Maestría en ciencia de los datos y analítica

Hurto de Motos en la ciudad de Medellín

Arcia Jesús Alberto jaarciah@eafit.edu.co
Fajardo Becerra Daian dpfajardob@eafit.edu.co
Salazar Carlos Enrique csalazar@eafit.edu.co
Sepulveda Jimenez Hernan hsepulvedj@eafit.edu.co

Las motocicletas son uno de los medios más efectivos para movilizarse en la ciudad de Medellín, puesto que ahorran tiempo y economía. Sin embargo, también se han vuelto un gran atractivo para las bandas delincuenciales que tienen diferentes zonas, horarios y modos de hurtar este vehículo.

Por esto, estrategias de seguridad pueden ser desarrolladas, teniendo en cuenta el histórico de los diferentes reportes realizados por los ciudadanos y así prevenir los posibles futuros hurtos.

Por lo que se propone realizar un análisis de los hurtos de motos que se han presentado en Medellín desde el 2003 al 2019 en el cual se busca identificar la probabilidad de hurto en un barrio a determinada hora.

Para poder realizar esto, se descargó un Dataset de hurto de motos en medata.gov.co el cual proporciona información relacionada a los hechos relacionados con la seguridad y hurtos de motos en la ciudad de Medellín. Este Dataset consta de 36 variables y 64.869 registros.

Table of Contents

1. Metodología.

Se utiliza la metodología Team Data Science Process (TDSP), es una metodología ágil e iterativa de ciencia de datos para entregar soluciones de análisis predictivo y aplicaciones inteligentes de manera eficiente.

El cual tiene 4 componentes principales:

- ***Data Science lifecycle:***
  - Business Understanding: Entender cuál es el problema a resolver y su relevancia.
  - Data Acquisition and Understanding: Entender el data set, usar técnicas de exploratorio, calidad de datos, relevancia.
  - Modelling: Realizar ingeniería de características, modelos y evaluación.
  - Deployment: Establecer los siguientes pasos.

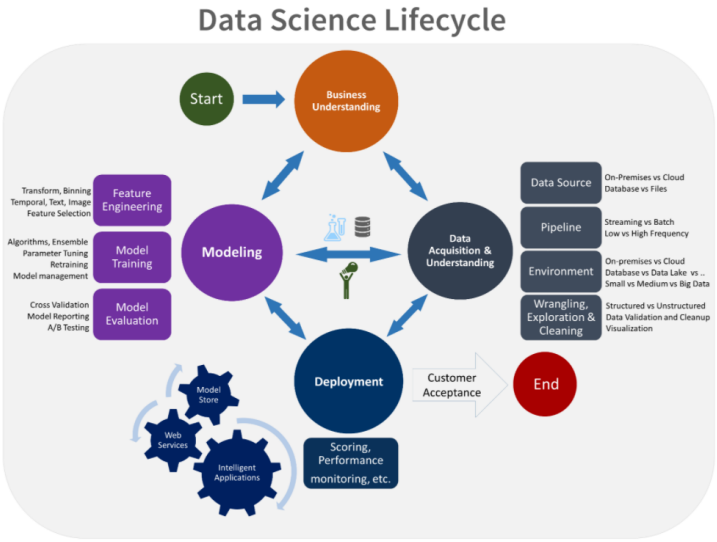


Figure 1: Modelo TDSP Microsoft Azure (2020). What is the Team Data Science Process? Recuperado de https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview

- ***Standardized Project Structure:***
  Estructura para almacenar la documentación, códigos y Dataset en GitHub.
  Dentro de este componente se especifica las tareas que tiene cada participante y los hitos importantes del proyecto.

- ***Infrastructure resources:***
  La infraestructura que se utilizará para el almacenamiento, generación y análisis de los modelos. (Local, GitHub).

- ***Tools and Utilities:***
  Dentro de este Proyecto se utilizará los siguientes softwares: Python (Anaconda, Google Colab), R (R-studio), Matlab. Con almacenamiento en GitHub.

2. Business Understanding.

2.1. Hurtos en Medellín

Las motocicletas son uno de los medios más efectivos para movilizarse en la ciudad de Medellín, puesto que ahorran tiempo y economía. Sin embargo, también se han vuelto un gran atractivo para las bandas delincuenciales que tienen diferentes zonas, horarios y modos de hurtar este vehículo.

Por esto, estrategias de seguridad pueden ser desarrolladas, teniendo en cuenta el histórico de los diferentes reportes realizados por los ciudadanos y así prevenir los posibles futuros hurtos.

2.2. Análisis de Hurtos

Para conocer cuáles son las posibles problemáticas para resolver, se debe entender cómo se debe realizar un análisis de crimen, el cual se debe tener en cuenta los 3 niveles:

1. ***Nivel Táctico****:* Estudio de incidentes criminales recientes y de la actividad criminal potencial a través del cómo, cuándo, y dónde ha ocurrido el crimen, para revisar patrones.
2. ***Nivel Estratégico:*** Estudio de problemas criminales y otros asuntos relacionados con las policías para y así evaluar patrones a largo plazo de los crímenes, y las respuestas de los procedimientos organizacionales.
3. ***Nivel Administrativo:*** Presentación de los resultados acerca de la investigación de los delitos, y su análisis en base a asuntos legales, políticos y prácticos, se comunica a audiencias como agencias gubernamentales y la ciudadanía.

Figure 2: Análisis de hurtos. Orchard M. (2014). MODELACIÓN Y PREDICCIÓN DE FOCOS DE CRIMINALIDAD BASADO EN MODELOS PROBABILÍSTICOS, Universidad Chile, Facultad de ciencias físicas y matemática.

El crimen mapping ayuda a conocer que crimen está ocurriendo, cuando y en qué lugar, identificando los puntos donde ocurren la mayor cantidad de crímenes y analizando las relaciones espaciales entre ellos ayuda a:

1. Facilitar análisis estadísticos
2. Visualizar la naturaleza del crimen
3. Vincular los eventos que pasan basado en las variables geográficas y demográficas
4. Asistir en la comunicación de resultados para generar estrategias

2.3. Problemática

Dentro de este trabajo se busca identificar las zonas de los hurtos en Medellín, respondiendo las siguientes preguntas:

1. Cuál es la incidencia porcentual de hurto en Medellín dados los datos de geoposicionamiento, hora del día, fecha y otras variables de tiempo
2. ¿Cómo se puede predecir la modalidad de hurto en las zonas?

2.4. Proceso y arquitectura del proyecto:

Figure 3: Proceso para la solución de predicción de hurtos

- El procesamiento y generación de modelos se realiza de forma local.
- Se comparten archivos por GitHub.

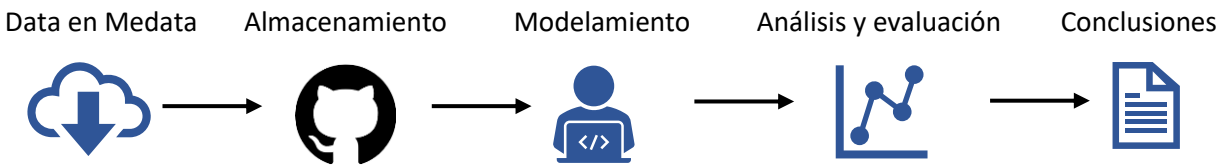


Figure 4:Data flow hurto de motos en Medellín

3. Data Acquisition and Understanding.

3.1. Dataset

El Dataset seleccionado de medata.gov.co el cual proporciona información relacionada a los hechos relacionados con la seguridad y hurtos de motos en la ciudad de Medellín.

Estos datos han sido recopilados por el Sistema de Información para la Seguridad y la Convivencia SISC.

Este Dataset consta de:

- 36 variables
- Histórico de datos de 2003 a 2019
- Total registros de 64.869
- Ultima actualización: marzo 2020

http://medata.gov.co/dataset/hurto-de-moto

3.2. Generación de variables de fechas

Se realizo la creación de nuevas variables dentro del Dataset, con dos objetivos específicos:

- Identificar si los hurtos de motos tienen comportamientos específicos dependiendo del día en que se realice el hurto, por ejemplo, ¿Es causal de aumento de hurto los días que Medellín se encuentre en Ferias y fiestas?
- Analizar como se puede predecir la probabilidad de hurto de motos en momentos específicos, por ejemplo, la probabilidad que el hurto de motos por halado se ejecute en las mañanas.

Las variables creadas son:

- Año
- Día semana
- Festivos
- Ferias y fiestas
- Semana del año
- Día
- Mes
- Hora
- Quincena
- Franja Horaria
- Hora, Minutos, Segundos

3.3. Exploratorio

Se realiza un exploratorio del Dataset en el cual se busca entender el Dataset, conocer su calidad y conocer cómo será el tratamiento de valores nulos.

Como se observa a continuación vemos que el crimen en Medellín tuvo un aumento en el hurto de motos del 216% durante el periodo de 2010 a 2011, y desde allí el hurto intenta estabilizarse con un promedio 4634 hurtos por año.

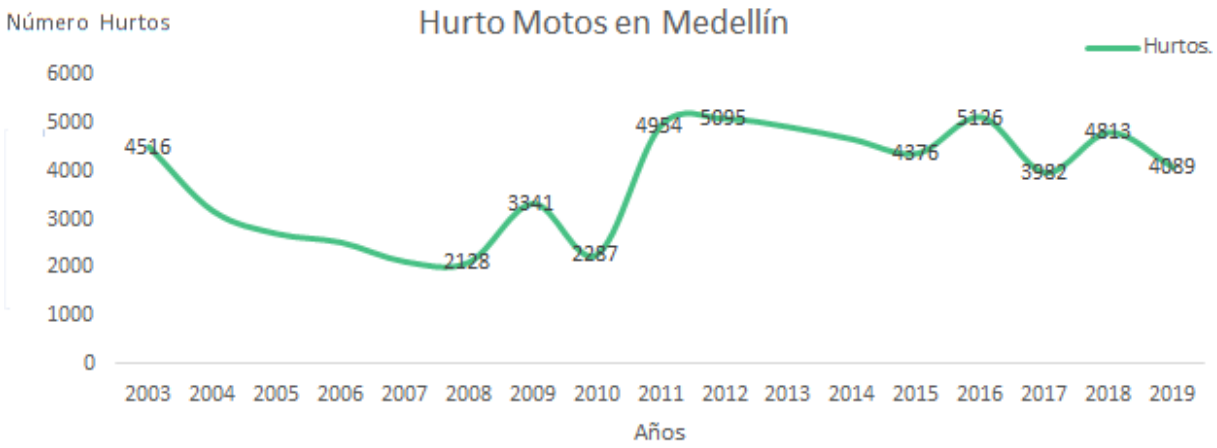


Figure 5: Hurtos en Medellín por año

Como se ve en la siguiente tabla, se evidencia que de 10 personas a las que se les hurta su moto, 2 de ellas son mujeres.

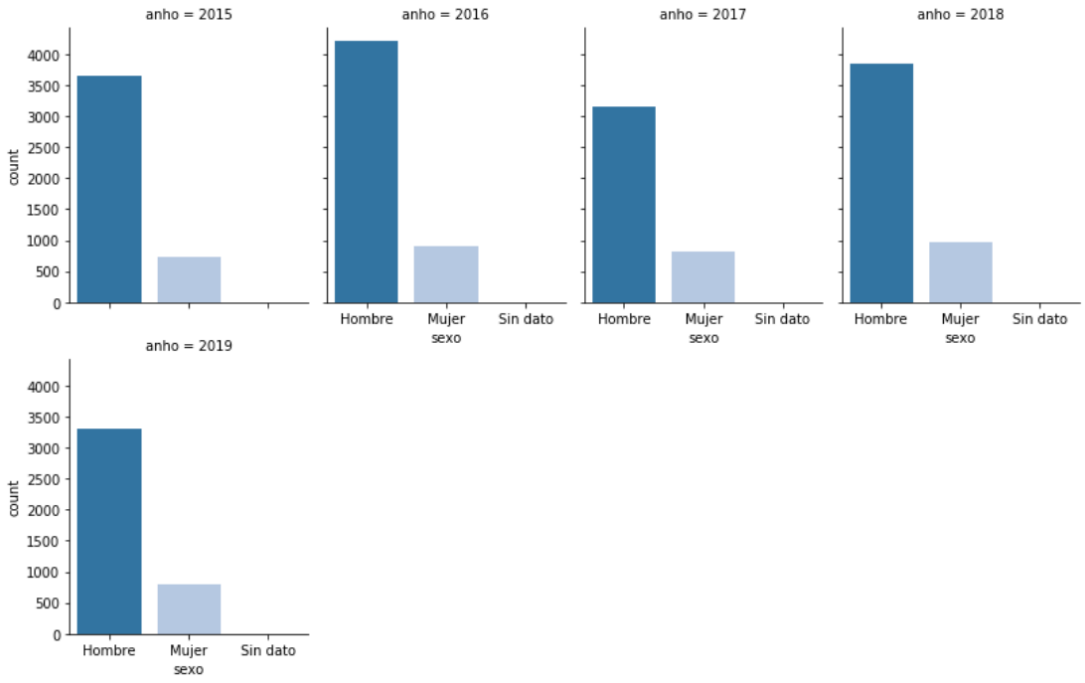


Figure 6: Últimos años de hurtos por genero

Table 1: Cantidad de hurtos por genero

Genero	Cantidad Hurtos
Hombre	527.771
Mujer	10.447
Sin Datos	1.661

También se puede observar que las edades de las victimas a las que mas roban se encuentran entre 20 y 30 años. Y adicional, se puede observar que la calidad de esta variable no es la mejor, ya que se encuentran edades entre 0 y 14 años y edades mayores a 70.

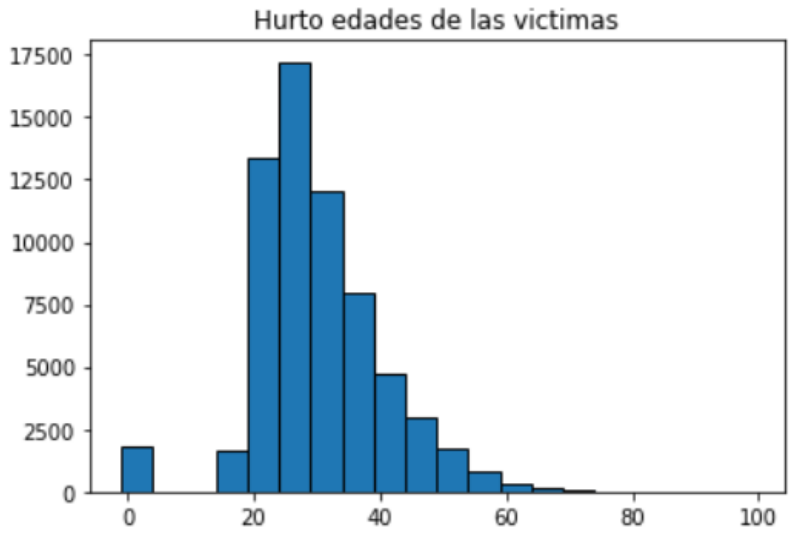
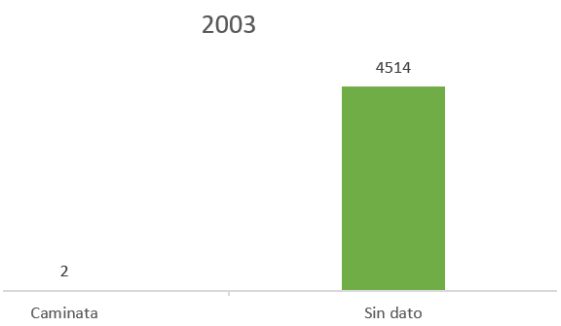
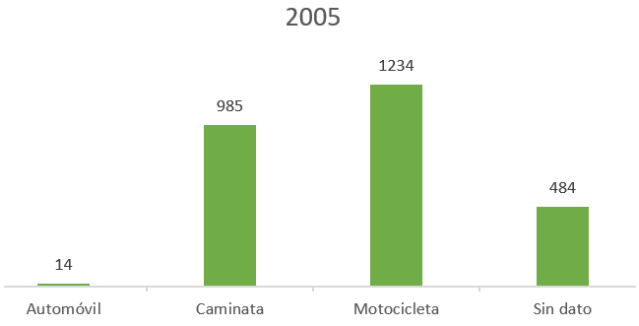


Figure 7: Distribución de edad de victimas

A continuación, se puede apreciar los diferentes de los criminales para hurtar las motos de los ciudadanos en Medellín, donde se puede observar que, al principio de la década, no se registraba este data, sin embargo, a mediados de 2005 se empieza a registrar que el medio de transporte mas usado por los delincuentes es la motocicleta. Durante el 2015 se empieza a observar nuevos medios de transporte como lo son la bicicleta y taxi. Finalmente, desde 2016 en adelante se observa que el medio de transporte mas utilizado cambia a caminata.





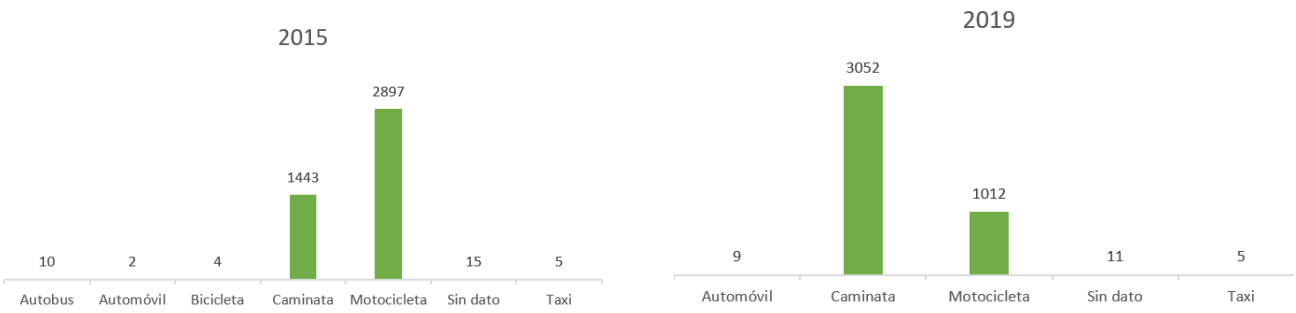


Figure 8: Transporte utilizado por los criminales para el hurto de motos

La modalidad de hurto se puede observar que existen dos modalidades muy comunes que son Halado con un 51% y Atraco con un 39% y como se puede observar, en las siguientes gráficas, a pesar de que aparecen nuevas modalidades durante los años, las modalidades de hurto y halado se mantienen como las más usadas.

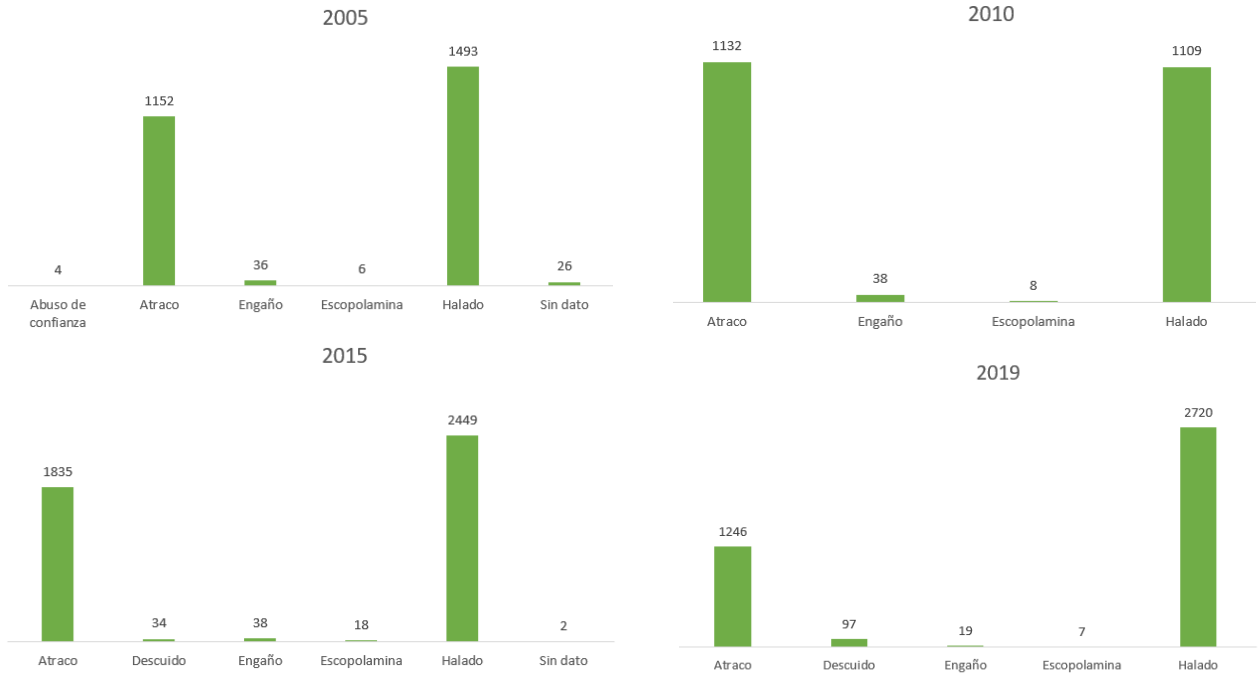


Figure 9: Número de hurtos por año por modalidad

Dentro del exploratorio, se evidencia que la comuna en la que más hurtos se presentan es en la comuna 10 que representa el 44% de los hurtos de motos, que se concentran en el barrio Prado, Boston y La Candelaria.

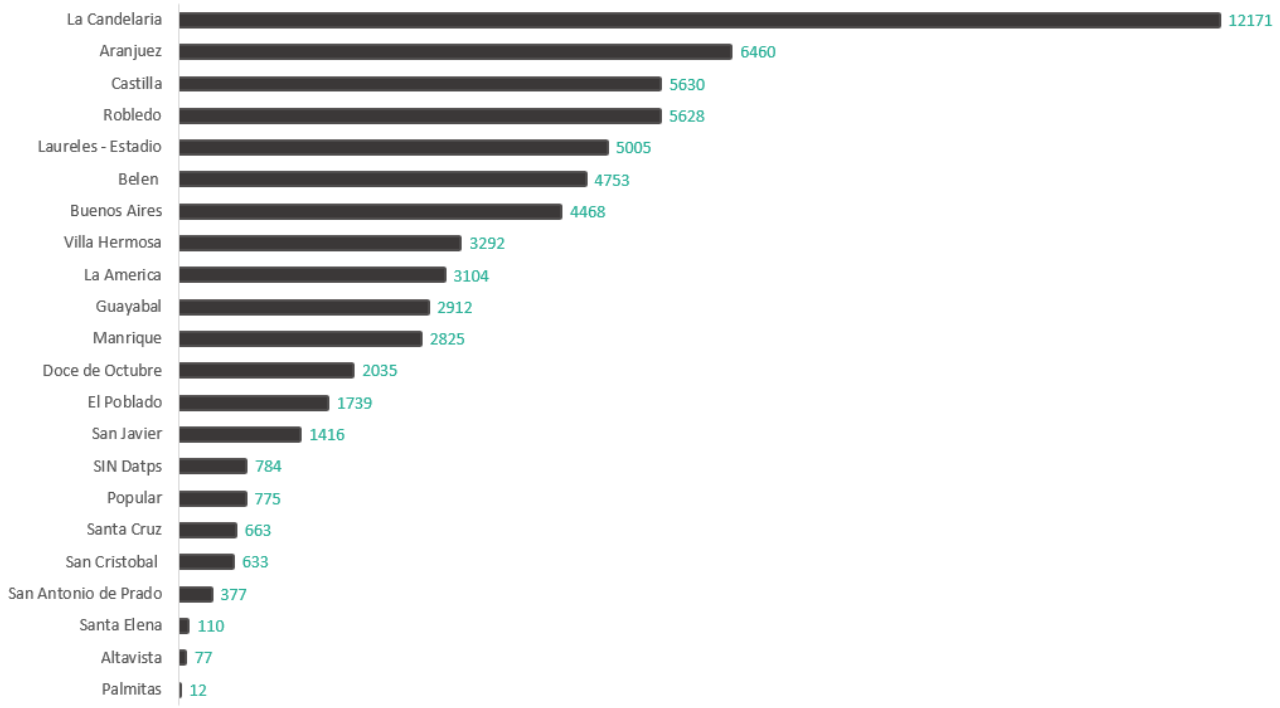


Figure 10: Número de hurtos totales por comuna

Finalmente se puede observar que los miércoles son los días mas populares para el hurto de Motos, y los domingos durante todos los datos los años es el día menos popular para realizar esta actividad.

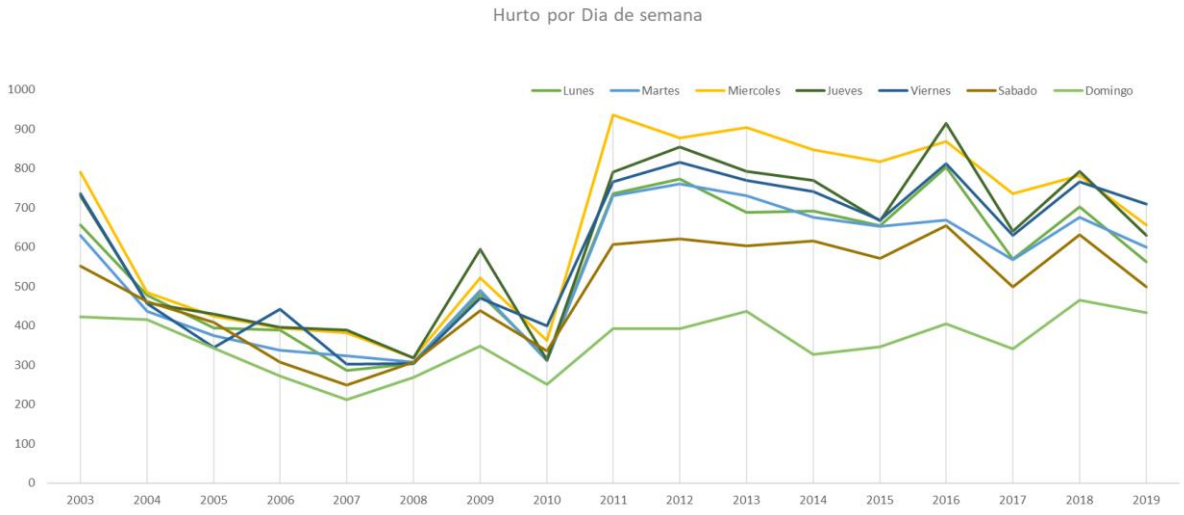


Figure 11: Hurtos en los diferentes días de la semana

3.4. Correlación

Con el fin de determinar las variables que están más relacionadas y así mismo poder realizar una reducción dimensional, teniendo en cuenta que las variables seleccionadas representan más del 60%, la correlación de variables es definida por Pearson para una muestra de datos como:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\,\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Su resultado es el rango [-1, 1], pero ¿qué pasa si los datos son categóricos? ¿Cómo se resta del promedio de la característica? Una solución muy usada es el método one-hot encoding, pero esto hace que cada variable se divida en la cantidad de características disponibles haciendo que las variables aumenten considerablemente, lo que necesitamos es una medida de asociación entre características categóricas, para esto utilizaremos el coeficiente V de Cramer, la cual se basa en una variación nominal de la prueba de Chi-Cuadrado de Pearson, el resultado de la correlación está en el rango [0, 1] donde 0 significa que no hay asociación y 1 es una asociación completa, en el algoritmo construido para calcular la correlación de las variables se tienen en cuenta:

- La matriz de confusión o tabulación cruzada se calcula con la función crosstab de pandas, de las dos variables a comparar.
- La matriz de contingencia se calcula usando la función chi2_contingency () de la librería scipy, aplicada a la matriz de confusión anterior.
- Con la matriz de contingencia se calcula el coeficiente phi:

$$\chi^2 = \sum_{ij} \frac{\left(n_{ij} - \frac{n_i n_j}{n}\right)^2}{\frac{n_i n_j}{n}}$$

$$\varphi = \sqrt{\frac{\chi^2}{n}}$$

- Con estos tres objetos podemos calcular el coeficiente V de Cramer:

$$V = \sqrt{\frac{\varphi^2}{min(k-1, r-1)}}$$

k: *número de columnas*
r: *número de filas*
n: *gran total de observaciones*

Se puede ver que existe una correlación completa entre las variables 7(nombre barrio) y 8(código barrio) ya que son variables 100% relacionadas debido a que una es la codificación de la otra.

No se encuentra relaciones mayore al 80%, lo que nos indica que las variables son independientes entre ellas.

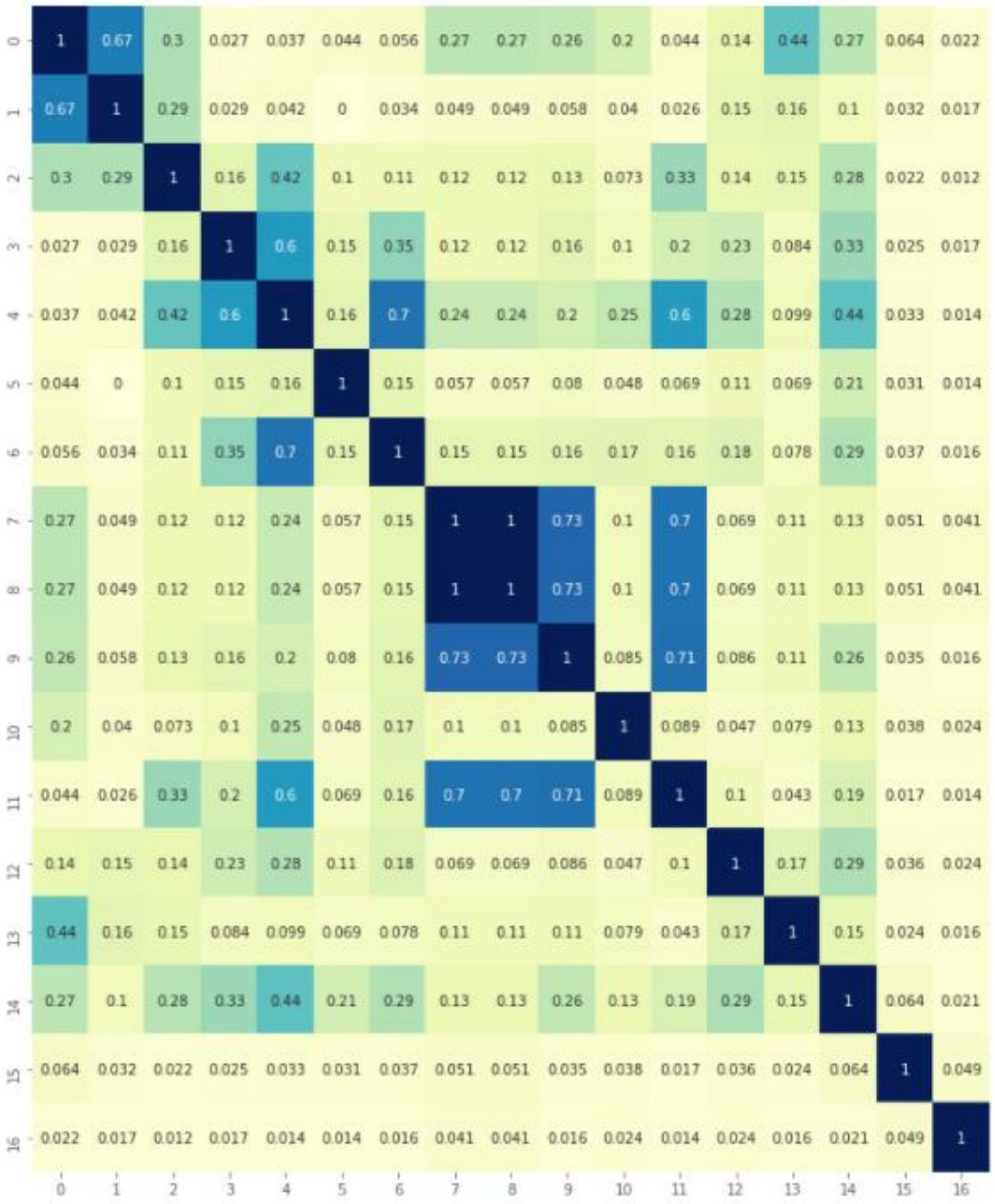


Figure 12: Correlación de variables por variable de Cramer

Se realizó una transformación por medio del método “one-hot enconding” con el fin de realizar una correlación entre todas las variables tanto categóricas como discretas. El resultado muestra que existe una relación 100% entre las variables nombre barrio y código barrio.

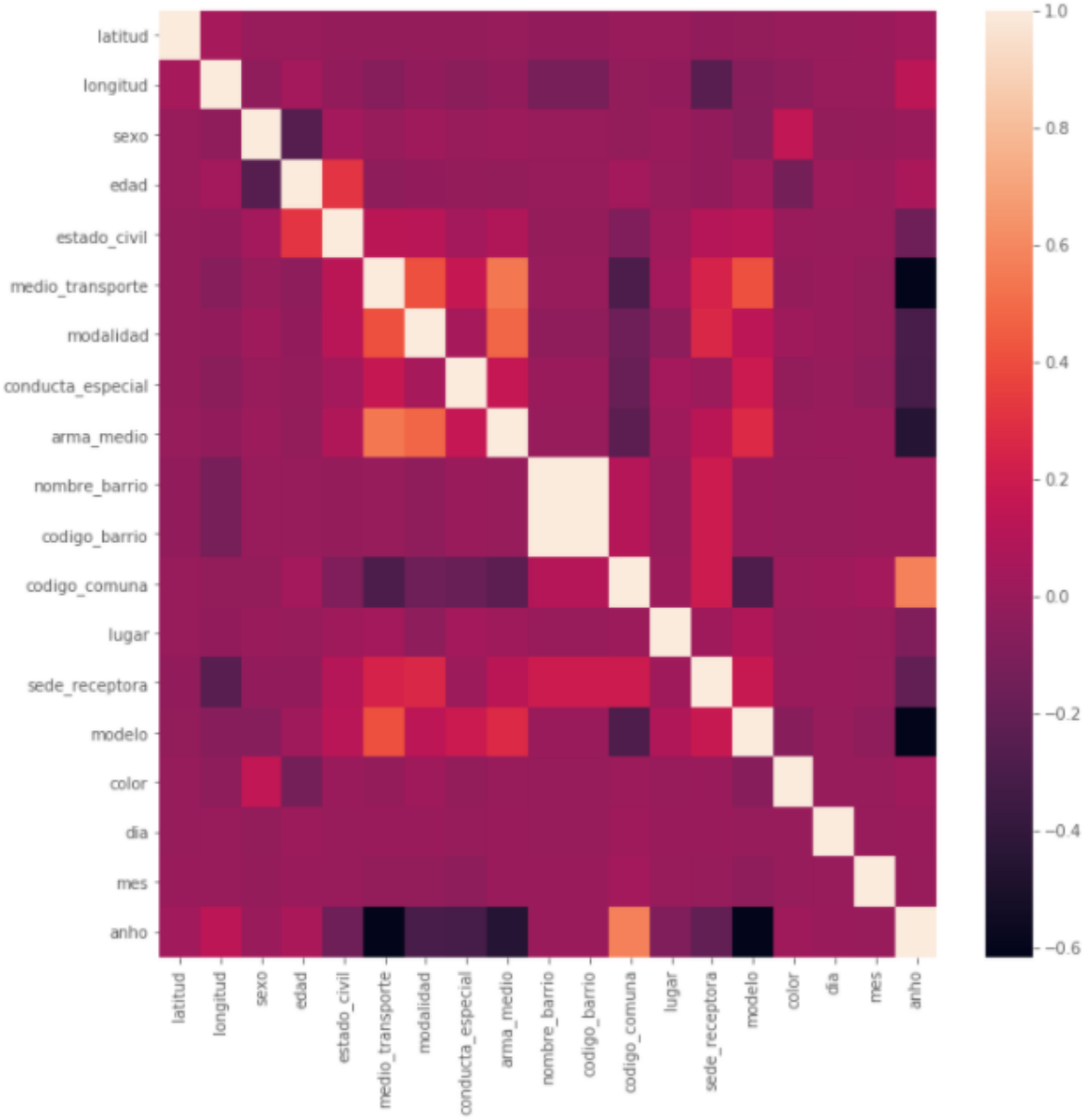


Figure 13: Correlación con coeficiente Pearson

Por medio del cálculo de mutual information, se revisa el ranking que tiene las variables contra la variable dependiente que se tomará para el estudio de los hurtos en Medellín.

Las variables que generan más información para la variable dependiente son el arma utilizada, medio de transporte del asaltante.

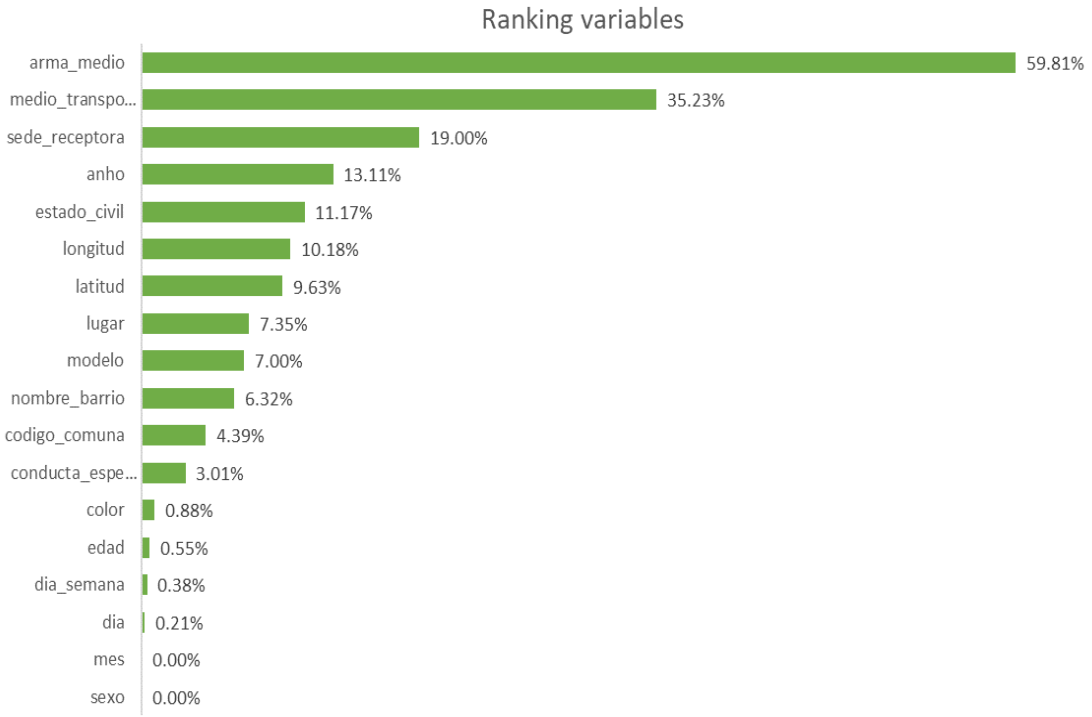


Figure 14: Ranking de variables

3.5. Data Quality (Preprocesamiento)

Para poder mejorar la calidad de los datos del Dataset, se realizan los siguientes ajustes de los datos:

- La longitudes y latitudes que se encuentran nulas son reemplazada por la longitud y latitud mas comunes de las comunas a las que pertenece el registro.
- Se cambian las edades menores a 13 por la media, ya que la edad mínima para conducir es de 14.
- Se agrupan las modalidades con frecuencia pequeña por una genérica “Otros”.
- Se eliminan los registros que tienen toda la ubicación geográfica nula.
- Se crea un nuevo Dataset, en el cual se cambia las variables categóricas por un número que las representará. Esto se realiza con el fin de poder alimentar los modelos y poder realizar las mediciones y procedimientos necesarios.

4. Modelamiento

4.1. Ingeniería de Características

Con el fin de seleccionar las variables relevantes de un total de variables del Dataset de hurtos para los diferentes modelos para conocer la probabilidad de hurto de robos. Esto ayudara a que nuestros modelos sean mas rápidos y con mejor desempeño.

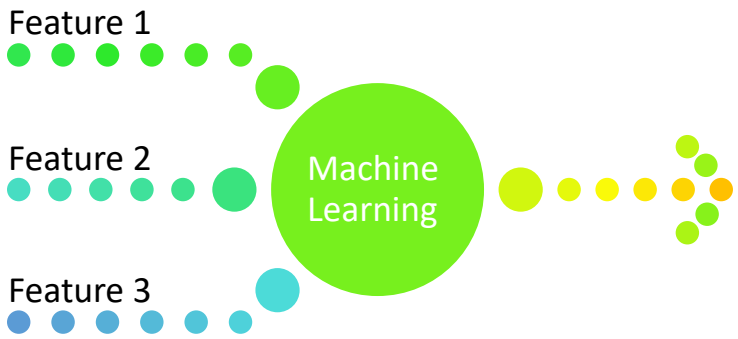


Figure 15: Feature Selection

La selección de características se realiza con diferentes fines:

- Los modelos son más fáciles de entender.
- Los tiempos de procesamiento se acortan.
- Se reduce overfitting.
- Se reduce el riesgo de errores en la data.
- Se evita la redundancia de variables.
- Reducir la dimensionalidad, ayuda a una mejor predicción de los modelos.

La selección de características se realizo con los siguientes métodos:

4.1.1. Métodos de filtro

Los métodos de filtros, es el primer tipo de método que se debería utilizar en cualquier ingeniería de características, ya que este tipo de método se utiliza para cualquier tipo de algoritmo de Machine Learning y se enfoca solamente en las características de la data.

Beneficios de este método:

- Modelo agnóstico
- Computación rápida
- Es un método muy económico (computacionalmente)

Este tipo de métodos se realiza en dos pasos:

1. Clasificar las características de acuerdo con un criterio
2. Seleccionar las mejores características de acuerdo con la calificación dada anteriormente

4.1.1.1. Constantes

Dentro de este paso, se seleccionan aquellas variables con el mismo valor en todas las observaciones, por lo que estas características no tienen un poder de discriminación y por ende no genera información a los modelos de ML.

Para este trabajo se utilizó Variance Threshold de la librería de Sklearn.feature_selection, donde se indica que se debe identificar aquellas variables con una varianza igual a 0.

Este primer método elimina un total de 15 Variables, que, al explorar sus valores únicos, nos muestra que estas variables tienen valores constantes o en el caso de las categóricas "Sin Dato" en todas las observaciones.

Variables para eliminar:

```
Index(['cantidad', 'grupo_actor', 'actividad_delictiva', 'parentesco',
       'ocupacion', 'discapacidad', 'grupo_especial', 'nivel_academico',
       'testigo', 'conducta', 'caracterizacion', 'articulo_penal',
       'categoria_penal', 'permiso', 'unidad_medida'],
      dtype='object')
```

Figure 16: Variables constantes

4.1.1.2. Quasi constantes

Este tipo de selección permite identificar aquellas variables que tienen el mismo valor en casi todas las observaciones, por lo que su varianza es muy cercana a 0.

Para identificar estas variables se utilizará la misma librería indicada para la técnica de identificación de constantes, con una edición en el Threshold = 0.01.

En este caso se eliminan 3 variables adicionales, en las que se pueden ver que el 99% de las observaciones tienen un valor especifico en sus variables. Adicional se puede identificar registros que deben ser eliminados del Dataset, ya que no aportan información.

Table 2: Variables quasi-constantes

Variable	Bien		categoria_bien		grupo_bien	
Proporcion	Moto	0.999532	Vehículos de 2 o 4 ruedas	0.999984	Vehículo	0.999984
	Moto carro	0.000359	Alimento	0.000016	Mercancía	0.000016
	Cuatrimoto	0.000094				
	Carne	0.000016				

4.1.1.3. Variables duplicadas

En esta técnica se busca eliminar esas variables que son idénticas entre sí. Para identificar estas variables, se necesita iterar todas las características y para cada una de ellas buscas las características que mas se parece a ella.

Dentro de este Dataset se encuentra que el código de barrio y el nombre de barrio son duplicados, ya que a pesar de que no tienen exactamente el mismo valor en las variables, la asignación del nombre del barrio tiene una codificación y por ende nuestro algoritmo también lo toma como una duplicidad de información.

4.1.2. Correlación

Dentro de la selección de características, también es válido realizar una visualización de la correlación de las variables, ya que estas ayudan a identificar la asociación que existe entre dos variables. La correlación indica si existe alguna relación predictiva.

La correlación mide la relación lineal que tienen dos variables, por lo tanto, si dos o más variables son altamente correlacionadas, se indica que una puede predecir la otra. Por lo que las variables independientes deben ser altamente correlacionadas con la variable objetivo, pero no pueden estar correlacionadas entre si ya que esto genera redundancia.

Beneficios de la correlación en la ingeniería de características:

- Reducir la dimensionalidad, ayuda a la precisión del modelo
- La correlación puede afectar la interpretabilidad de los modelos lineales
- Los clasificadores tienen cierta sensibilidad a la correlación.

Dentro de este trabajo se realizan dos métodos de correlación para identificar las características más correlacionadas, ambos métodos se calcularon por la correlación por medio del coeficiente de Pearson.

$$\frac{Sum\ ((x_1\ -\ x_{1\,mean})\ *\ (x_n\ -\ x_{n\,mean}))}{VarX_1\ *VarX_n}$$

Donde sí nos muestra 1 son variables muy correlacionadas y -1 son variables anti correlacionadas correlacionadas.

4.1.2.1. Brute force function

Donde se busca la correlación de las características sin esperar generar ningún conocimiento futuro.

Con Pandas se utiliza la función “corr” en la cual se genera una matriz de características tanto en las filas como en las columnas y por medio de un heatmap de la librería seaborn, donde se puede visualizar la correlación que tiene cada variable:

Se tomará como una alta correlación cuando dos variables estén correlacionadas > 0.80 o > -0.80.

Una vez visualizado la correlación, se genera una función en la cual se realiza un ciclo en el cual se recorre característica a característica y se compara una característica a la vez, con todas las características de la fila. Si la correlación es mayor a la dada en el threshold (0.8 o -0.8) entonces se agrega a una matriz de variables correlacionadas y así poder ser identificadas.

El resultado muestra que existe una relación 100% entre las variables nombre barrio y código barrio, ya que una es la codificación de la otra.

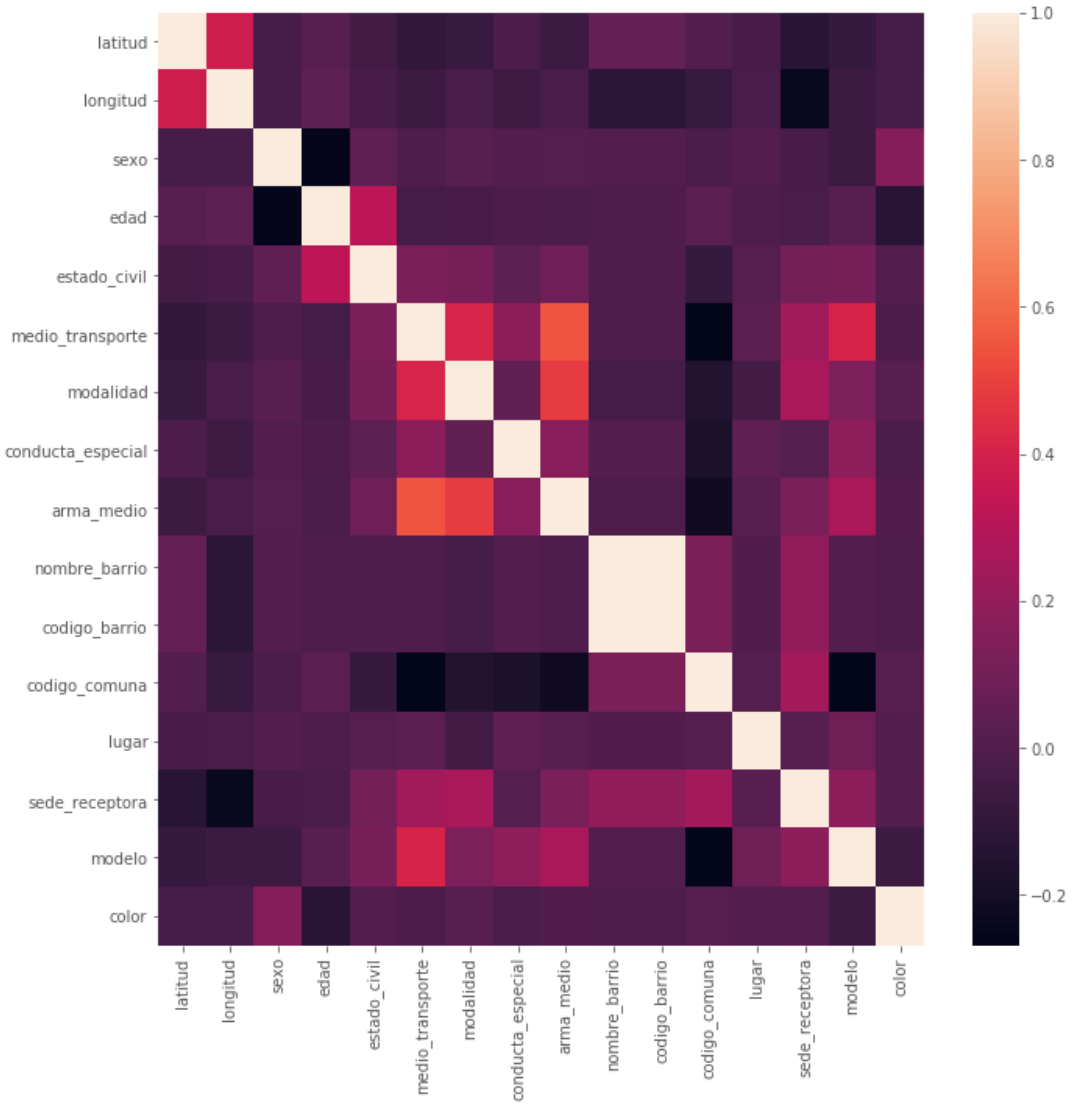


Figure 17: Correlación ing Característica, Método Brute

4.1.2.2. Correlación por grupos

Se busca encontrar grupos de correlación entre características y así identificar si dos o más variables están correlacionadas entre grupos y analizar si se debe o no eliminar un grupo específico.

Para este método se crea nuevamente la matriz con la librería pandas, luego se transforma todas las correlaciones a valor absoluto, con esta información, se crea un dataframe que contiene el nombre de la característica 1 y 2 con su respectiva correlación.

Luego, se crea una función para generar grupos correlacionados, donde se genera una lista en la cual se almacenan las características ya analizadas, y una segunda lista de los grupos correlacionados. Luego, se llama cada una de las variables de la primera columna del dataframe creado anteriormente, para correlacionarlo con todas las variables y finalmente se agrega la característica al grupo correlacionado.

Con este método se busca identificar esas variables que están correlacionadas con mas de una variable, y así poder estudiar si se debe eliminar una variable que esta correlacionada con varias, o eliminar todas las variables que están relacionadas a la misma variable.

En este proyecto no se da este tipo de caso.

4.1.3. Statistical Measurements

Este tipo de método evalúa cada característica de forma individual, sin embargo, están alineados con la variable objetivo. Este método se evalúa de forma estadística por lo que se aplica las siguientes técnicas para identificar las mejores características del Dataset:

Table 3: Técnica para la ganancia de información

Técnica	Función
Information gain	Mutual Information
Univariante Test	Anova

4.1.3.1. Information gain

En esta técnica se mide la ganancia de información o cuanta información genera una variable sobre otra, la dependencia se mide por la probabilidad de la variable X sobre la Y con la siguiente ecuación:

$$I(X;Y) = \sum_{y\epsilon Y}\sum_{x\in X} p(x,y)\ \ (x,y)\log(\frac{p(x,y)\ \ (X,Y)}{px(x)\ \ py(y)})$$

- Si X y Y son independientes entonces este será igual a 0.
- Si x es determinístico de Y entonces X será diferente de 0.

Para seleccionar las variables, nos interesa la información mutua entre las variables de predicción y el objetivo. Valores más altos de información mutua, indican poca incertidumbre sobre el objetivo Y dado el predictor X.

Para este trabajo se tomó la librería Scikit-learn, para determinar la información mutua entre una variable y el objetivo utilizando el mutual_info_classif y mutual_info_regression para objetivos binarios o continuos.

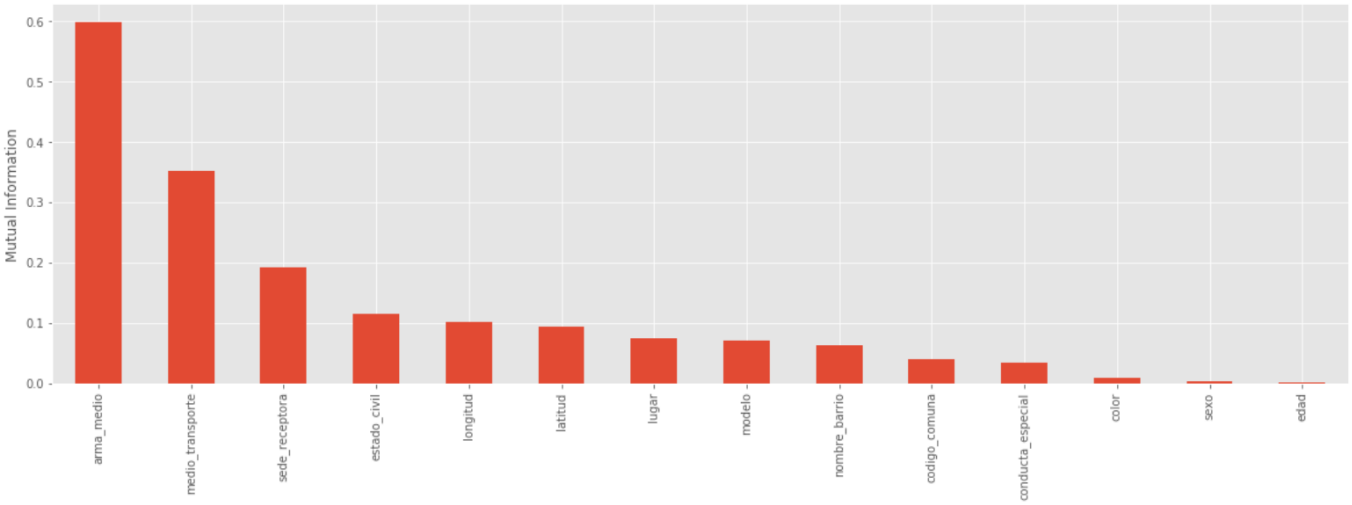


Figure 18: Mutual Information

Como se puede observar, para predecir la variable modalidad, el tipo de arma utilizada da más información que la demás variable, y se puede ver que el color, genero y edad son variables que generan ninguna información.

4.1.3.2. Univariante Test

Por medio de la Anova, o la evaluación estadística de la hipótesis que dos variables tienen la misma media. Para esto, se calcula:

1. Un dataframe en el cual se diferencie las variables de las Y, una columna donde se encuentren las observaciones sea=0, y otro ejemplo donde las observaciones en y sea=1
2. Great Mean: Promedio de todas las observaciones
3. Model sum of square:

$$\sum Obs(Mean - great\ mean)^2$$

4. Residual sum of squares:

$$\sum (Obs(0) - mean0)^2 + \sum (Obs(1) - mean1)^2$$

5. Mean Square model:

$$\frac{Model\ sum\ of\ squared}{Column\ (1)}$$

6. Mean Square error:

$$\frac{Residual\ sum\ of\ squared}{Total\ Obs - 1}$$

7. F-Statistics:

$$\frac{Mean\ Squared\ model}{Mean\ Squared\ error}$$

Para el algoritmo, lo primero a realizar es calcular los p-values de cada una de las variables y organizarlos de mayor a menor, lo que nos indica que aquellos P-values con un valor mayor a 0.05 no son relevantes para el modelo y por ende pueden ser eliminados

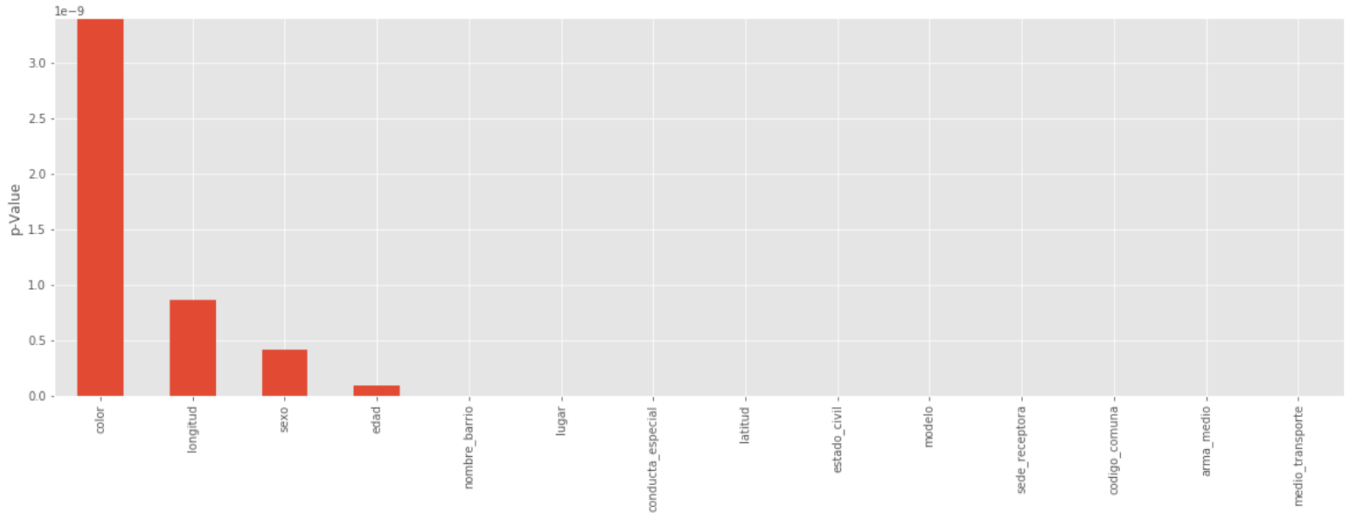


Figure 19: Anova

4.1.4.Step Forward Selection

La motivación detrás de los algoritmos de selección de características es seleccionar automáticamente un subconjunto de características que sea más relevante para el problema. El objetivo de la selección de características es doble: Queremos mejorar la eficiencia de los cálculos y reducir el error de generalización del modelo eliminando las características irrelevantes o el ruido.

Utilizando la librería mlxtend se utiliza el método Sequential Feature Selector el cual tiene el siguiente procedimiento, teniendo en cuanta que se debe tomar toda la dimensión de las características del dataset.

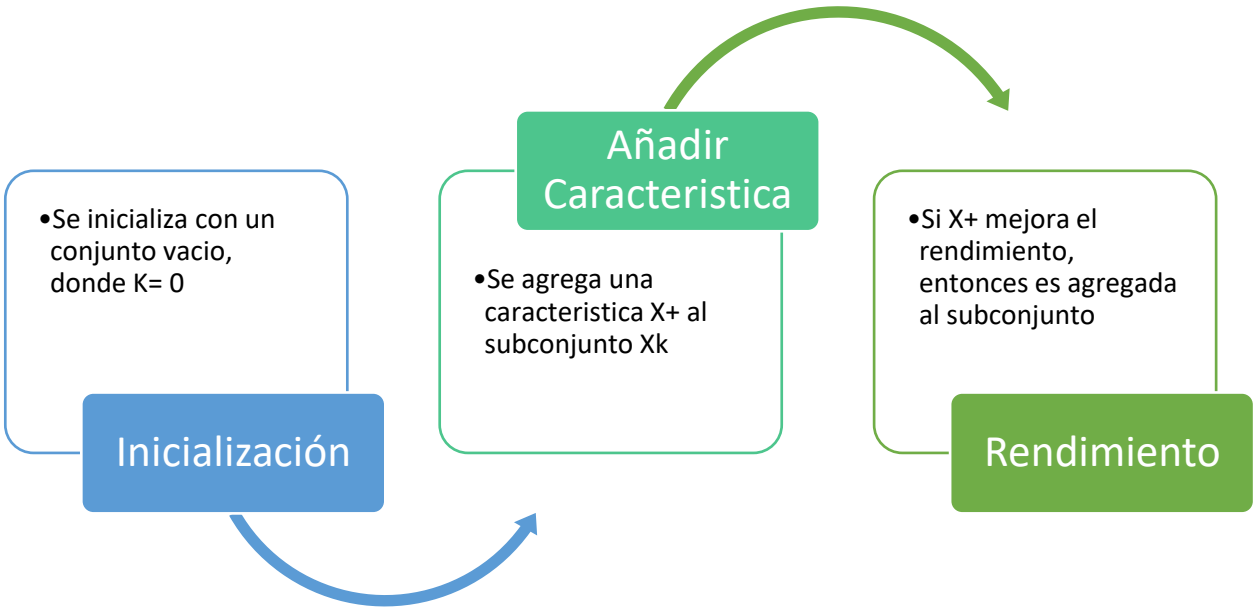


Figure 20: Proceso sfs

El Sequential Feature Selector devuelve un subconjunto de características; el número de características seleccionadas k, donde k < d (dimensión dataset original), tiene que ser especificado a priori.

El modelo realizado con el hurto de motos en la ciudad de Medellín muestra la siguiente selección de características:

- ocupacion
- medio_transporte
- nivel_academico
- testigo
- arma_medio
- articulo_penal
- categoria_penal
- sede_receptora
- grupo_bien
- unidad_medida

4.1.5.Dataset final

El Dataset que será utilizado para los modelos que se implementaran para responder los objetivos de este proyecto consta de:

- 12 variables originales del Dataset:
    - fecha_hecho, latitud, longitud, estado_civil, medio_transporte, modalidad, arma_medio, nombre_barrio, codigo_comuna, lugar, sede_receptora, modelo
- 11 variables generadas en el exploratorio.

4.2. Clustering

4.2.1.Hot Spots

Con el fin de detectar y visualizar de una forma fácil el comportamiento de los hurtos de motos en Medellín durante el periodo de 2003 y 2019, se genera un mapa hotspots, en la cual por medio de longitudes y latitudes se detectan las áreas con grandes densidades de delitos y poder conocer de forma exploratoria como se debería evaluar el algoritmo de Clustering.

Como se puede visualizar a continuación el crimen no se distribuye uniformemente a lo largo de Antioquia, por lo que se ve que algunas zonas se muestran como zonas rojas que indican que hay más concentración de crimen que en otras áreas.

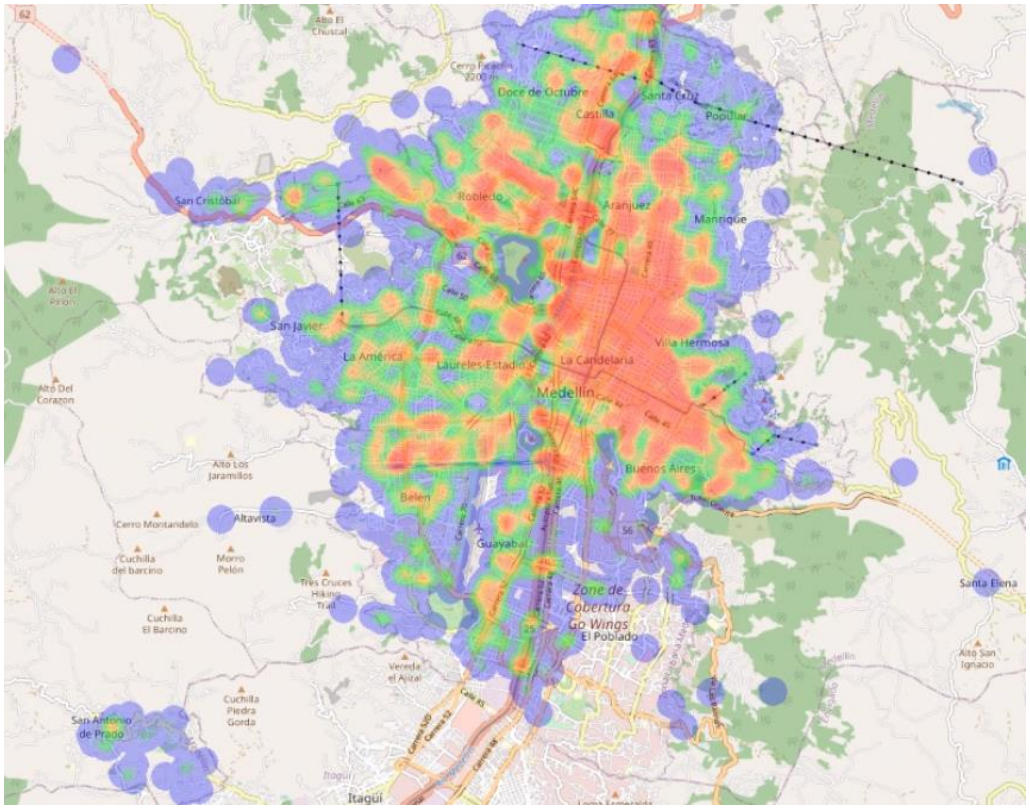


Figure 21: HotSpot por año sobre el hurto de motos

Este tipo de exploración ayuda a explicar la salida del modelo, también puede proporcionar un patrón que ayuda a guiarnos si es recomendable usar varios modelos, acá es donde la clusterización con aprendizaje no supervisado y el Clustering puede ser una característica muy importante. La agrupación puede darnos muchas pistas de información, podemos saber por grupo como es su comportamiento por días, por horas, etc.

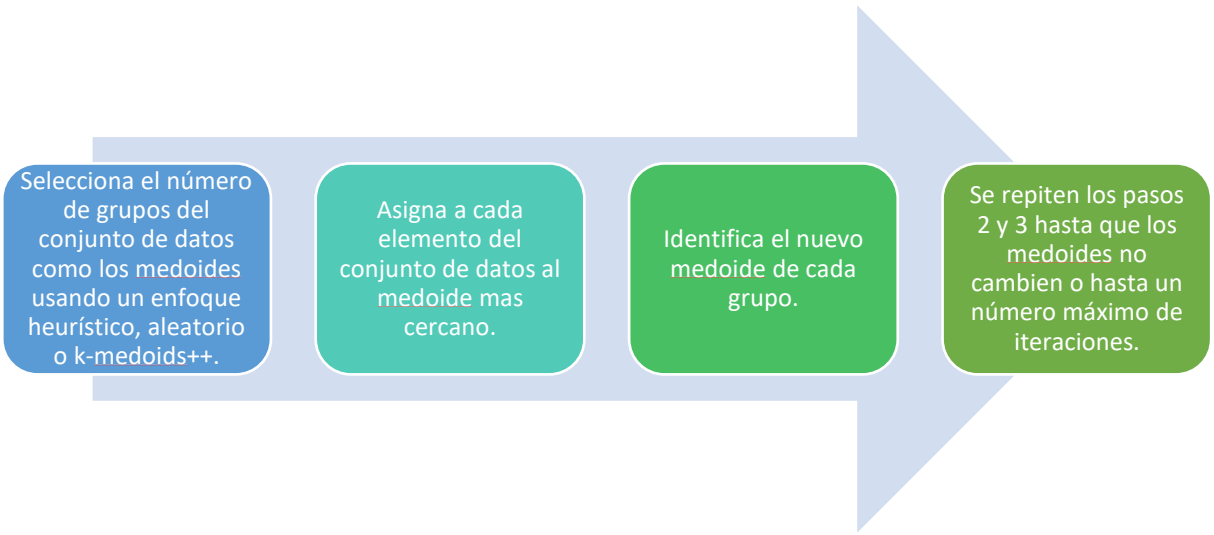
4.2.2.Algoritmo de Clustering de zonas

Se utilizarán las coordenadas de latitud y longitud, así como la frecuencia del hurto para realizar un agrupamiento de tal manera que los datos de cada clúster sean similares entre sí y que los datos de los grupos separados sean menos similares, para este caso la distancia es un factor más relevante para ello.

Para el agrupamiento se utilizará el algoritmo modificado de *k-means* llamado *k-medoids*, se utilizará este algoritmo porque se basa en el *medoide* que es un grupo de datos, a diferencia del centroide, que tiene la menor distancia total a los otros miembros de su clúster. Esto quiere decir que es más robusta frente a los valores atípicos, aunque *k-means* es más eficiente, *k-means* intenta minimizar la suma de cuadrados dentro del grupo, mientras que *k-medoids* intenta minimizar la suma de distancias entre cada punto y él *medoide* de cada grupo [1].

Se determina entonces utilizar *k-medoids* ya que el tiempo del procesamiento para este caso no es inconveniente, el entrenamiento solo será realizado una vez.

Para entrenar el modelo se utilizó la librería *sklearn_extra* usando la función *K-medoids*, de la siguiente manera:



4.2.2.1. Selección de número de grupos

Para la medida de similitud será expresada en términos de distancia, utilizando la distancia mínima de euclidiana, para definir la cantidad de grupos eficientes utilizaremos el criterio de los coeficientes de silueta y el score de distorsión por método del codo.

- ***Coeficientes de silueta.***

Comparando los dos promedios más altos de los coeficientes de silueta, se puede visualizar en la siguiente grafica que según el promedio del coeficiente de silueta k=5 es el mejor de los k evaluados.

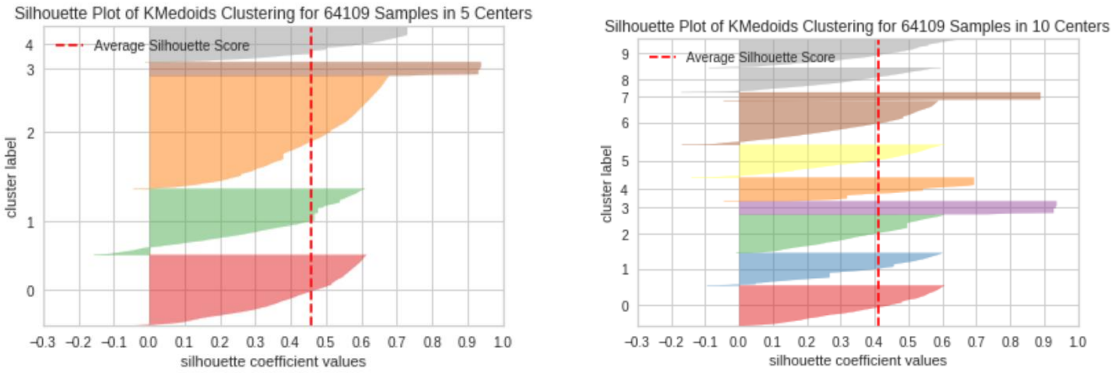


Figure 22: Coeficiente de silueta

- ***Coeficiente de distorsión.***

Como se puede apreciar en el siguiente gráfico, por medio del coeficiente de distorsión se sugiere que el número k debería ser 5, por lo que es el grupo más eficiente según el coeficiente.

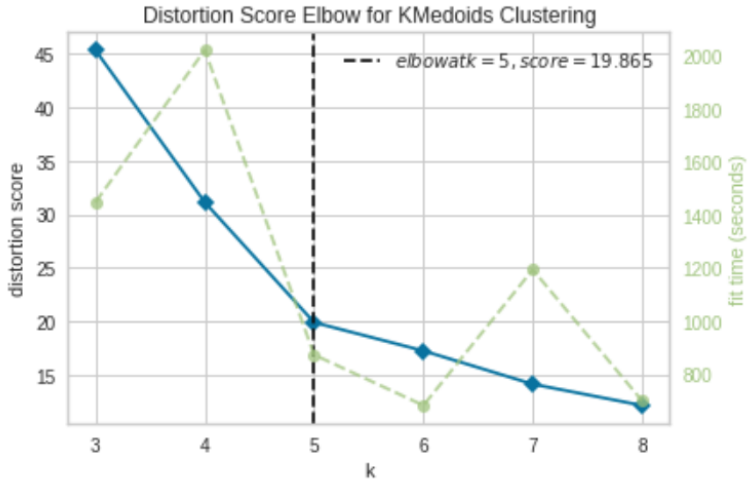


Figure 23: Coeficiente de distorsión

4.2.2.2. Resultado de clusterización por zonas k = 5

En la siguiente gráfica se puede visualizar como se distribuyen los 5 grupos de las zonas generadas por el modelo k-medoids.

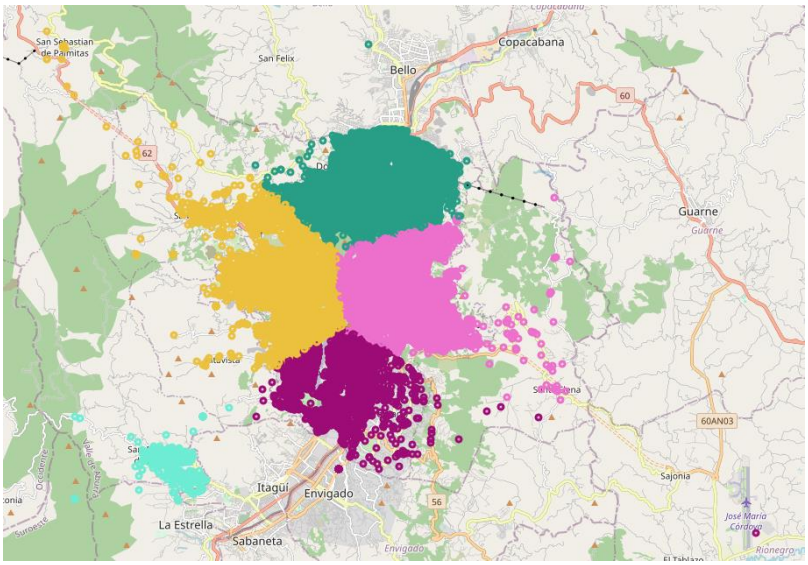


Figure 24: Clustering K-medoids con K=5

4.2.3.Divisiones por Cuadrantes o Crime Mapping.

Se realizó una división de 60x60 cuadriculas utilizando las variables latitud y longitud creando así una matriz de 60x60 en cada cuadrante se le asignó una etiqueta que corresponde al número de la fila y al número de la columna, se crea un dataframe con la cantidad de hurtos por cuadrante por año y se realizó un agrupamiento usando nuevamente el algoritmo de *k-medoids* con k=50 y 100.

Este mapa es conocido como Crime Map, se muestra por cuadrantes y según la densidad del crimen en ese cuadrante se le asigna un color de un termómetro desde los puntos fríos, puntos con menos densidad de robo, hasta los puntos más calientes, se utilizó la escala de calor en los colores del espectro visible por el ojo humano. Desde el violeta representando el color más frío hasta el ojo como el color más cálido

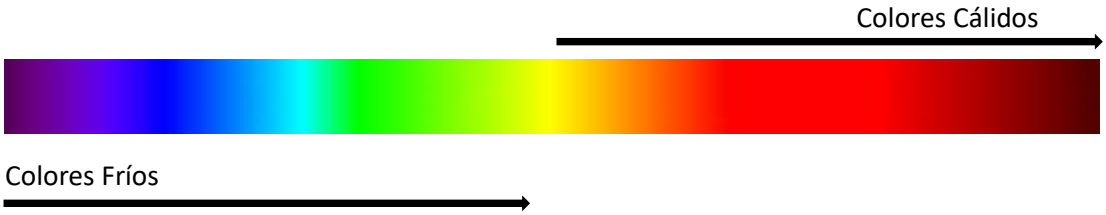


Figure 25: Referencia de color en Crime Map

Para graficar los mapas se utilizó una librería llamada Folium, de código abierto, que permite crear varios tipos de mapas *Leaflet.* Estos mapas son interactivos por lo que es muy útil para cuadros de mando, en este informe no utilizaremos esta función.

En la siguiente gráfica se puede visualizar la división del mapa de Medellín por cuadriculas.

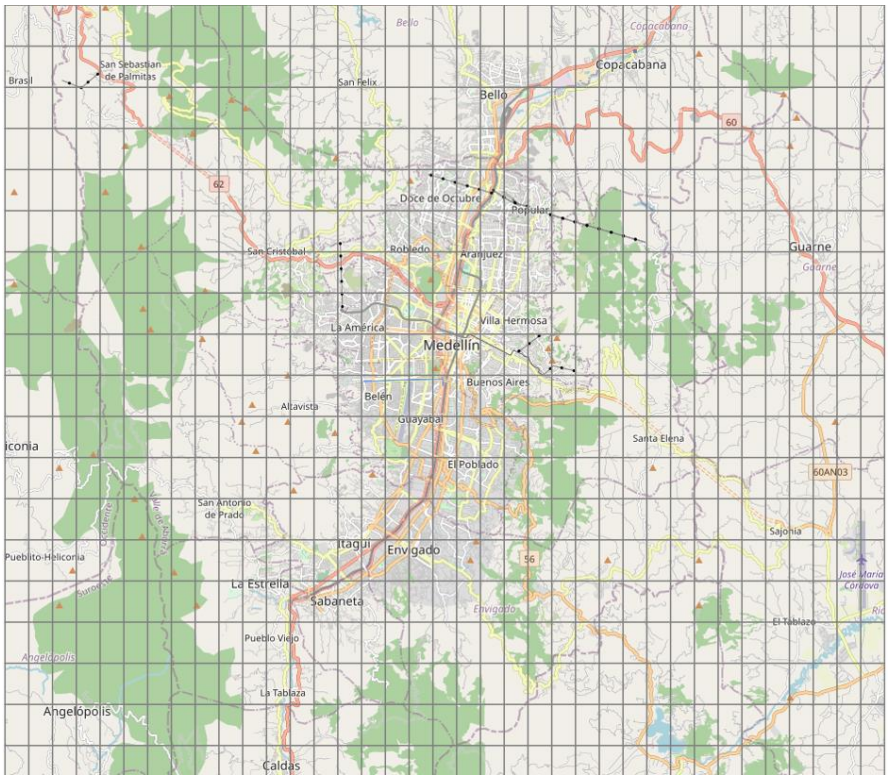


Figure 26: Mapa con cuadriculas de Medellín

Se tomo este número K, buscando una agrupación de las cuadriculas que nos permitiera tener grupos homogéneos de cuadrantes, que pudieran corresponder a un mismo patrón de comportamiento en cuanto la cantidad de siniestros por año, de tal forma que pudiese modelarse conjuntamente.

50 Grupos:

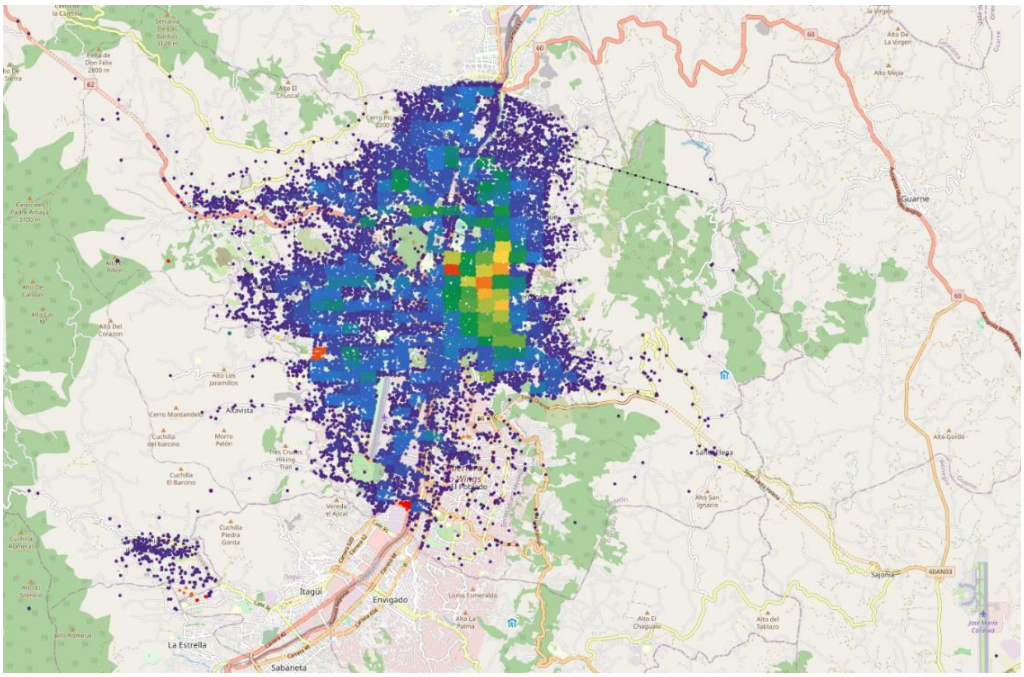


Figure 27: Agrupamiento por cuadrante K=50

100 grupos:

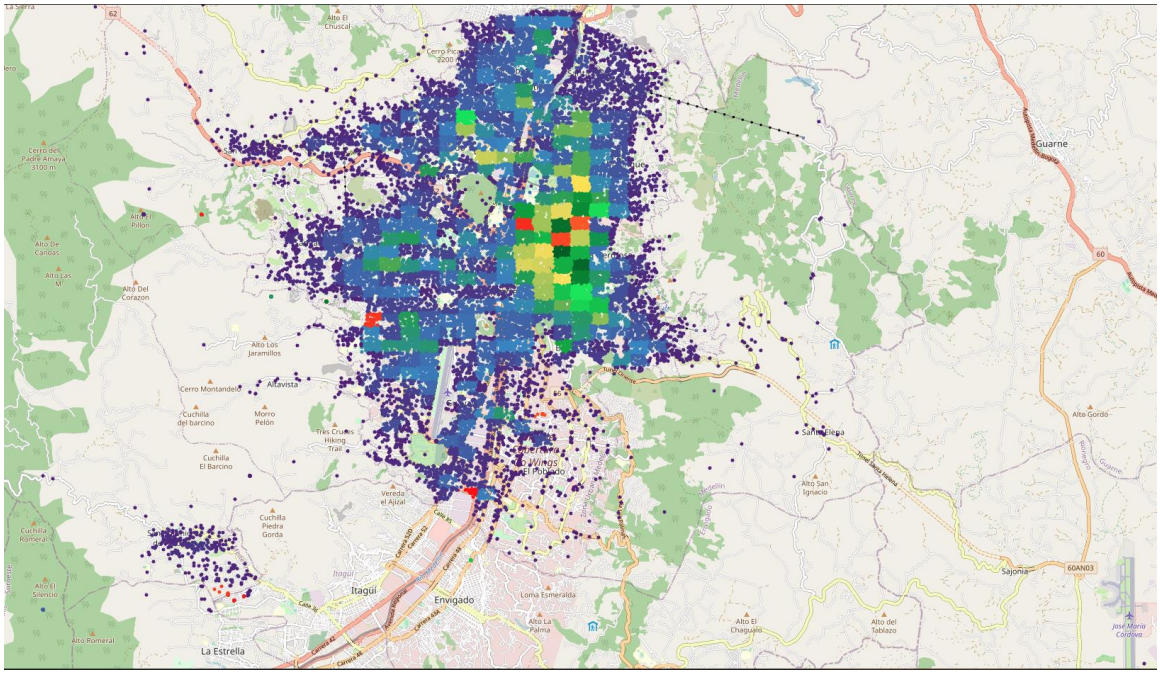


Figure 28: Agrupamiento por cuadrante con K=100

Se puede observar en ambas graficas que los puntos fríos se encuentran en la periferia de la ciudad y los puntos más calientes en el centro oriente de la ciudad con algunos puntos calientes en el sur en San Antonio de Prado, Puente de la Aguacatala hacía Guayabal y la Carrera 80 con la Calle 33 por el sector de la Villa de Aburrá y el sector rojo del centro es en el barrio Prado Centro en el sector conocido como El Chagualo. Ahora revisemos un mapa de calor generado por la librería Folium.

4.3. Regresión

El primer objetivo de este trabajo es la proyección de hurto de moto dada una localización geográfica y hora. Para el efecto se hace un entrenamiento de modelos lineales dividiendo la ciudad en subzonas geográficas, y ejecutando un modelo por cada una de estas subzonas o grupos de ellas, con el objetivo de poder hacer proyección de datos con dichos modelos.

Una vez realizado esta proyección, se realizará una proyección por barrio, ya que se quiere identificar como una zona más pequeña como lo son los barrios afectarían las precisiones del modelo.

4.3.1.Zonificación Geográfica de los datos.

Las variables de geoposicionamiento de la base de datos son a saber, latitud, longitud, barrio (nombre_barrio y código_barrio) además de la comuna (código comuna). El reto de la geostratificación es el de encontrar un nivel granular de zonas o grupos de zonas adecuado, de tal forma que no sean tan consolidadas como para que los modelos den un ajuste muy bajo por alta heterogeneidad, ni tan poco tan granulares que el ajuste sea más alto, pero tenga que usarse una cantidad muy grande de modelos. Para el efecto se estudiaron los siguientes tipos de división geográfica.

- ***Barrios:***
  Como primera aproximación se definió hacer modelos por cada uno de estos de acuerdo con el código consignado como barrio_nombre o su respectiva "dummización" que fue generada en la ingeniería de características.

- ***Segmentación:***
  Se hizo una segmentación de la ciudad en un marco de 60 por 60 cuadrículas y se ejecutaron procesos de clusterización sobre las mismas para 50 y 100 clústeres, con el objeto de no ejecutar 900 modelos, que es el número de cuadrantes, sino por el contrario ejecutar un modelo por cada clúster que constituyen un número menor. La clusterización se hizo, por tanto, sobre una segmentación del dataframe extrayendo las variables: cuadrante, año y cantidad, agrupadas en suma por cuadrante, año y después de hacer un proceso de pivoteo para tener como características de clusterización las columnas año, en la forma que se muestra en la siguiente imagen.

Table 4: Tabla Pivot para regresion

2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	2
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0	0	0	0	2	0	2	4	3	3	1	0	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0	0	0	0	3	1	1	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	2	3	0	2	5	9	3	3
0	0	0	0	2	2	0	0	0	1	1	8	2	6	8	1	3

4.3.2. Configuración de los modelos lineales

Una vez definidas las posibilidades de división geográfica, se procede a elaborar modelos lineales de la forma:

$$modeloLineal = cantidad \sim \text{fecha} + factor_1 + factor_2 + \cdots + factor_n$$

Donde la variable respuesta es cantidad, y los factores son variables o dummies de variables de tiempo como fecha, año, mes, día de la semana, número de la semana, si es festivo o día de quincena, entre otras.

A continuación, se puede apreciar la configuración del modelo lineal para la zona 124 (Palenque), donde se puede observar que el modelo se explica el 67% y la variable con mayor relevancia es el tiempo en el que se produjo el hurto:

```
   Modelo de regresión lineal para la zona No 194

R2adj=  0.6697673

Call:
lm(formula = formulalm, data = dffiltered)

Residuals:
     Min       1Q   Median       3Q      Max
-0.41271 -0.18723 -0.06428  0.14828  1.19791

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.765e+02  1.629e+01  10.835 3.71e-15 ***
tiempo         -8.711e-04  8.103e-05 -10.750 4.96e-15 ***
franja_horaria  2.433e-02  4.511e-02   0.539    0.592
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2797 on 54 degrees of freedom
Multiple R-squared:  0.6816,    Adjusted R-squared:  0.6698
F-statistic: 57.79 on 2 and 54 DF,  p-value: 3.817e-14

R2.Adj =  0.6697673       p-value según fórmula = 3.816789e-14
```

Figure 29: Summary Regresión lineal zona 124

4.3.3. Optimización del ajuste de los modelos lineales

4.3.3.1. Eliminación de residuales.

Para mejorar el ajuste de los modelos, se creó un procedimiento cíclico en el cual se calcula el modelo, se determina cual es el mayor residual y se elimina dicho evento de la base de datos, para luego correr de nuevo el modelo y repetir este procedimiento hasta que se elimine un número máximo predefinido de residuales que se configura como un porcentaje de la cantidad de datos, o se llegue a un R2 ajustado superior a un nivel también predeterminado.

```
        ####    Modelo de regresión lineal para el barrio o cuadrante No    194    ###

[1] "encontró un máximo"
Número de residuales desechados 1  de  4  que se esperaba desechar
Modelo de la forma cantidad ~ tiempo + franja_horaria
R2.Adj =  0.7833006       p-value según fórmula = 9.417351e-19
                  Estimate   Std. Error      t value     Pr(>|t|)
(Intercept)     1.905694e+02 1.335817e+01  14.26612779 8.658983e-20
tiempo         -9.407747e-04 6.642441e-05 -14.16308642 1.175639e-19
franja_horaria  1.472714e-03 3.654303e-02   0.04030082 9.680047e-01
```

Figure 30: Eliminación de residuales

4.3.3.2. Eliminación de los factores no significantes

Incluso si el modelo es significante con un p-valor por debajo el 5%, se pueden hallar diferentes coeficientes no significativos en el modelo. Para eliminar estas variables no significativas, se creó un procedimiento cíclico en el cual se calcula el modelo, se determina cual es el factor con mayor p-value por encima de 0.05 y se elimina, para luego correr de nuevo el modelo y repetir este procedimiento hasta que se eliminen los factores no significantes con p-value mayor a 5%.

```
Nuevo modelo de la forma cantidad ~ tiempo
R2.Adj =  0.787307       p-value según fórmula = 5.208898e-20
                Estimate   Std. Error   t value     Pr(>|t|)
(Intercept)  1.905520e+02 1.322725e+01  14.40602 3.831408e-20
tiempo      -9.406795e-04 6.576591e-05 -14.30345 5.208898e-20
```

Figure 31: Eliminación de variables no significantes

4.3.4. Guardado de los modelos

Una vez hecha la optimización de cada modelo, se guarda el mismo con una concatenación de la palabra *modelm_* y la identificación de la zona geográfica o el clúster, para que luego puedan ser utilizados para proyección. Un ejemplo de estos modelos y sus modificaciones tanto en eliminación de residuales y eliminación de factores significantes es el siguiente:

Table 5: Modelos de regresión

	resultszone	resultsname	resultsR2	resultspval	resultsfact
271	302	modelm_302	0.98468274	2.728598e-02	(Intercept),tiempo,franja_horaria
229	225	modelm_225	0.98033974	1.145731e-06	(Intercept),tiempo,franja_horaria
287	219	modelm_219	0.93692588	6.122976e-02	(Intercept),tiempo,franja_horaria
253	185	modelm_185	0.92004555	7.882677e-10	(Intercept),tiempo,franja_horaria
279	216	modelm_216	0.91017918	2.728598e-02	(Intercept),tiempo,franja_horaria
293	237	modelm_237	0.85375676	6.122976e-02	(Intercept),tiempo,franja_horaria
242	204	modelm_204	0.85352149	7.273027e-09	(Intercept),tiempo,franja_horaria
245	262	modelm_262	0.82880847	1.718704e-03	(Intercept),tiempo,franja_horaria
277	213	modelm_213	0.81151427	2.728598e-02	(Intercept),tiempo,franja_horaria
103	228	modelm_228	0.79481492	2.677883e-17	(Intercept),tiempo
266	271	modelm_271	0.79415833	2.329641e-05	(Intercept),tiempo,franja_horaria
302	285	modelm_285	0.78995452	6.122976e-02	(Intercept),tiempo,franja_horaria
198	194	modelm_194	0.78730703	5.208898e-20	(Intercept),tiempo
248	242	modelm_242	0.76796291	5.196842e-05	(Intercept),tiempo,franja_horaria
300	257	modelm_257	0.76687493	6.122976e-02	(Intercept),tiempo,franja_horaria
258	183	modelm_183	0.74970930	1.234852e-29	(Intercept),tiempo
61	154	modelm_154	0.73393339	4.654708e-24	(Intercept),tiempo
280	303	modelm_303	0.72251536	2.728598e-02	(Intercept),tiempo,franja_horaria
301	280	modelm_280	0.71984382	6.122976e-02	(Intercept),tiempo,franja_horaria
6	141	modelm_141	0.71431723	4.103961e-20	(Intercept),tiempo
259	221	modelm_221	0.70656761	1.451832e-17	(Intercept),tiempo

4.3.5. Prueba de los modelos

Con la base de datos de prueba, se ejecutó una predicción de las cantidades de hurtos por cada zona y se calcularon los residuales respecto a los valores reales, agrupando todos los residuales en un solo vector, para calcular finalmente el error cuadrático medio del modelo.

Con el fin de realizar la medición de la regresión, se toma como data de prueba el año 2019. Este se tomará 2019 como *Y* y las predicciones *Y-hat*

Table 6: Prueba de los modelos

	resultestzone	resultesttime	resultest1sthfactor	resultestnumobserv	resultestzonemse	resultestmsetotal	resultestYreal	resultestYhat	resultestYhatIndx
1	0	201903	0	15	0.47967942	0.006324175	1	0.7360708	0.1698423
2	0	201906	0	15	0.47967942	0.006324175	1	0.7331737	0.1691738
3	0	201907	0	15	0.47967942	0.006324175	1	0.7322080	0.1689510
813	0	201901	1	15	0.47967942	0.006324175	1	0.7380022	0.1702880
814	0	201903	1	15	0.47967942	0.006324175	1	0.7360708	0.1698423
815	0	201908	1	15	0.47967942	0.006324175	1	0.7312423	0.1687282
816	0	201910	1	15	0.47967942	0.006324175	1	0.7293109	0.1682825
1475	0	201902	2	15	0.47967942	0.006324175	2	0.7370365	0.1700652
1476	0	201903	2	15	0.47967942	0.006324175	1	0.7360708	0.1698423
1477	0	201904	2	15	0.47967942	0.006324175	1	0.7351051	0.1696195
1478	0	201905	2	15	0.47967942	0.006324175	2	0.7341394	0.1693967
1479	0	201906	2	15	0.47967942	0.006324175	1	0.7331737	0.1691738
1480	0	201907	2	15	0.47967942	0.006324175	2	0.7322080	0.1689510
1481	0	201908	2	15	0.47967942	0.006324175	2	0.7312423	0.1687282
1482	0	201910	2	15	0.47967942	0.006324175	1	0.7293109	0.1682825
356	83	201901	0	24	3.02156279	0.006324175	1	1.4480996	0.3341372
357	83	201904	0	24	3.02156279	0.006324175	3	1.4469232	0.3338657
358	83	201906	0	24	3.02156279	0.006324175	1	1.4461389	0.3336847
359	83	201907	0	24	3.02156279	0.006324175	2	1.4457468	0.3335943

4.3.6.Alternativas estudiadas y análisis de resultados

Dentro de la búsqueda de las alternativas estudiadas para encontrar el mejor conjunto de modelos predictores, se pretende hallar la que presente menor error cuadrático medio. Se exploraron por tanto modelos ejecutados clasificados por barrio y por los clústeres de cuadrante denominados CLUSTER50 (50 clústeres de cuadrante) y CLUSTER100 (100 clústeres de cuadrante). También, se estudió que ocurre si se disminuye el tamaño de la base de datos de entrenamiento acotándola por ejemplo a que comience en una fecha no tan lejana como enero 01 de 2009 que es como se tiene en los datos, sino por ejemplo con enero 01 de 2011, que es cuando termina el incremento drástico en hurtos.

Los resultados de estas alternativas y su respectivo error cuadrático medio se muestran a continuación.

Table 7: Resultados MSE

Iteración	*MSE*
Por Barrio (nombre_barrio)	0.0063
Con la identificación de los 50 grupos. (CLUSTER50)	0.8809
Con la identificación de los 100 grupos. (CLUSTER100)	0.1994

Las tres iteraciones se generaron con la siguiente formula Lm = AñoMes ~ Franja_horaria

4.3.7.Alternativas estudiadas y análisis de resultados

Con el fin de mostrar un indicador que muestre el comportamiento de hurtos que se tiene por barrio, Año, mes y franja horaria. El indicador se calcula como la relación entre el valor estimado por la regresión de la cantidad de hurtos sobre el valor de la mayor estimación de hurtos dada una georeferenciación (barrio), una determinación de tiempo (añomes) y una franja horaria (mañana/tarde/noche). Es decir, el estimador es una relación de cuanto es más probable que ocurra un hurto en el tiempo y lugar indicado comparado con el lugar y tiempo de mayor incidencia.

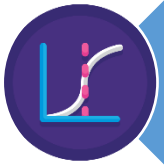
Ejemplo Barrio **Boston**:

Mañana **Tarde** **Noche**

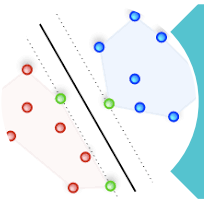
4.4. Clasificación.

4.4.1. Competencia de clasificadores

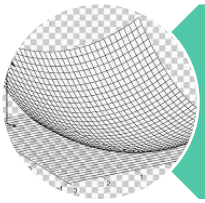
Buscando la resolución a partir de la información histórica de los hurtos de motos, en donde se comprende información de donde y cuando fueron robadas las motos e información demográfica, se realiza una serie de corridas de los siguientes algoritmos:



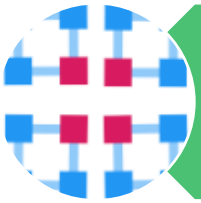
Regresión Logistica: Tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica



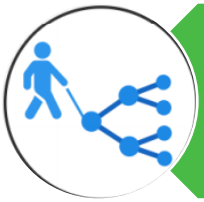
SVM: es un modelo de aprendizaje de máquina supervisada que utiliza algoritmos de clasificación para problemas de clasificación de dos grupos. Después de dar a un modelo SVM conjuntos de datos de entrenamiento etiquetados para cada categoría, son capaces de categorizar el nuevo texto. Así que estás trabajando en un problema de clasificación de texto.



Stochastic Gradient Descent: es un método iterativo para optimizar una función objetiva con propiedades de suavidad adecuadas, Puede considerarse una aproximación estocástica de la optimización del descenso de gradiente, ya que sustituye el gradiente real }



KNN: Estima la probabilidad posterior de que un elemento X pertenezca a la clase C a partir de la información obtenida en la elaboración de la distancia entro los K puntos

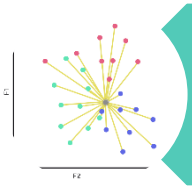


Naive Bayes: Es una aplicación simplificada de la estadistica bayesiana que parte de unos valores iniciales de confianza en una clasificación durante el set de entrenamiento asignado. Se asume que una caracteristica en un conjunto de datos no esta relacionada con la presencia de otra caracteristica

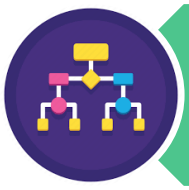
Decision Tree: Algoritmo supervisado, que nos ofrece una estructura en forma de arbol con soluciones posibles y asi ayuda a tomar una decision basada n ciertas condiciones

Extra Tree Classifier: Extra Trees utiliza toda la muestra original; su algoritmo ahorra tiempo porque todo el procedimiento es el mismo, pero elige al azar el punto de separación y no calcula el óptimo.

Bagging C Decision Tree:se utiliza cuando nuestro objetivo es reducir la varianza de un árbol de decisión. que combina varios árboles de decisión para producir un mejor rendimiento predictivo que la utilización de un solo árbol de decisión.

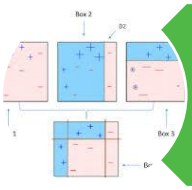


Bagging C KNN: consiste esencialmente en promediar sobre una (n infinita) serie de realizaciones de un mismo predictor base - en nuestro caso, k-NN. Las virtudes y limitaciones de los métodos basados en la combinación de predictores

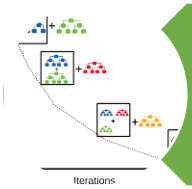


Bagging C Extra Tree: Aleatorizar fuertemente tanto la elección de atributos como la del punto de corte mientras se divide un nodo del árbol. En el caso extremo, construye árboles totalmente aleatorios cuyas estructuras son independientes de los valores de salida de la muestra de aprendizaje.

Random Forest: Cominación de arboles predictores que dependen de los valores de un vector aleatorio ndependeinte con la misma distribución para cada arbol. Contruye una larga colección de arbolesno correlacionados y luego se promedian.



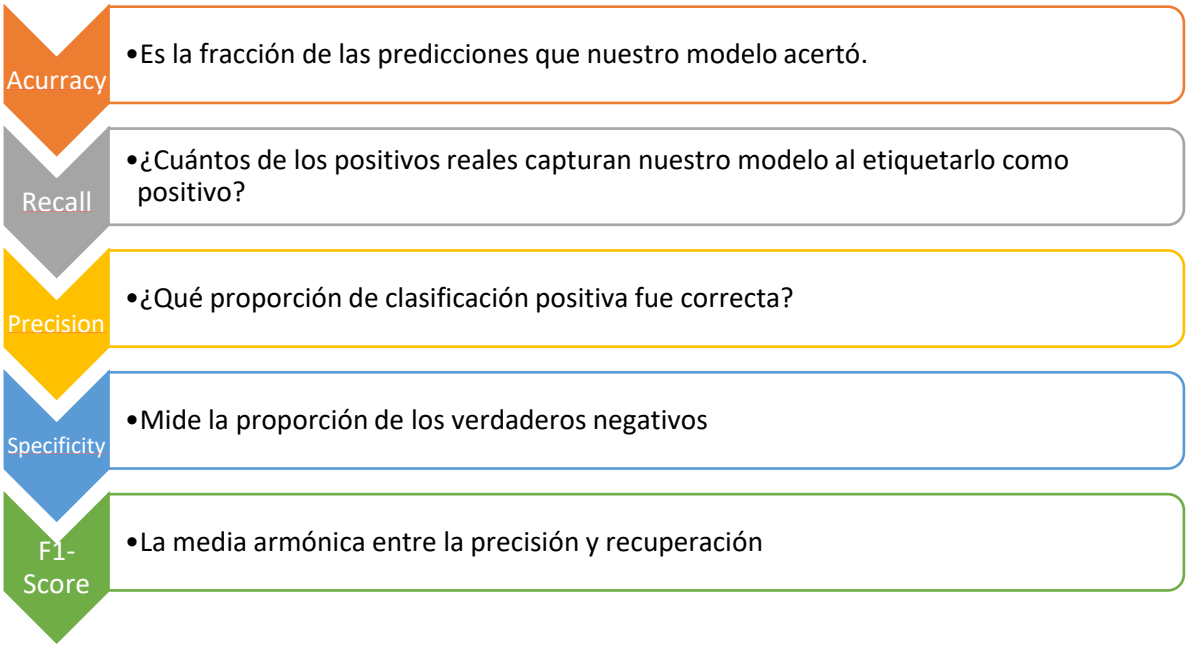
AdaBoost: que fue creado inicialmente para aumentar la eficiencia de los clasificadores binarios. AdaBoost utiliza un enfoque iterativo para aprender de los errores de los clasificadores débiles, y convertirlos en fuertes.



Gradient Tree Boosting: Construye el modelo por etapas como lo hacen otros métodos de potenciación, y los generaliza permitiendo la optimización de una función de pérdida diferenciable arbitraria.

4.4.2.Medición de clasificadores:

Con el fin de conocer cual clasificador debe tomarse para la clasificación de los hurtos de motos de acuerdo a la modalidad, se tendrá en cuenta las siguientes mediciones:



4.4.3.Iteraciones de los clasificadores

En la presente sección se implementación tres iteraciones para realizar la clasificación de la modalidad de hurtos (atraco o halado) en la ciudad de Medellín de la siguiente manera:

Primera iteración.

Se eligieron 21 características del Dataset después del proceso de agregación de nuevas variables:

- Latitud
- Longitud
- Estado_civil
- Medio_transporte
- Nombre_barrio
- Codigo_comuna
- Lugar
- Sede_receptora
- Modelo
- Fecha
- Anho
- Mes
- Día
- Dia_semana
- Hora
- Festivos
- Quincena
- Week_number
- Ferias_fiestas
- Franja_horaria
- Modalidad

Posteriormente se implementó un pipeline de clasificadores donde utilizamos el tipo de validación tradicional hold out separando los datos de entrenamiento y prueba en una proporción 80 – 20.

A continuación, se visualizan los rendimientos para cada clasificador implementado:

Table 8: Primera iteración de clasificación

Clasificador	Acurracy	Recall	Precision	Specificity	F1
Regresión logistica	0.568	0.000	0.000	1.000	0.000
SVM	0.568	0.000	0.000	1.000	0.000
Stochastic Gradient Descent	0.432	1.000	0.432	0.000	0.603
KNN	0.629	0.569	0.570	0.674	0.569
Naive Bayes	0.628	0.682	0.557	0.588	0.613
Decision Tree	0.864	0.846	0.841	0.878	0.844
ExtraTreeClassifier	0.805	0.788	0.767	0.818	0.777
Bagging C Decision Tree	0.906	0.877	0.903	0.928	0.890
Bagging C KNN	0.624	0.557	0.565	0.675	0.561
Bagging C Extra Tree	0.895	0.860	0.893	0.922	0.876
RandomFores	0.914	0.907	0.895	0.920	0.901
AdaBoost	0.882	0.890	0.845	0.876	0.867
Gradient Tree Boosting	0.889	0.888	0.860	0.890	0.874

El mejor clasificador en esta iteración fue el random forest, ya que se muestra que dentro de las diferentes mediciones fue el que mejor se desempeñó.

Segunda iteración.

En esta iteración se escogieron las variables no correlacionadas y seleccionadas en el proceso de ingeniería de características dando como resultado las siguientes 11 variables:

- Latitud
- Longitud
- estado_civil
- medio_transporte
- nombre_barrio
- codigo_comuna
- lugar
- sede_receptora
- modelo
- fecha
- modalidad

A continuación, se visualizan los rendimientos para cada clasificador:

Table 9: Iteración 2 clasificación

Clasificador	Acurracy	Recall	Precision	Specificity	F1
Regresión logistica	0.568	0.000	0.000	1.000	0.000
SVM	0.568	0.000	0.000	1.000	0.000
Stochastic Gradient Descent	0.568	0.000	0.000	1.000	0.000
KNN	0.622	0.576	0.560	0.656	0.568
Naive Bayes	0.612	0.664	0.542	0.573	0.597
Decision Tree	0.866	0.852	0.839	0.876	0.846
ExtraTreeClassifier	0.853	0.838	0.825	0.865	0.831
Bagging C Decision Tree	0.896	0.870	0.886	0.915	0.878
Bagging C KNN	0.616	0.554	0.556	0.664	0.555
Bagging C Extra Tree	0.893	0.866	0.883	0.913	0.874
RandomFores	0.908	0.901	0.888	0.913	0.894
AdaBoost	0.879	0.896	0.836	0.866	0.865
Gradient Tree Boosting	0.884	0.891	0.848	0.879	0.869

El mejor clasificador en esta iteración fue el random forest, ya que, dentro de la iteración con mejores variables, este clasificador continúa teniendo mejor desempeño.

Tercera iteración.

En esta iteración se escogieron las mismas variables indicadas en la iteración numero 2 buscando los mejores hiper parámetros de los clasificadores implementados en el pipeline.

A continuación, se visualizan los rendimientos para cada clasificador:

Table 10: Iteración 3 Clasificación

Clasificador	Acurracy	Recall	Precision	Specificity	F1
Regression logistical	0.657	0.528	0.620	0.754	0.570
Stochastic Gradient Descent	0.568	0.000	0.000	1.000	0.000
KNN	0.718	0.687	0.669	0.742	0.678
Naive Bayes	0.612	0.664	0.542	0.573	0.597
Decision Tree	0.883	0.879	0.855	0.887	0.867
ExtraTreeClassifier	0.837	0.831	0.799	0.841	0.815
Bagging C Decision Tree	0.888	0.889	0.856	0.886	0.872
Bagging C KNN	0.717	0.683	0.669	0.743	0.676
Bagging C Extra Tree	0.882	0.893	0.843	0.874	0.867
RandomForest	0.909	0.904	0.887	0.912	0.895
AdaBoost	0.882	0.894	0.842	0.872	0.867

El mejor clasificador en esta iteración fue el random forest, ya que una vez tomados los hipermetros se muestra que es el mejor clasificador.

4.4.4.Resultado de la clasificación

Como se ve a continuación el random forest tiene las mejores medidas y adicional su comportamiento dentro de las iteraciones:

- Acurracy:
  Exactitud de clasificación entre las predicciones correctas para la modalidad de hurto y el número total de predicciones.

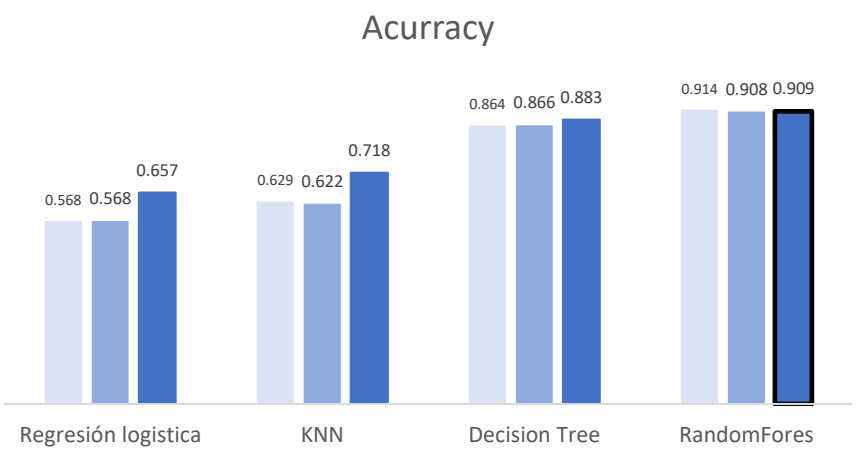


Figure 32: Acurracy de los clasificadores

- Precisión:
  Precisión de clasificación entre las predicciones correctas y el número total de predicciones correctas previstas. Casos positivos.

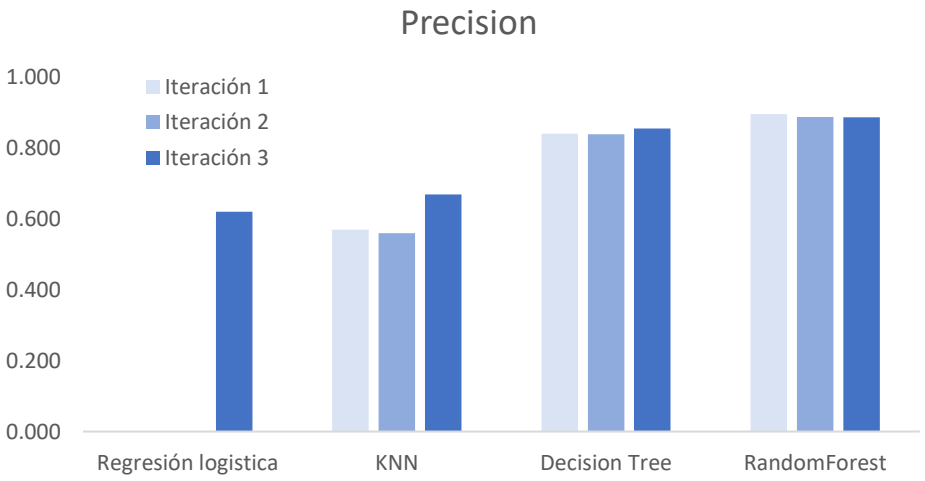


Figure 33: Precision de los clasificadores más representativos

- F1-Score:
  Media armónica de la memoria y la precisión

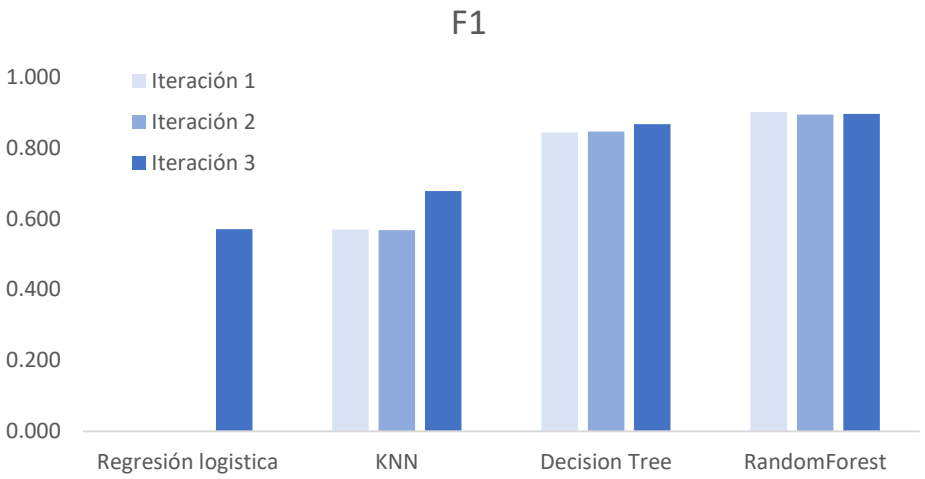


Figure 34: F1-score de los clasificadores más representativos

Evidenciamos en cada una de las iteraciones, el clasificador con mejores métricas para predecir la modalidad de hurto (halado o atraco) de motos en la ciudad de Medellín y área metropolitana fue random forest obteniendo las siguientes métricas:

Table 11: Resumen Random Forest

Iteración	ACURRACY	RECALL	PRECISION	SPECIFICITY	F1
1	0.914	0.907	0.895	0.920	0.901
2	0.908	0.901	0.888	0.913	0.894
3	0.909	0.904	0.887	0.912	0.895

5. Conclusiones:

1. La regresión lineal que se ejecutó por barrios, se encontraron 326 Barrios; donde se observa que 40 Barrios tiene un R^2 mayor al 65%, 53 por encima del 60%, 62 del 55% y 80 del 50%.
2. Dado lo estocástico del proceso para el modelo del hurto de motos, solo se encontró modelos con un R^2 apropiado sobre un 20% de los barrios. Lo que indica que existe alta varianza en la ocurrencia de hurtos a nivel de las regiones geográficas seleccionadas.
3. Sería conveniente utilizar un modelo lineal generalizado con distribución de Poisson, dado que la variable de conteo es discreta y sus residuales no se comportan de forma normal.
4. Se intento utilizar la variable tiempo un nivel de granularidad de día, mes y año, sin embargo, existió un decremento importante en la métrica de ajuste, por lo que se decidió tomar una variable mas agrupada como lo fue año y mes.
5. Como se observó en los resultados de la regresión, a medida que aumenta los grupos (clusters), se disminuye el error cuadrático medio, aunque este genera un número mayor de modelos. Por esta razón se tomó la decisión de realizar la regresión por barrio, debido a que su error cuadrático medio es el menor de todos.

6. Bibliografía

1. Microsoft Azure (2020). What is the Team Data Science Process? Recuperado de https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview
2. Análisis de hurtos. Orchard M. (2014). MODELACIÓN Y PREDICCIÓN DE FOCOS DE CRIMINALIDAD BASADO EN MODELOS PROBABILÍSTICOS, Universidad Chile, Facultad de ciencias físicas y matemática
3. Park, H.S. and Jun, C.H., 2009. A simple and fast algorithm for K-medoids clustering. Expert systems with applications, 36(2), pp.3336-3341.
4. Supervised learning — scikit-learn 0.23.2 documentation. (n.d.). Retrieved December 4, 2020, from https://scikit-learn.org/stable/supervised_learning.html.
5. Wilson R., Filbert K. (2008) Crime Mapping and Analysis. In: Shekhar S., Xiong H. (eds) Encyclopedia of GIS. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-35973-1_226
6. Ahamed Kameel Meera (2006). Determinants of crime in a developing country: a regression model. College of Business Administrator, University of North Texas.
7. Osgood, D.W (2000). Poisson-Based Regression Analysis of Aggregate Crime Rates. Journal of Quantitative Criminology 16, 21–43
8. Al Amin Biswas, Sarnali Basak (2019). Forecasting the Trends and Patterns of Crime in Bangladesh using Machine Learning Model, Intelligent Communication and Computational Techniques.