Informe Ciencia de Datos "Ecobicis-CABA"

Leandro Prchal y Luka Herman

UTN-FRBA, Ciudad Autónoma de Buenos Aires, Argentina.

Abstracto

El trabajo consistió en dos partes. En la primera parte hemos realizado un análisis exploratorio de los datos del dataset con el fin de entenderlo y comprenderlo mejor. En la segunda parte hemos realizado machine learning, utilizando un SVM y un KNN, para predecir el sexo del usuario a partir de la edad y el tiempo de uso.

Palabras Clave

Estaciones, bicicletas, tiempo de uso y edad.

Introducción

Queremos realizar un análisis que nos permita, mediante el tiempo de uso de la bicicleta y la edad del usuario nos predecir el sexo del usuario. Para esto realizamos un aprendizaje supervisado de clasificación, evaluando los resultados obtenidos mediante los métodos SVM y KNN.

Data set

Utilizamos dos dataset: El dataset principal utilizado fue "Recorridos-realizados-2018.csv" donde pudimos observar la cantidad de viajes realizados durante el año 2018 y fue complementado con "Estaciones-bicicletas.csv" donde pudimos ver la cantidad de estaciones y a que barrio pertenecen.

Ambos dataset fueron provistos por el gobierno de la Ciudad Autónoma de Buenos Aires en su página. Los podemos encontrar en:

https://data.buenosaires.gob.ar/dataset/bicicletas-publicas

https://data.buenosaires.gob.ar/dataset/estaciones-ecobici

Desarrollo

Lo primero que realizamos fue un EDA (Exploratory Data Analysis) del dataset para saber con qué tipo de dataset nos encontramos y que información podemos sacar de él. Era un dataset con muchos datos, ya que cuenta 2.619.968 de samples y 9 features. La información que nos proveía el dataset era el ID del usuario, la fecha de retiro, el tiempo de uso, nombre de la estación de origen y la de destino, el sexo del usuario y la edad del usuario. Luego realizamos una limpieza de datos para sacar los NaNs y nulls, lo que redujo nuestro dataset a 2.576.245. Por lo que se eliminaron 43.723 samples.

Al realizar un ".describe()" pudimos observar que la media de edad es de 33,24 años. Nos llamó la atención que el usuario mas grande tenia 140 años mientras que el mas chico tiene 16 años, lo que suena lógico ya que es el mínimo exigido para registrarse como usuario.

Luego nos topamos con nuestro primer inconveniente. Los datos de tiempo de uso estaban en formato string, dándonos la cantidad de días utilizadas, horas, minutos, segundos y microsegundos y no podíamos trabajar con los mismos. Lo que hicimos fue pasarlo a formato numérico para poder trabajarlo, separamos los números y buscamos quedarnos con las horas y los minutos, creamos una

nueva feature donde las horas las multiplicabamos por 60 minutos y le sumamos la cantidad de minutos. La media de uso es de 25,27 minutos. Luego mediante un “join” juntamos la columna a nuestro dataset original y eliminamos la columna original.

El dataset nos arrojó que, de la cantidad total de usuarios, el 71,817% son usuarios masculinos, el 28,179% son femeninos mientras que el 0,00264% son sin sexo definido.

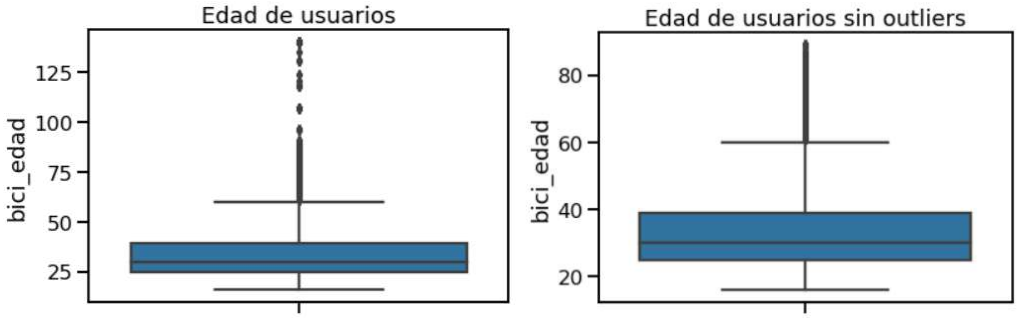
Para analizar mejor el dataset, agrupamos por el tiempo de uso. Agrupamos por corta duración (menor a 15 minutos), media duración (entre 15 y 30 minutos) y larga duración (mayor a 30 minutos). También agrupamos por rango de edad. Definimos como menor de edad (menor a 18 años), adulto joven (entre 18 y 35 años), adulto mayor (entre 35 y 65 años) y jubilados (mayor a 65 años). Nos llamó la atención que los menores de edad tienen un tiempo promedio de uso mayor al adulto joven y adulto mayor. Y los jubilados son los que poseen mayor tiempo de uso.

	GrupoEdad	Tiempo_uso
0	Adulto Joven	24.949375
1	Adulto Mayor	25.598123
2	Jubilados	31.990170
3	Menor de Edad	27.434818

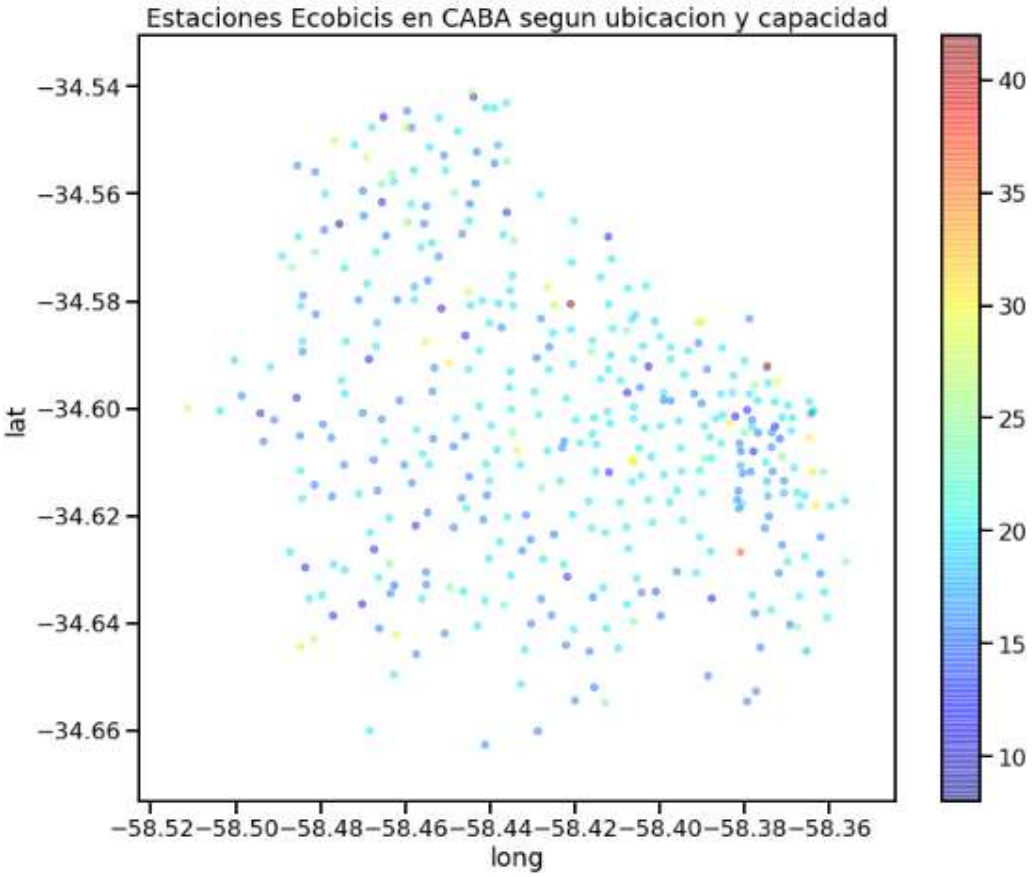
También realizamos un gráfico de tortas para conocer el porcentaje de la cantidad usuarios por banda etarias. La cual es la siguiente:

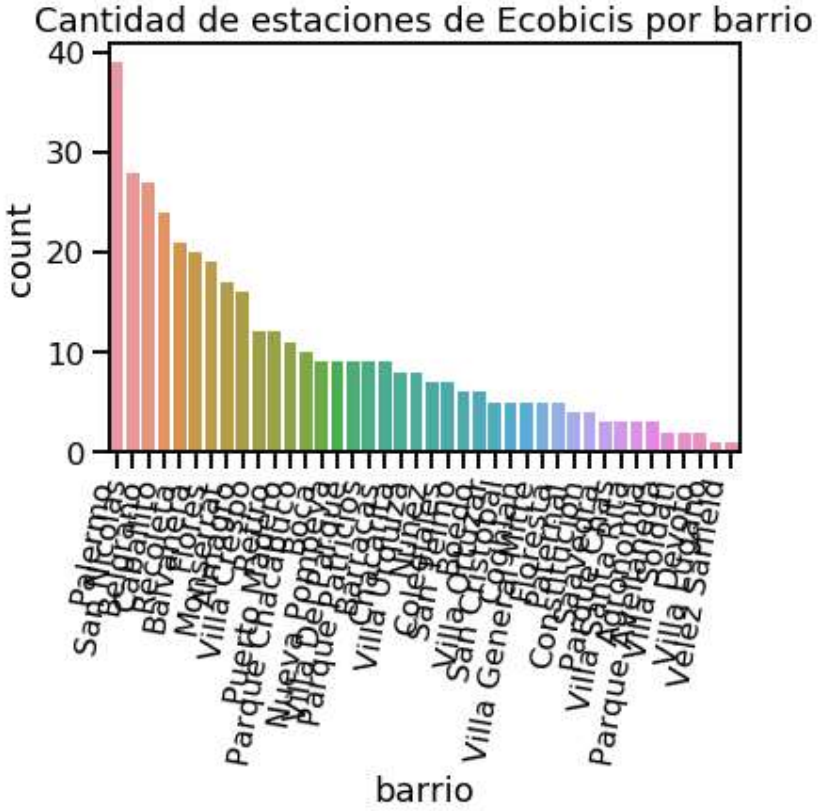
Realizamos un análisis por la hora de retiro de la bicicleta. Para ellos utilizamos un histograma de la hora a la que se retira la bicicleta y observamos que la hora pico es a las 17 y 18 hs mientras que el horario menos usamos es a las 3 y 4 de la mañana.

Luego corrimos un boxplot para la edad de los usuarios y otro eliminando los outliers, ya que entendemos que la gente mayor a 90 años, no usa una bicicleta publica arrendada vía la aplicación de un teléfono celular, sino que son usuarios que fueron mal cargados en el sistema ya sea por un error del usuario o del sistema mismo. Elimino un total de 107 samples. Los Boxplot obtenidos son los siguientes:



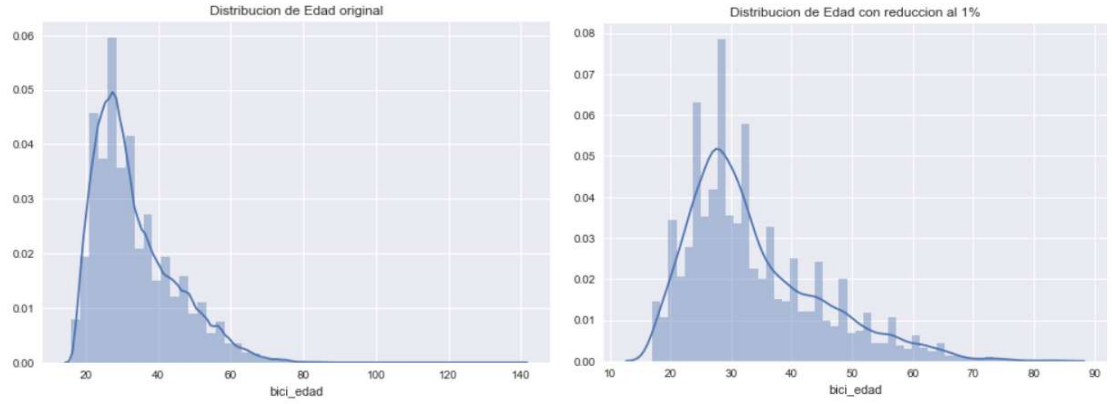
Buscamos las estaciones más usadas, a las que denominamos Top15. Las mismas eran Facultad de Medicina, Pacifico, Parque Las Heras, Retiro, entre otras. Y luego usamos el segundo dataset “Estaciones – Bicicletas” y analizamos que los barrios con mayor cantidad de estaciones son Palermo, San Nicolas, Belgrano, Caballito y Recoleta. Realizamos un “scater plot” sobre la ubicación de estas según su longitud y latitud, y su capacidad. Y realizamos un “countplot” sobre la cantidad de estaciones en cada barrio. Obtuvimos los siguientes mapas:


Estaciones Ecobicis en CABA segun ubicacion y capacidad
lat
long
−34.54
−34.56
−34.58
−34.60
−34.62
−34.64
−34.66
−58.52 −58.50 −58.48 −58.46 −58.44 −58.42 −58.40 −58.38 −58.36
40
35
30
25
20
15
10


Cantidad de estaciones de Ecobicis por barrio
count
barrio
40
30
20
10
0

Ahora llegamos a la segunda parte del trabajo practico. Decidimos utilizar para Machine Learning, un aprendizaje supervisado por clasificación. Vamos a realizar una clasificación por Support Vector Machine (SVM) y otra por K-Nearest Neighbor (KNN). Dado que nuestra muestra es muy grande, y contienen mas de 2 millones de samples se nos hacía imposible correr un modelo porque tardaba demasiado.

Para resolver este problema, achicamos nuestra muestra al 1%. Pero necesitamos asegurarnos de que este 1% sea representativo de la misma. Para eso realizamos un “displot“ de la muestra original y de la muestra del 1%. Además, comparamos la media de las muestras: 33,24 años (distribución original) y 33,72 años (distribución 1%), y el desvío estándar: 11,12 (distribución original) y 10, 99 (distribución 1%).



Para realizar el SMV, utilizamos un Grid Search CV con un Cross Validation = 3. Además, utilizamos dos parámetros para el Kernel, dos para C y dos para gamma.

```
params_svm = {'kernel':['linear', 'rbf'], 'C':[0.1 , 1], 'gamma':[0.01, 0.1]}
svc = svm.SVC(probability=True)
svm_cv = GridSearchCV(svc, param_grid = params_svm, refit = True ,cv = 3)
```

Lo que nos dio que los mejores parámetros y el accuracy son:

```
Los mejores parametros son {'C': 0.1, 'gamma': 0.01, 'kernel': 'linear'} con resultado de 0.73

El accuracy train con SVM es 0.7252259746021184
El accuracy test con SVM es 0.7299780049165481
```

Y por último realizamos una matriz de confusión, la que obtuvimos la siguiente:

Para realizar el KNN, decidimos que el hiper-parametro la cantidad de vecinos sea igual a 5. También utilizamos el 1% de la muestra representativa. Para el cual obtuvimos un accuracy de:

```
El accuracy es 0.6762841247250615
```

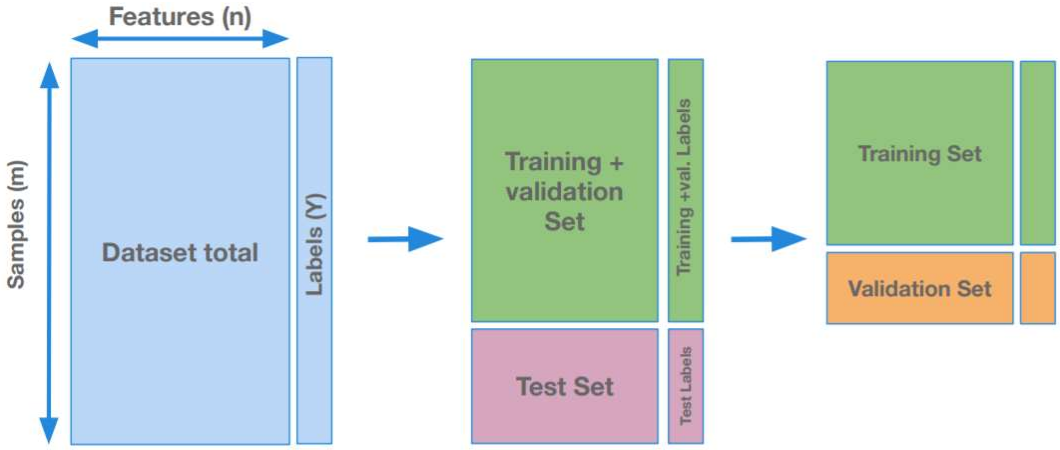
Mientras que la matriz de confusión para el KNN nos dio:

Background Teórico

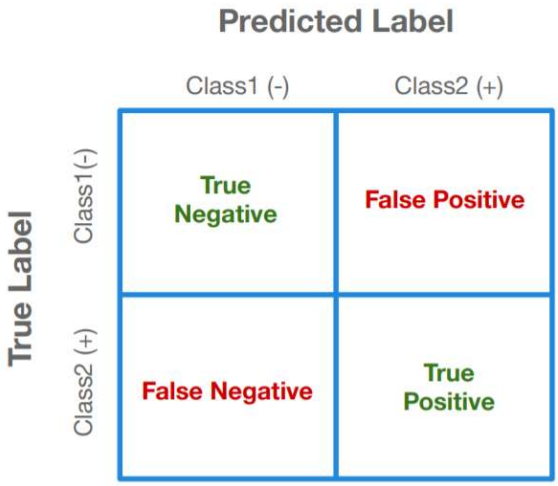
Para realizar el auto scaling trabaja asumiendo que cada feature responde a una distribución normal y estandariza los valores afectándolos por la media y el desvío estándar.

$$x_i' = \frac{(x_i - \mu)}{\sigma}$$

Train, Validation, Test sets: El clasificador aprenderá la regla usando el train set. Luego clasificará las muestras del test y se medirá la exactitud.



La matriz de confusión trabaja de la siguiente forma:



KNN: Clasifica cada nuevo dato en el grupo que corresponda, según tenga K vecinos más cerca de un grupo o del otro. Ordena las distancias para saber a qué grupo pertenece. El hiper-parametro son los K vecinos.

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Conclusiones

El SVM con el GridSearch CV predice mejor el sexo a partir del tiempo de uso y la edad que el KNN. La diferencia en el accuaracy es de 5,36%. El mejor acurracy es de 73%, por lo que podemos concluir que es un muy buen valor alcanzado.

Referencias

Apuntes cursada Cluster AI-UTN FRBA 2019

Python data science handbook - Jake Van der Plas

https://medium.com/machine-learning-101/k-nearest-neighbors-classifier-1c1ff404d265

https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

Reconocimiento

Queríamos reconocer especialmente a nuestro mentor, Hernan Joaquin Magallanes, quien nos ha brindado ayuda y soporte durante toda la cursa, no solamente para la realización de este trabajo practico, sino también para comprender mejor la materia y poder expandir nuestros conocimientos.