

Predicción del tipo de delito que ocurre en la Ciudad Autónoma de Buenos Aires y búsqueda de similaridad mediante Clustering.

Alcuri Federico, Girola Federico, Pantoni Francisco – Nov 2020

Ingeniería Industrial, Universidad Tecnológica Nacional, Facultad Regional Buenos Aires

Abstract

El objetivo del paper está enfocado en la aplicación de modelos de Machine Learning con el fin de clasificar los diferentes tipos de delitos en la Ciudad de Buenos Aires y además lograr el entendimiento del comportamiento de los datos mediante la búsqueda de similaridad de muestras a través de la aplicación de Clustering.

Introducción

A partir del registro durante cuatro años (2016-2019) de delitos en Capital Federal, se procederá a realizar un análisis de los datos para entender cuáles son las variables que intervienen y cómo se comportan a la hora de registrar un hecho delictivo. El objetivo es poder clasificar los delitos una vez ocurridos en base al "Tipo" del cual se trate, mediante la utilización de diferentes modelos de clasificación (Machine Learning), como ser SVM, KNN, LR. Se buscará clasificar la ocurrencia del delito según sea Hurto, Robo con violencia, Lesiones y Homicidios. Por otro lado, mediante aprendizaje no supervisado, aplicando Clustering se estudiará la similitud entre las muestras y así obtener cuáles son las variables que intervienen a la hora de que éstas sean agrupadas.

Datasets utilizados

Para el estudio partió de seis datasets principales, cuatro de ellos corresponden a los delitos registrados en la Ciudad de Buenos Aires en los años 2016 a 2019 inclusive^[1] y los dos restantes con información sobre los barrios porteños en cuanto a características poblacionales e ingresos por barrio. Estos últimos son de elaboración propia en base a los datos abiertos de la Ciudad de Buenos Aires.

Pre-procesamiento de Datos

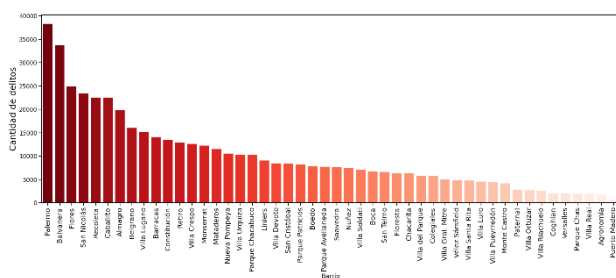
Una vez importados los datos anteriormente mencionados, el primer paso fue el de unificar los correspondientes a los delitos en un único dataset, obteniendo un dataset de 488.541 samples y 10 features. De estas últimas, las más relevantes son las siguientes: fecha, franja horaria, tipo de delito, comuna, barrio, latitud y longitud; mientras que las columnas id y subtipo de delito fueron descartadas para el análisis, además se verificó la existencia de valores nulos, los cuales representaban un 1.74%, por lo que se decidió eliminarlos.

Luego se importó uno de los dataset correspondiente a los barrios de Capital Federal, que contiene como información el nombre de los barrios, precio promedio del metro cuadrado, la comuna y el promedio de ingresos por cada hogar, con el objetivo de incluir mayor cantidad de features al análisis. Para obtener el dataset completo, se unieron los delitos con los barrios mediante la función merge, utilizando como key a la feature "Barrio". Debido a que la columna fecha estaba de la forma año/mes/día, antes de eliminarla se crearon tres nuevas features con el fin de tener por separado cada una de las componentes.

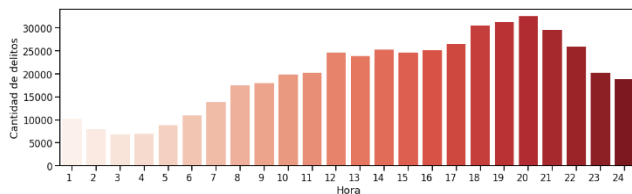
Al finalizar con la limpieza de los Nans, la unión de los dataset y las medidas antes indicadas, el dataset final que se utilizó para los siguientes análisis quedó formado por 480.177 samples y por 12 features, las cuales fueron: Día, Mes, Año, Hora, Tipo, Cantidad, Comuna, Barrio, Latitud, Longitud, Precio_m2 Ingresos. A modo de aclaración, "tipo" hace referencia al tipo de delito, pudiendo ser robo con violencia, hurto sin violencia, homicidio o lesiones. Mientras que la cantidad se refiere a la cantidad de delitos registrados por cada muestra.

Análisis exploratorio de datos EDA

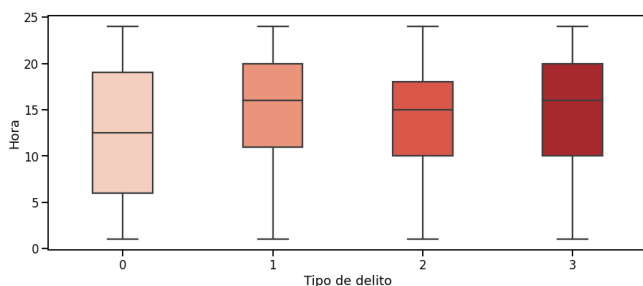
A partir del dataset completo, contemplando los 4 años se graficó un countplot (gráfico de barras) para contabilizar la cantidad de delitos por barrio y así visualizar cuál de estos, en un principio, es el que mayor cantidad de delitos tiene. Los resultados arrojados fueron que el barrio de Palermo era el mayoritario con alrededor de 40.000 delitos, seguido por Balvanera con casi 35.000 y en tercer lugar por Flores con una suma de 25.000 delitos. Los resultados mencionados se pueden visualizar a continuación:



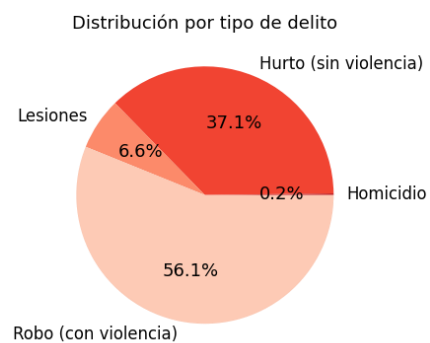
Por otro lado, con el objetivo de ver la hora en la cual ocurrían la mayor cantidad de delitos, se realizó un gráfico de barras, obteniendo de este que la franja horaria es la más delictiva, esta se da entre las 17 y las 22 horas, siendo las 20 la de mayor cantidad de delitos.



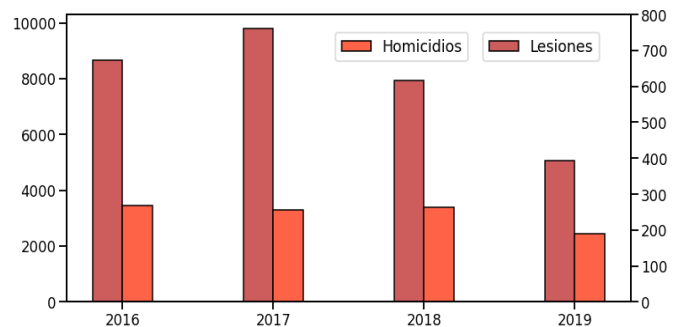
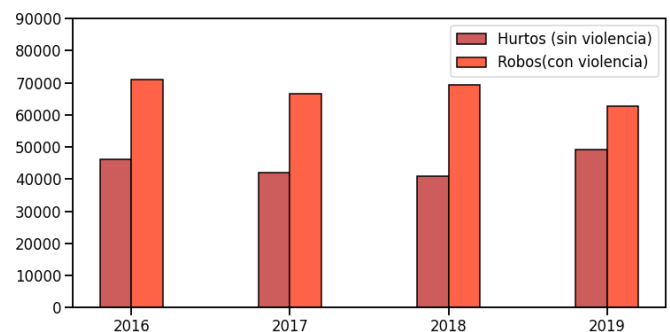
Se realizó un boxplot para describir la distribución de los tipos de delitos a lo largo de las horas del día. Se puede observar cómo los robos, lesiones y hurtos están concentrados entre las 10 de la mañana y las 20 horas, mientras que los homicidios tienen un margen mayor entre aproximadamente las 5 de mañana y las 20 horas.



Mediante la creación de un nuevo dataframe que contiene la cantidad de delitos agrupándolos por cada "tipo" de delito y cada uno de los años, se realizó un gráfico de torta para poder observar la distribución por tipo de delito. En este se puede distinguir que predomina el robo con violencia con un 56,1%, seguido por los hurtos con el 37,1% y en menor medida lesiones con 6,6% y homicidios con 0,2%.

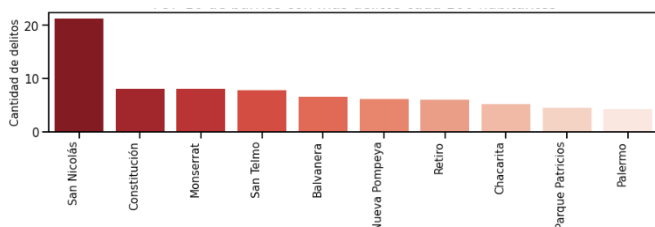


Del dataframe creado anteriormente, se agrupo los hurtos con los robos y las lesiones con los homicidios, con el fin de observar la evolución año a año.

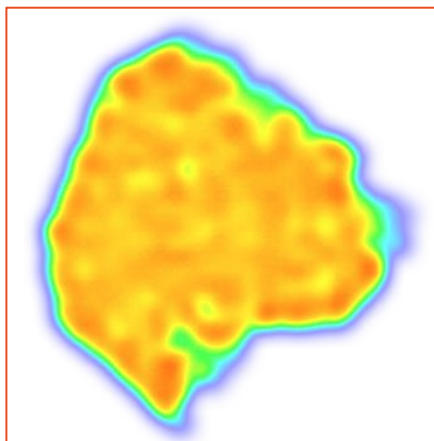


Sin embargo, luego de lo analizado anteriormente se llegó a la conclusión de que la forma en la que se calificó al barrio con mayor peligrosidad era incorrecta.

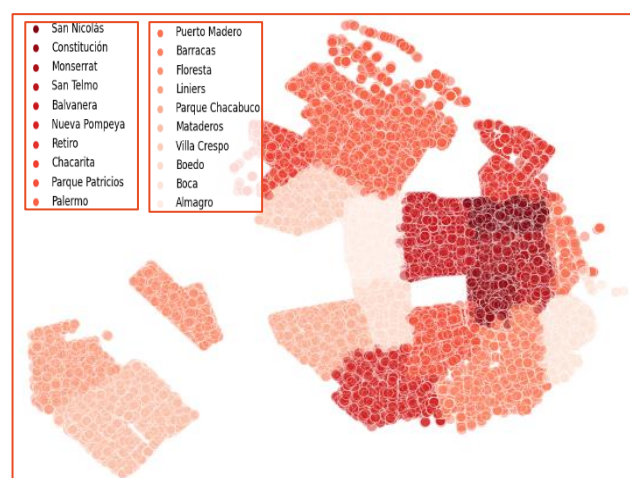
Esto se debe a que no se podría comparar un barrio con otro solo por la cantidad de delitos, sino que necesariamente se debe agregar información relevante sobre cada uno de los barrios, como la superficie y la cantidad de población de cada uno de ellos. Es por esto por lo que se agregó un nuevo dataset que contiene la población, la superficie y la densidad de cada barrio. Con estos nuevos datos se calculó un índice en base a la cantidad de delitos y al número de habitantes (cantidad de delitos/población) y un índice por densidad (cantidad de delitos/densidad). A partir de lo mencionado, se graficó un countplot del top 10 de barrios con más delitos por cada 100 habitantes.



El grafico refleja que los barrios más peligrosos son San Nicolás (zona de Microcentro), Constitución y Monserrat. Es muy importante destacar que Palermo pasó de ser top "1" con el primer análisis solo contemplando la cantidad total de delitos, a ser top "10" luego de este nuevo estudio. Concluyendo con el análisis exploratorio de datos se realizó un mapa de calor interactivo, donde la zona de mayor cantidad de delitos se visualiza con un color rojizo, mientras que las de menor peligrosidad están color verde. El mapa nos permite acercarnos para ver más en detalle una zona específica.



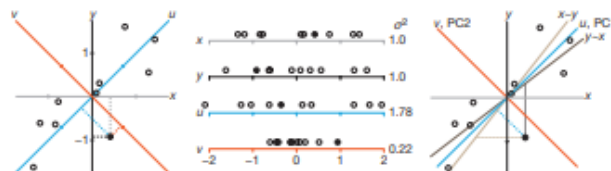
Por último, para llevar a cabo un análisis más detallado y focalizando en aquellos barrios más peligrosos de la Ciudad de Buenos Aires en el año 2019, se realizó un top 20 de los barrios con mayor número de acciones delictivas teniendo en consideración el índice de delictividad antes mencionado. Aprovechando los datos de latitud y longitud de cada delito se visualizó con un scatterplot donde ocurrieron geográficamente. Las referencias corresponden a los barrios ordenados de mayor a menor índice de delictividad.



Reducción de la dimensionalidad PCA & K-Means

Introducción^[2]

PCA es un método utilizado para reducir la dimensionalidad creando nuevas features llamadas componentes principales mediante la obtención de autovalores y autovectores, y seleccionar para los análisis las Features que mejor explican la variabilidad de los datos.



Cluster Analysis consiste en agrupar o segmentar un conjunto de muestras en subconjuntos, grupos o clusters. Los cluster se construyen de manera tal que las muestras pertenecientes a un mismo cluster deben ser más similares entre sí y que

simultáneamente haya baja similaridad entre muestras de distintos clusters. En el algoritmo de K-Means cada cluster se identifica por un centroide y la muestra es asignada al centroide más cercano ^[3]

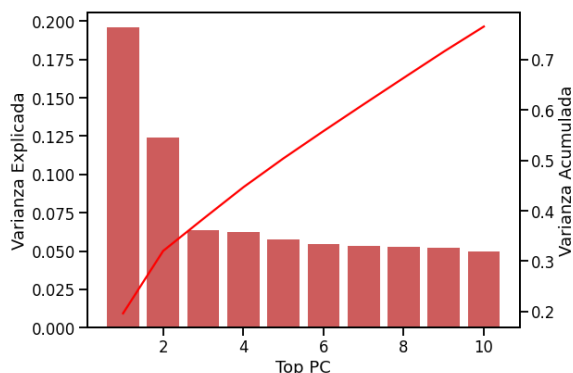
Procesamiento de datos

A partir del dataset completo realizamos la clusterización, para esto, la primera medida fue utilizar Label Encoder para transformar las variables categóricas correspondientes al tipo de delito en numéricas. De esta forma, se le asignó al tipo de delito valores que van del 0 al 3. Luego se tomó un top 10 de los barrios más delictivos generando dummies para obtener features numéricas y así poder trabajar con los modelos.

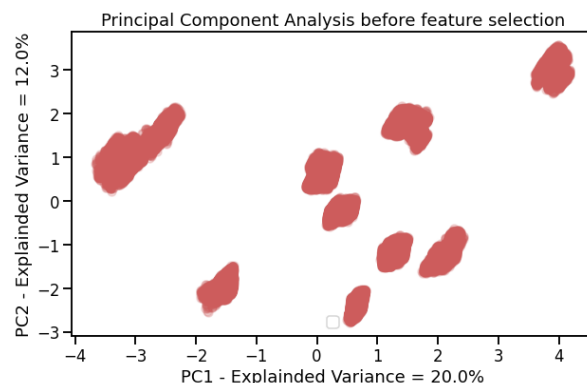
PCA: Principal Components Analysis

Del dataset anterior más las dummies creadas, se filtró por años 2018 y 2019, escaló y generó un Principal Components Analysis (PCA) con las 10 principales componentes, con el objetivo de reducir la dimensionalidad y seleccionar las componentes que expliquen de mejor manera la variabilidad de los datos.

A través del gráfico de barras a continuación se puede visualizar la varianza explicada por cada una de las componentes del dataset en el eje izquierdo, mientras que en el derecho se puede observar la varianza acumulada. Las primeras dos componentes explican el 33,56% de la varianza, mientras que las primeras 10 alcanzan casi el 80% de la variabilidad explicada.

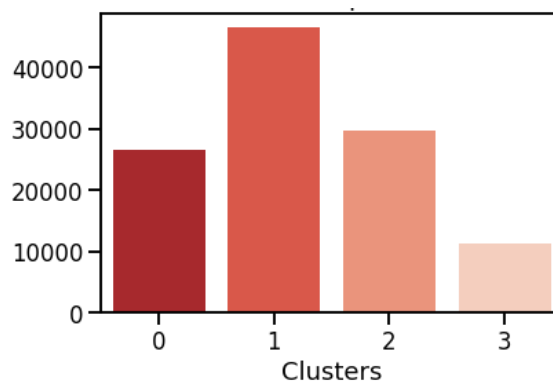


Sin embargo, para poder observarlo y analizarlo se realizó un scatterplot con las dos componentes principales (PC1 y PC2).

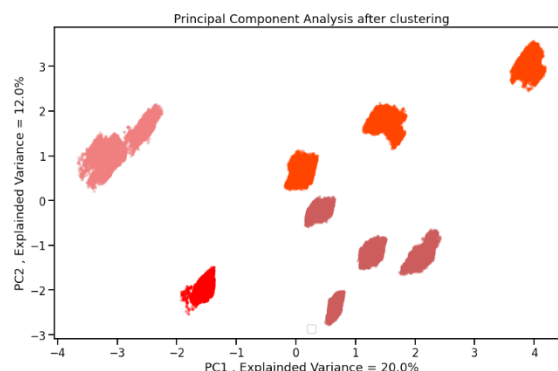


Clustering, K-Means

Mediante el uso del algoritmo K-Means se buscó identificar similaridad entre las muestras, agrupando estas en distintos clústeres según diferentes características en común. Para la utilización del algoritmo se determinó una cantidad máxima de 4 clusters. Este determinó los centroides de cada uno de los clusters y asignó las muestras al clúster más cercano luego de varias iteraciones. A continuación, podemos ver la cantidad de muestras asignada a cada uno de los clusters.



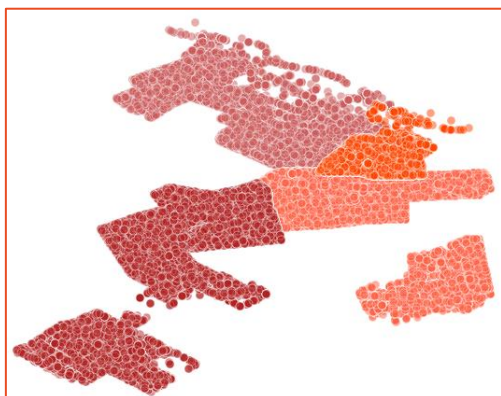
Con las 2 principales componentes halladas se procedió a visualizar mediante un scatterplot cada uno de los clusters con sus respectivas muestras para observar cuál es su comportamiento.



Como podemos observar en el gráfico anterior, se separa en los colores correspondientes a los cuatro clusters formados durante el análisis.

Con el fin de evaluar si la aplicación de clusters fue efectiva, se evaluó su calidad con Silhouette score, el cual arrojó un resultado de 0,34, dicho resultado está cercano a 0 por lo que podemos suponer que los clusters estarían superpuestos y es difícil encontrar grupos definidos.

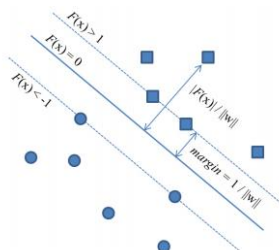
Para no concluir el análisis en los resultados obtenidos se procedió a realizar un EDA reducido para cada uno de los clusters y observar cuáles fueron las variables que determinaron la creación de cada cluster, es decir, poder entender cuál fue la similitud entre las muestras. Para ellos se realizó un nuevo gráfico en el cual se puede observar que las muestras fueron agrupadas por cercanía, es decir por el número de comuna y el barrio.



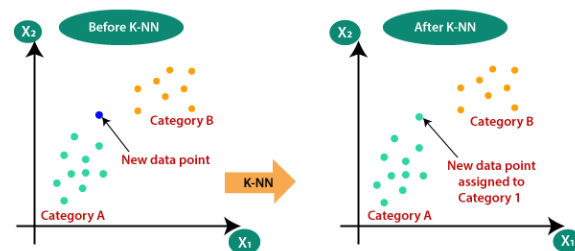
Clasificación KNN, SVM & LR

Introducción

SVM es un clasificador lineal que busca un hiperplano separador que maximice el margen entre las clases. Las muestras que maximicen ese margen son conocidas como Support Vectors. [4]



KNeighbors Classifier [5], es un clasificador que determina a qué grupo pertenece un dato según la cantidad K de vecinos más cercanos de un grupo, según la distancia.



Procesamiento de datos

Para llevar a cabo la clasificación, se seleccionó los 5 barrios más delictivos para el año 2019, se generó dummies con el objetivo de obtener nuevas feature numéricas. Luego el dataset fue dividido en "x" e "y". "x" compuesto por todas las features, excepto las etiquetas (tipo de delito), mientras que "y" contiene el tipo de delito. Posteriormente, utilizando train-test-split de la librería sklearn, se separó en "train" y "test", tomando un 20% para entrenamiento y el 80% restante para evaluar. Con el StandardScaler se escaló los datos de "xtrain" y de "xtest" para luego entrenar a los modelos. Como la variable "y" puede tomar 4 diferentes valores se realizaron clasificaciones multiclases.

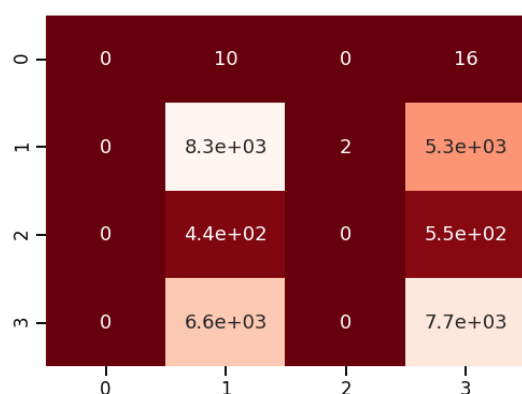
KNeighbors classifier

Se utilizó GridSearch y CrossValidation para seleccionar el conjunto de parámetros que mejor se adapta a los datos y menor error estadístico tiene. Para ello se determinó el número K como un arrange de (1, 20, 1). Se llevó a cabo la predicción utilizando los mejores parámetros determinados por el GS y se comparó dicha predicción con las etiquetas reales, la cual arrojó un Accuracy de 55,37%, además graficamos una matriz de confusión:

0	0	10	0	16
1	0	8.3e+03	2	5.3e+03
2	0	4.4e+02	0	5.5e+02
3	0	6.6e+03	0	7.7e+03
	0	1	2	3

Support Vector Machines SVM

Al igual que modelo de clasificación mencionado anteriormente, se utilizó GridSearch y CrossValidation. Los parámetros seleccionados para realizar el GS fueron: $C = 0.1, 1$ y 10 ; $\text{Gamma} = 1, 0.1$ y 0.01 ; Kernel = 'rbf' y 'linear'. Una vez obtenidos la mejor combinación de hiperparámetros se llevó a cabo la predicción y la comparación de esta con las etiquetas reales, obteniendo un Accuracy de 57,36%.

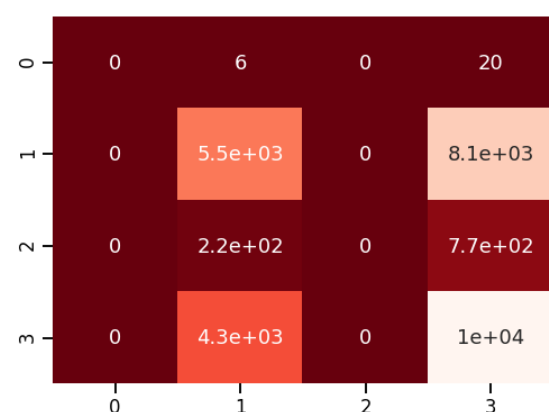


A confusion matrix for the SVM model. The matrix is a 4x4 grid with rows and columns indexed 0 to 3. The diagonal elements represent correct classifications, while off-diagonal elements represent misclassifications. The values are: Row 0: [0, 10, 0, 16]; Row 1: [0, 8.3e+03, 2, 5.3e+03]; Row 2: [0, 4.4e+02, 0, 5.5e+02]; Row 3: [0, 6.6e+03, 0, 7.7e+03].

0	0	10	0	16
1	0	8.3e+03	2	5.3e+03
2	0	4.4e+02	0	5.5e+02
3	0	6.6e+03	0	7.7e+03
	0	1	2	3

Logistic Regression

Por último, se llevó a cabo la clasificación utilizando Logistic Regression, para esto, fue realizado un GS y CV utilizando como único hiperparámetro el costo C , el cual podía tomar los siguientes valores $[0.1, 1, 100]$. Una vez obtenido el mejor C (0.1), se realizó la predicción, comparando con las etiquetas reales (ytest), en este caso, el Accuracy fue de 53,74%.



A confusion matrix for the Logistic Regression model. The matrix is a 4x4 grid with rows and columns indexed 0 to 3. The diagonal elements represent correct classifications, while off-diagonal elements represent misclassifications. The values are: Row 0: [0, 6, 0, 20]; Row 1: [0, 5.5e+03, 0, 8.1e+03]; Row 2: [0, 2.2e+02, 0, 7.7e+02]; Row 3: [0, 4.3e+03, 0, 1e+04].

0	0	6	0	20
1	0	5.5e+03	0	8.1e+03
2	0	2.2e+02	0	7.7e+02
3	0	4.3e+03	0	1e+04
	0	1	2	3

Conclusiones

En cuanto a los modelos de clasificación podemos observar que todos tuvieron un resultado aceptable, con valores entre 53% y 57%.

De esta manera se podrían utilizar alguno de ellos para la clasificación de los delitos ocurridos en la Ciudad Autónoma de Buenos Aires. Sin embargo, creemos que es necesario encontrar nuevas features que se relacionen de mejor manera con los datos obtenidos para poder mejorar la predicción de los modelos, ya que luego de varios análisis se pudo observar que no hay una relación estrecha entre cada una de las features más allá de las que corresponden a las características poblacionales, es decir que, por lo que se pudo observar mediante los datos utilizados, no existe una o más variables que puedan determinar de qué tipo de delito se está tratando.

Al igual que en el caso de clasificación, analizando los resultados obtenidos en el Clustering, se pudo observar que no existe similitud entre las muestras más allá de las características poblacionales de cada uno de los barrios, es por eso que las muestras fueron agrupadas por cercanía y existe una superposición entre ellas.

Referencias

- [1] <https://data.buenosaires.gob.ar/dataset/delitos>
- [2] Lever, J., Krzywinski, M., & Altman, N. (2017). Points of significance: Principal component analysis.
- [3] Arora, P., & Varshney, S. (2016). Analysis of k-means and k-medoids algorithm for big data. *Procedia Computer Science*, 78, 507-512.
- [4] Yu, H., & Kim, S. (2012). SVM Tutorial- Classification, Regression and Ranking. *Handbook of Natural computing*, 1, 479-506.
- [5] <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
- [6] Catedra Ciencia de Datos 2do cuatrimestre 2020 - UTN Frba – ingeniería Industrial.