferson Park, Norwood Park
Distrito 46	Uptown, Buena Park, Lakeview
Distrito 47	Ravenswood, North Center
Distrito 48	Edgewater, Andersonville, Uptown
Distrito 49	Rogers Park
Distrito 50	West Ridge, West Rogers Park