UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Master Big Data Analytics

2023-2024

¿Qué es DATA SCIENCE?

Memoria proyecto Valenbisi

Alumno:
José Carlos Ávila Palazón

Usuario Kaggle:

IntroBI_joavpa

Tabla de contenido

1. Introducción del proyecto

El proyecto que se desempeña en la asignatura de ¿Qué es Data Science? Es la predicción de cuantas bicis a una previsión de 3h habrán en las distintas estaciones.

En nuestro caso, nos presentan las estaciones 7, 8, 106 y 110 a predecir en los meses de junio y Julio de 2014.

Los datos proporcionados son 4 distintos ficheros:

1. Train file: datos de las estaciones 1,6,9,10,95,104,105,107,109,112,113 de distintos años
2. Deploy file: datos de las estaciones a predecir, pero de unos meses anteriores
3. Test file: datos utilizados para la predicción del modelo
4. Distancias estaciones: Distancias entre estaciones

2. Carga y Exploración de los Datos

Los datos que se pueden representar dentro de los ficheros de entrenamiento(train y deploy) proporcionados son los siguientes:

1. Datos de latitud y longitud de las estaciones
2. Datos temporales (fecha, año, mes, día...)
3. Datos meteorologicos (humedad, velocidad viento...)
4. Datos de cantidad de bikes(shor y full profile bikes ...)

```
Data columns (total 26 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   station                   198120 non-null  int64
 1   latitude                  198120 non-null  float64
 2   longitude                 198120 non-null  float64
 3   numDocks                  198120 non-null  int64
 4   timestamp                 198120 non-null  float64
 5   year                      198120 non-null  int64
 6   month                     198120 non-null  int64
 7   day                       198120 non-null  int64
 8   hour                      198120 non-null  int64
 9   weekday                   198120 non-null  object
 10  weekhour                  198120 non-null  int64
 11  isHoliday                 198120 non-null  int64
 12  windMaxSpeed.m.s          156064 non-null  float64
 13  windMeanSpeed.m.s         194364 non-null  float64
 14  windDirection.grades      164732 non-null  float64
 15  temperature.C             194380 non-null  float64
 16  relHumidity.HR            193196 non-null  float64
 17  airPressure.mb            194348 non-null  float64
 18  precipitation.l.m2        194524 non-null  float64
 19  bikes_3h_ago              198120 non-null  float64
...
 23  short_profile_bikes       1172 non-null    float64
 24  Id                        1172 non-null    int64
```

Podemos apreciar que tenemos valores faltantes que deberemos solucionar y variables categoricas que habran que codificar para la elaboracion del modelo.

Luego tenemos el fichero de distancas entre estaciones, el cual nos da informacion importante a la hora de saber mas de las estaciones.

	station	X1	X10	X104	X105	X106	X107	\
0	7	0.668155	0.515966	2.781240	3.026953	3.268591	3.665819	
1	8	0.923482	0.750714	2.435199	2.679080	2.914950	3.308108	
2	106	3.593521	3.606036	0.543021	0.313813	0.000000	0.453848	
3	110	3.968945	4.091112	0.940925	0.708904	0.597346	0.654877	

	X109	X110	X112	X113	X2	X3	X4	\
0	3.927129	3.715692	3.209306	3.148029	0.578547	0.257617	0.414689	
1	3.572673	3.372971	2.861704	2.811886	0.739454	0.541615	0.271538	
2	0.659364	0.597346	0.225769	0.531987	3.324009	3.367535	2.914433	
3	0.514739	0.000000	0.531186	0.590896	3.699700	3.786277	3.337544	

	X5	X6	X7	X8	X9	X95
0	0.734364	0.479204	0.000000	0.360870	0.798914	2.907578
1	0.442322	0.802950	0.360870	0.000000	0.440700	2.567349
2	2.569640	3.712353	3.268591	2.914950	2.474869	0.537658
3	2.993221	4.175886	3.715692	3.372971	2.939672	0.809740

Se puede ver a simple vista que las estaciones mas lejanas a las estaciones a predecir son 1, 95 y 104.
Esto nos da informacion relevante a la hora de posteriormente seleccionar los datos exactos a predecir, es decir, optar por eliminar los datos de esas estaciones o contar con ellos(no se opta por eliminarlos ya que la difernecia no es lo suficiente robusta como para quitarnos datos de entrenamiento útiles).

3. Limpieza de los datos

Los datos que nos proporcionan están realmente limpios, por lo tanto, en este apartado nos dedicaremos a imputar los valores faltantes que aparecen en nuestros datos y a unir train y deploy para entrenarlo en conjunto.

Imputaremos a las variables con valores faltantes que son

['windMaxSpeed.m.s', 'windMeanSpeed.m.s', 'windDirection.grades', 'temperature.C', 'relHumidity.HR', 'airPressure.mb', 'precipitation.l.m2']

Para la sustitución de estos valores atípicos se ha utilizado la mediana, ya que es una medida más robusta, tiene mayor representación del valor central de los datos, por ello se ha considerado más estable y consistente que la media.

4. Eliminar y Generar nuevas características

En este punto del proyecto, tenemos toda la información respecto a nuestras variables que utilizaremos para entrenar al modelo. Es importante tener en cuenta que existen características que no nos aportan información de valor, es por ello que se ha considerado eliminar las siguientes columnas:

1. Timestamp; Teniendo columnas que nos indican información más detallada de fecha, es una columna que se puede considerar eliminar para evitar tener sobreinformación.
2. Id: No aporta información, simplemente enumera los registros
3. Station: Saber las estaciones es irrelevante, ya que el modelo tiene que entrenar a unas en específico.
4. Precipitation.l.m2: esta columna si exploras sus datos gran parte de ellos son 0, sin contar los valores faltantes, no es útil para el entrenamiento.

   También, lo contrario a eliminar columnas, se ha realizado la técnica de derivar variables o dicho de otra forma ingeniería de características, que consiste en utilizar la información que se nos proporciona para generar nuevas columnas.

   Esto nos hace mejor el rendimiento del modelo y capturar relaciones más complejas entre las características originales y el objetivo a predecir.

Se han implementado las siguientes:

1. la variable **wind_humidity_interaction:** producto de **windMeanSpeed.m.s** y **relHumidity.HR** para capturar el efecto conjunto de la velocidad del viento y la humedad en la disponibilidad de bicicletas.

2. **day_period**: para dividir el día en segmentos significativos (como mañana, tarde, noche) puede ayudar a entender mejor cómo varía la demanda de bicicletas a lo largo del día.

3. **day_of_year**: Calcula el día del año a partir de la fecha.

5. Preparación de datos para el modelo

Tenemos toda la información que hemos creído conveniente tener para la predicción de las bicicletas en las distintas estaciones, no obstante, es importante a la hora de elaborar el modelo convertir las columnas categóricas, pasando a ser variables indicadoras o numéricas, depende de la técnica de encoding utilizada.

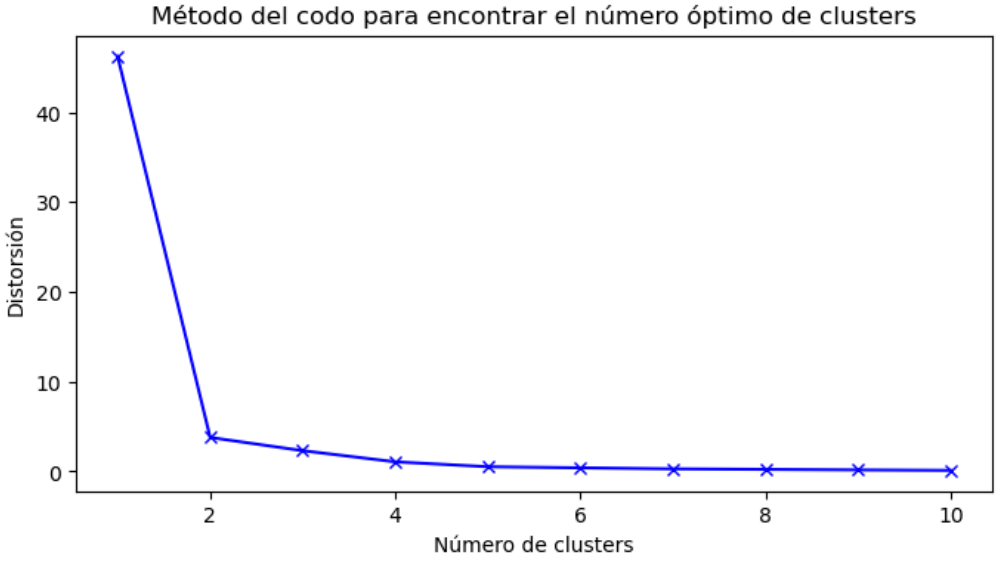
Utilizaremos la técnica “One-hot encoding “ la cual genera para cada tipo de valor categórico una nueva columna booleana.
Nuestras variables quedan de la siguiente forma:

```
Data columns (total 37 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   station                        129384 non-null  int64
 1   latitude                       129384 non-null  float64
 2   longitude                      129384 non-null  float64
 3   numDocks                       129384 non-null  int64
 4   timestamp                      129384 non-null  float64
 5   year                           129384 non-null  int64
 6   month                          129384 non-null  int64
 7   day                            129384 non-null  int64
 8   hour                           129384 non-null  int64
 9   weekhour                       129384 non-null  int64
 10  isHoliday                      129384 non-null  int64
 11  windMaxSpeed.m.s               129384 non-null  float64
 12  windMeanSpeed.m.s              129384 non-null  float64
 13  windDirection.grades           129384 non-null  float64
 14  temperature.C                  129384 non-null  float64
 15  relHumidity.HR                 129384 non-null  float64
 16  airPressure.mb                 129384 non-null  float64
 17  precipitation.l.m2             129384 non-null  float64
 18  bikes_3h_ago                   129384 non-null  float64
 19  full_profile_3h_diff_bikes     129384 non-null  float64
 20  full_profile_bikes             129384 non-null  float64
 21  short_profile_3h_diff_bikes    129384 non-null  float64
 22  short_profile_bikes            129384 non-null  float64
 23  bikes                          129384 non-null  float64
 24  Id                             129384 non-null  int64
 25  wind_humidity_interaction      129384 non-null  float64
 26  weekday_Friday                 129384 non-null  int64
 27  weekday_Monday                 129384 non-null  int64
 28  weekday_Saturday               129384 non-null  int64
 29  weekday_Sunday                 129384 non-null  int64
 30  weekday_Thursday               129384 non-null  int64
 31  weekday_Tuesday                129384 non-null  int64
 32  weekday_Wednesday              129384 non-null  int64
 33  day_period_Night               129384 non-null  int64
 34  day_period_Morning             129384 non-null  int64
 35  day_period_Afternoon           129384 non-null  int64
 36  day_period_Evening             129384 non-null  int64
```

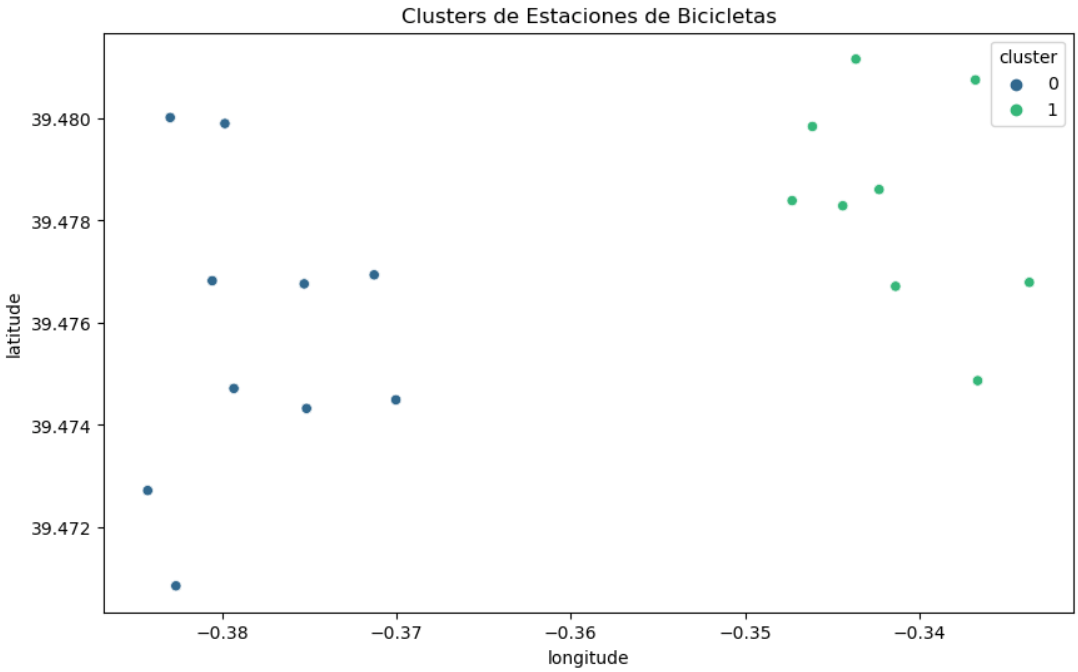
6. Clustering de Estaciones

Si revisamos latitud y longitud de las estaciones podemos obvervar que existen separaciones entre sí bastante evidentes, esto nos lleva a optar por hacer clustering para diferenciar entre las estaciones que se encuentran en el centro de Valencia (1-10) a las estaciones que se encuentran por las universidades (95-113)

Es bastante evidente que hay que utilizar dos clusters pero como confirmación vamos a realizar la teoria del codo:



Claramente se puede ver que son 2 clusters, ahora elaboramos clustering utilizando K-Means con el numero optimo de clusters ya verificado.



Se puede ver claramente la diferenciacion entre las dos agrupaciones y esto nos sirve de ayuda para realizar el modelo de regresion lineal.

7. Predicción

Para la predicción hemos realizado la prueba de dos modelos, una de regresión lineal y la segunda opción RandomForest, el cual nos ha proporcionado mejores resultados.

Se ha utilizado el fichero que se proporciona para el test y para el train los dos ficheros train y deploy.

Además, simplemente comentar que se ha realizado posteriormente a definir X e Y un escalado utilizando la función StandarScaler() para que ciertas características con valores mayores dominen el comportamiento del algoritmo.

8. Conclusiones

Para finalizar, la puntuación conseguida como usuario IntroBI_joavpa es de 3.35988, una puntuación mejorable pero no está mal.

Aspectos que he sacado como conclusión es que los datos el cómo manejarlos (limpieza, derivar y eliminar variables...) tiene una gran importancia a la hora de realizar el modelo y entender que herramientas usar en cualquier situación es lo más complejo de estos proyectos.

Y por último, quería sugerir feedback por tu parte, es decir, poder ver cosas mejorables (que seguro que serán muchas) y aspectos que pueden ayudarme a llevar este tipo de proyectos al siguiente nivel.