MINERÍA
PRÁCTICA 03

PABLO SIMÓN SAINZ

IVÁN RUIZ GÁZQUEZ

ÍNDICE

INTRODUCCIÓN

Para esta práctica haremos un modelo de clasificación de regresión, a parte de ver cómo debemos tratar con los outliers.

TRATAMIENTO DEL DATASET

Las columnas de los datos crudos que nos encontramos son los siguientes:

- **String** pickup_datetime: Fecha y hora de recogida.
- **Double** pickup_longitude, **Double** pickup_latitude: Coordenadas de la recogida.
- **Double** dropoff_longitude, **Double** dropoff_latitude: Coordenada y de la recogida.
- **Integer** passengers: cantidad de pasajeros a llevar.

TRANSFORMACIÓN DE DATOS

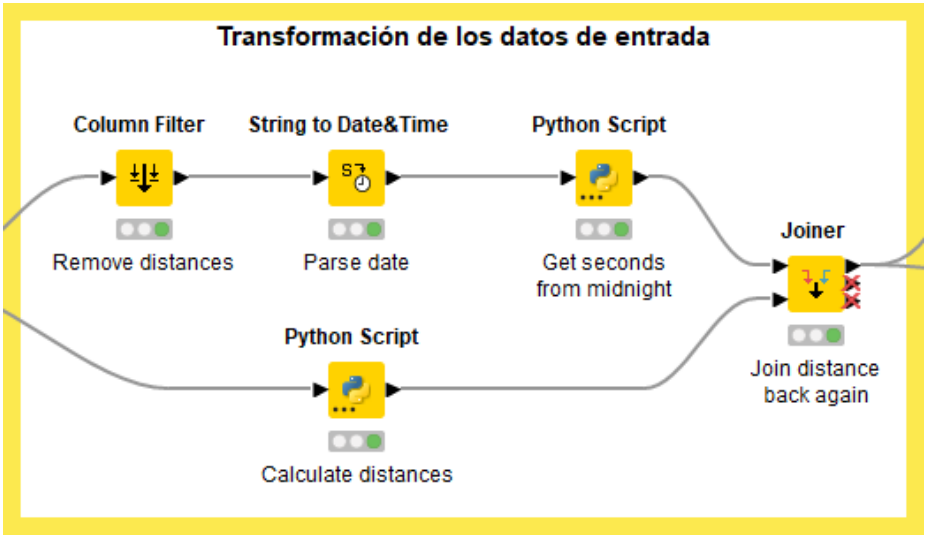
Dada la naturaleza del problema, hemos visto conveniente sustituir las 4 coordenadas dadas por la distancia que hay entre el punto de recogida y el de destino.

La distancia que obtenemos es en grados, y al no estar familiarizados con esa magnitud, hemos convertido a kilómetros, para así detectar más adelante los valores atípicos con mayor facilidad.

Para la obtención de la distancia, hemos decidido calcular la distancia en kilómetros entre dos coordenadas que pesamos se parece más a la distancia real que nos podemos encontrar en una situación de este tipo.

```
def get_distance_between_points(x, y):
    lat1, lon1 = x
    lat2, lon2 = y
    R = 6371.0 # Radius of the earth in km
    dlat = math.radians(lat2 - lat1)
    dlon = math.radians(lon2 - lon1)
    a = math.sin(dlat / 2) * math.sin(dlat / 2) + math.cos(math.radians(lat1)) \
        * math.cos(math.radians(lat2)) * math.sin(dlon / 2) * math.sin(dlon / 2)
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    d = R * c # Distance in km
    return d
```

En cuanto a el momento de recogida, tenemos por una parte la fecha y por otra la hora. La hora la hemos transformado a segundos, para tratar así con datos numéricos.



OUTLIERS

Los datos que tenemos no siempre pueden ser idílicos aun habiendo sido tratados, a veces, podemos encontrar valores inusuales que ni se parecen al resto del dataset.

Sin embargo, estos podrían tomar estos valores, o bien por algún tipo de error (siendo así ruido), o bien porque se trata de algún caso anómalo. En caso de ser este último no podemos simplemente descartarlo de los datos, ya que estaríamos especializando en solamente una parte del problema el modelo.

Efecto eliminar los outliers en la correlación:

corr = -1 corr = +1 corr = n/a	fare_amount	pickup_da...	passenger...	km
fare_amount				
pickup_datetime				
passenger_count				
km				

corr = -1 corr = +1 corr = n/a	fare_amount	pickup_da...	passenger...	km
fare_amount				
pickup_datetime				
passenger_count				
km				

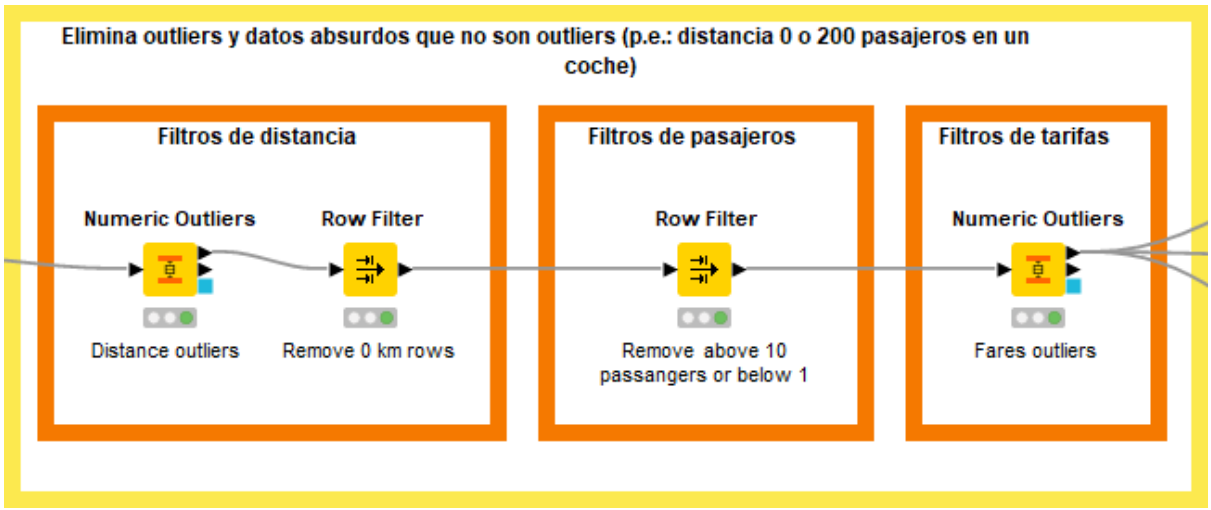
Como podemos observar, ahora los kilómetros guardan una fuerte relación con el precio del viaje. Este podría ser un indicio de que hemos conseguido quitar outliers que hacían de ruido dentro del data set.

Efecto eliminar los outliers en el error cuadrático:

Row ID	D Predicti...
R^2	0.001
mean absolut...	6.043
mean square...	97.927
root mean sq...	9.896
mean signed ...	0
mean absolut...	NaN
adjusted R^2	0.001

Row ID	D prediction
R^2	0.635
mean absolut...	1.557
mean square...	4.552
root mean sq...	2.133
mean signed ...	0
mean absolut...	0.199
adjusted R^2	0.635

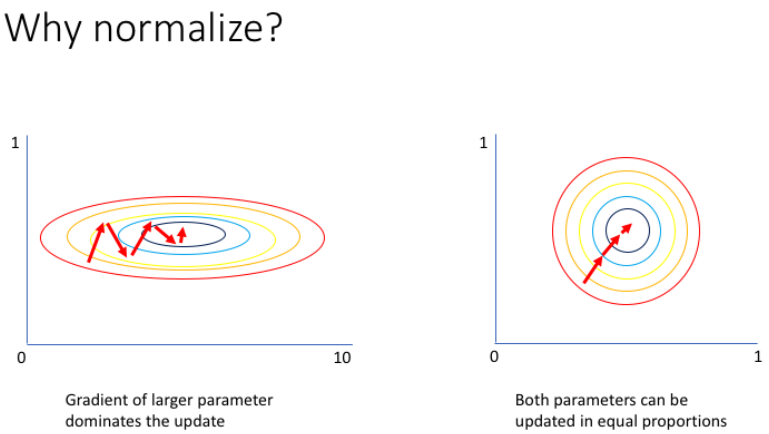
En el caso de que hemos hayamos hecho correctamente, hemos conseguido averiguar que los parámetros de el **momento de recogida** y la **cantidad de pasajeros** solamente afecta en un **0,1%** del **R^2**, por lo que podríamos ahorrar recursos eliminándolos del proceso.



NORMALIZACIÓN

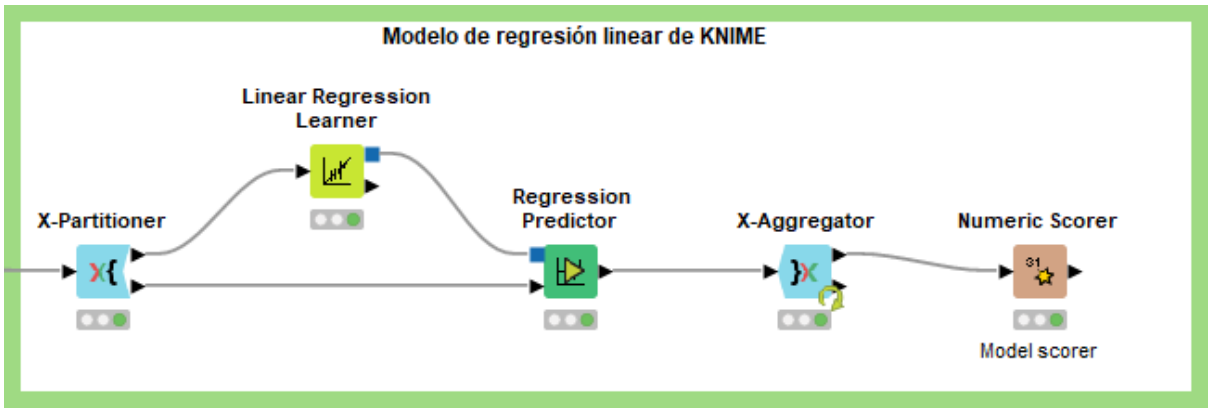
En el caso de la regresión con descenso de gradiente, es interesante hacer la normalización para a ver que la superficie de error sea más “esférica” y regular.

De este modo, conseguimos que el modelo mucho mejor, ya que acelera la convergencia y la hace de manera más controlada, ya que hacemos que la dirección del descenso del gradiente apunte más en la dirección del mínimo local de la función del costo.

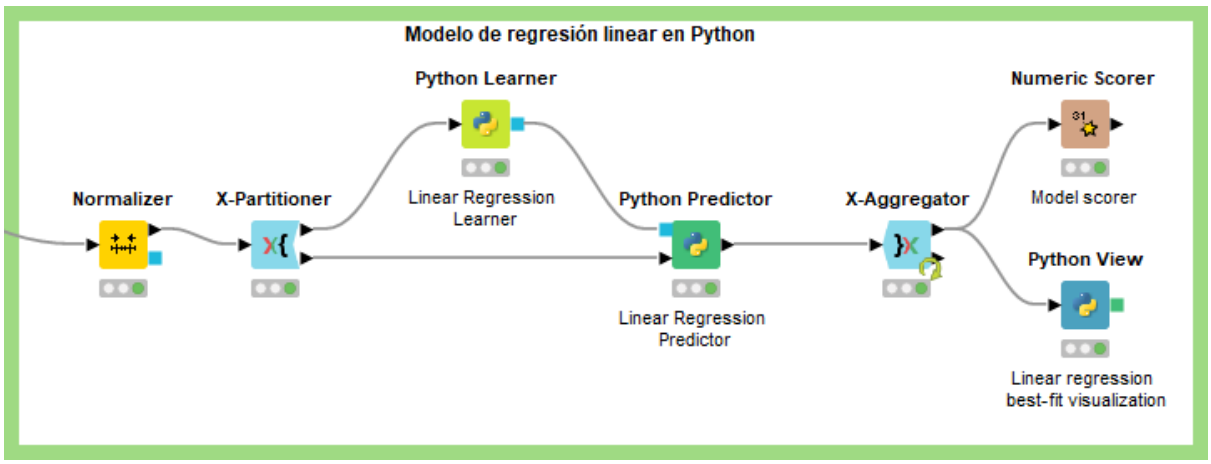


EVALUACIÓN DEL ALGORITMO

KNIME



PYTHON



KNIME VS PYTHON

En cuanto a los resultados, como podemos observar, hemos conseguido alcanzar muy de cerca al modelo presentado por KNIME, siendo este el ganador.

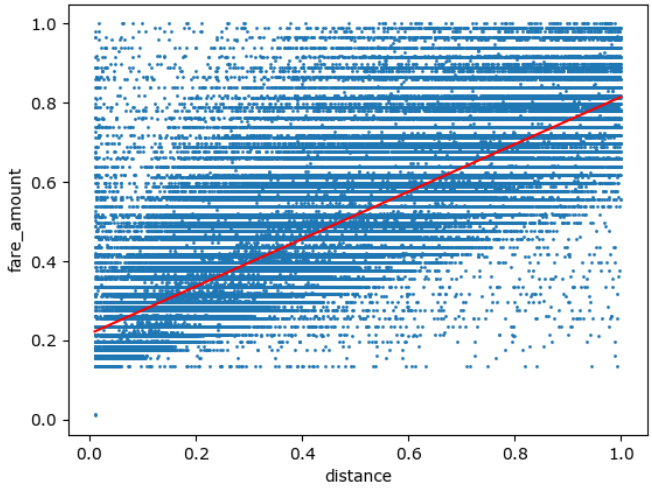
Row ID	D prediction
R^2	0.635
mean absolut...	1.557
mean square...	4.552
root mean sq...	2.133
mean signed ...	0
mean absolut...	0.199
adjusted R^2	0.635

Row ID	D prediction
R^2	0.634
mean absolut...	0.079
mean square...	0.012
root mean sq...	0.107
mean signed ...	0
mean absolut...	0.194
adjusted R^2	0.634

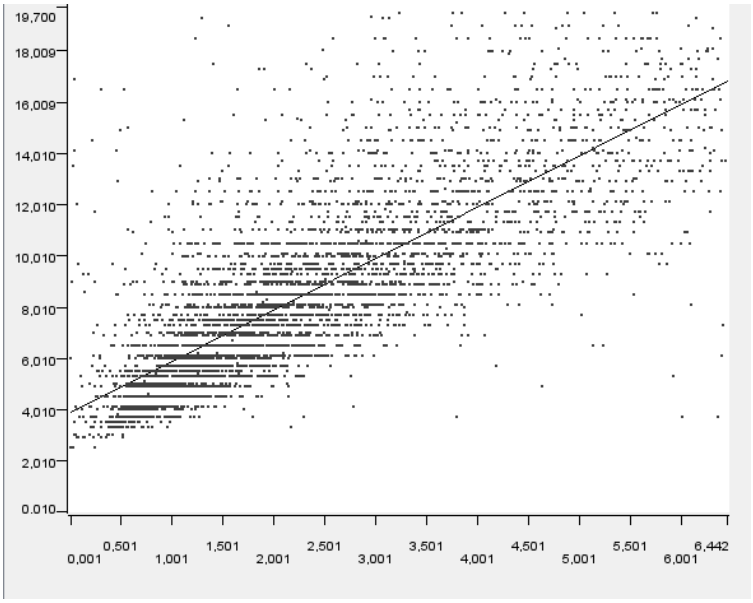
CONCLUSIÓN

Como hemos podido comprobar en los resultados, queda demostrado que el uso de las herramientas utilizadas en las anteriores prácticas nos proporciona una mejor visión de los datos y un aumento de los resultados en el modelo final.

En el proceso del tratamiento de datos hemos aprendido a como afrontar los **outliers** y a la problemática que conlleva su permanencia o borrado sin discreción, por lo que tendremos que añadir este elemento más a nuestro "arsenal" de trabajo.



Scatter Plot Python



Scatter Plot KNIME

BIBLIOGRAFÍA

- https://qph.fs.quoracdn.net/main-qimg-afdfbc63c83e097b3d831777397d905d
- https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931
- https://en.wikipedia.org/wiki/Linear_regression
- https://towardsdatascience.com/stochastic-gradient-descent-explained-in-real-life-predicting-your-pizzas-cooking-time-b7639d5e6a32
- https://en.wikipedia.org/wiki/Coefficient_of_determination
- https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1
- https://www.deeplearningwizard.com/deep_learning/boosting_models_pytorch/lr_scheduling/
- https://blog.knoldus.com/linear-regression-with-knime/
- https://www.dummies.com/article/academics-the-arts/math/statistics/how-to-calculate-a-regression-line-169795/
- https://www.mathcentre.ac.uk/resources/uploaded/mc-ty-strtlines-2009-1.pdf