Taxi trajectory Prediction I

Final Project

M11015Q06 顏齊

B10733010 陳安

Introduction

In this challenge, we have to build a predictive framework that is able to infer the final destination of taxi rides in Porto, Portugal based on their (initial) partial trajectories.

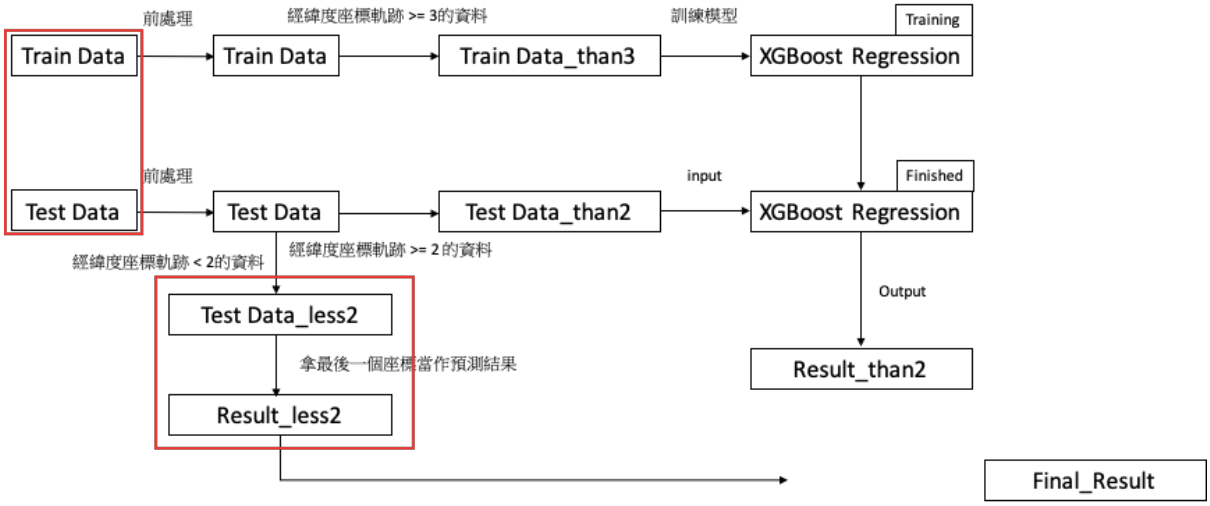
TRIP_ID	CALL_TY	ORIGIN_C	ORIGIN_S	TAXI_ID	TIMESTAMP	DAY_TYP	MISSING_	POLYLINE
T1	B	NA	15	20000542	1408039037	A	FALSE	[[-8.585676,41.148522],[-8.585712,41.148639]
T2	B	NA	57	20000108	1408038611	A	FALSE	[[-8.610876,41.14557],[-8.610858,41.145579],[
T3	B	NA	15	20000370	1408038568	A	FALSE	[[-8.585739,41.148558],[-8.58573,41.148828],[
T4	B	NA	53	20000492	1408039090	A	FALSE	[[-8.613963,41.141169],[-8.614125,41.141124]
T5	B	NA	18	20000621	1408039177	A	FALSE	[[-8.619903,41.148036],[-8.619894,41.148036]
T6	A	42612	NA	20000607	1408037146	A	FALSE	[[-8.630613,41.178249],[-8.630613,41.178249]
T7	B	NA	15	20000310	1408038846	A	FALSE	[[-8.585622,41.148918],[-8.58564,41.1489],[-8
T8	A	31780	NA	20000619	1408038948	A	FALSE	[[-8.582922,41.181057],[-8.582004,41.181813]
T9	B	NA	9	20000503	1408038563	A	FALSE	[[-8.606529,41.14467],[-8.606673,41.144724],[
T10	B	NA	15	20000327	1408038021	A	FALSE	[[-8.585658,41.148576],[-8.585703,41.148603]
T11	B	NA	56	20000664	1408038267	A	FALSE	[[-8.591229,41.162706],[-8.591229,41.162715]
T12	C	NA	NA	20000160	1408038946	A	FALSE	[[-8.585694,41.148603],[-8.585757,41.148684]
T13	C	NA	NA	20000017	1408039130	A	FALSE	[[-8.580105,41.159394],[-8.580231,41.159358]
T14	C	NA	NA	20000312	1408036255	A	FALSE	[[-8.665353,41.18616],[-8.666892,41.188311],[
T15	C	NA	NA	20000497	1408038388	A	FALSE	[[-8.649423,41.154354],[-8.650197,41.154138]

Looking at the dataset, we thought “Timestamp” was important because some people may have the habit of going to certain places at certain times. But after trying for several times, the effect was not as good as expected, so we did not use this feature in our final version.

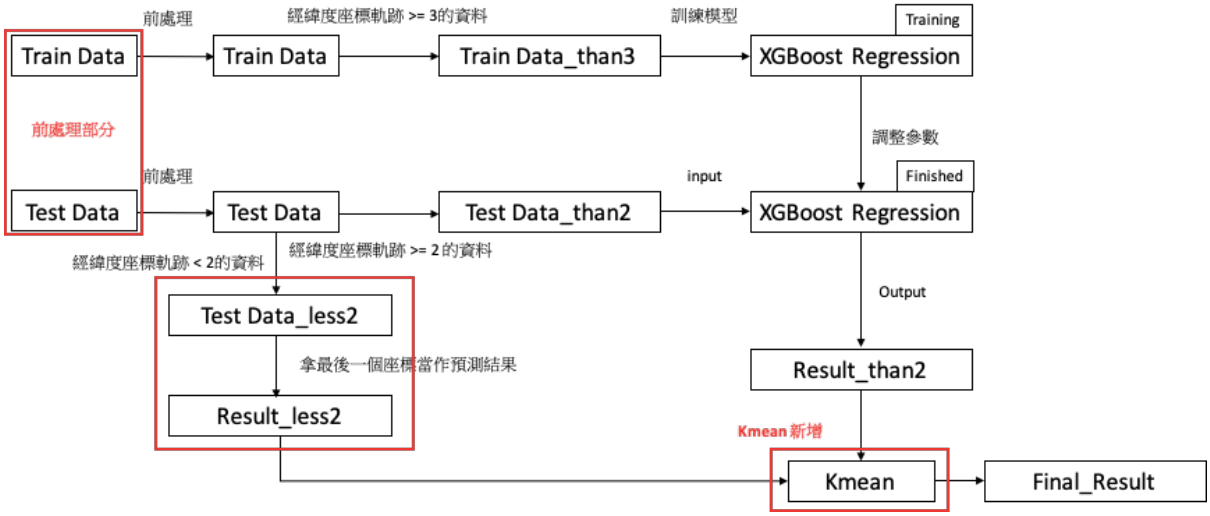
In the part of latitude and longitude, we have considered it for a long time, because we are not sure whether the coordinates close to the starting point are more important or the coordinates close to the end point, or whether it is not important, etc.

Architecture diagram

First version



Final version



1. First version : We select the training data whose longitude and latitude trajectory is greater than 11, which are the first 5 coordinates, the last five coordinates, and the predicted value. The model we use is XGBOOST.

2. Final version : We select the training data whose longitude and latitude trajectory is greater than 3, which are the first 1 coordinate, the last 1 coordinate, and the predicted value. The model we used is XGBOOST, and K-means algorithm is also added to help the classification problem of the destination.

Feature selection & Preprocessing

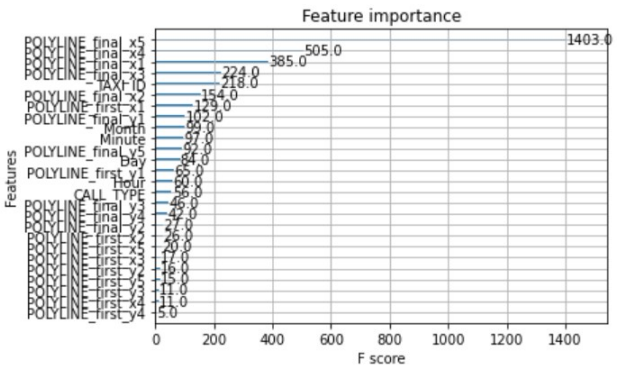
1. First version :

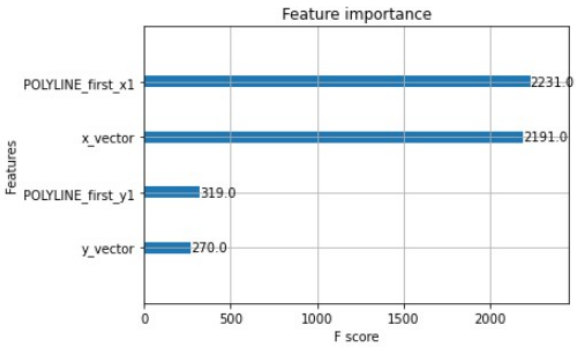
   - CALL_TYPE
   - TAXI_ID
   - Month
   - Day
   - Hour
   - Minute
   - POLYLINE_first_x1 (and final)
   - POLYLINE_first_x2 (and final)
   - POLYLINE_first_x3 (and final)
   - POLYLINE_first_x4 (and final)
   - POLYLINE_first_x5 **(and final)**

2. Final version :

   - x_vector
   - y_vector
   - POLYLINE_first_x1
   - POLYLINE_first_y1

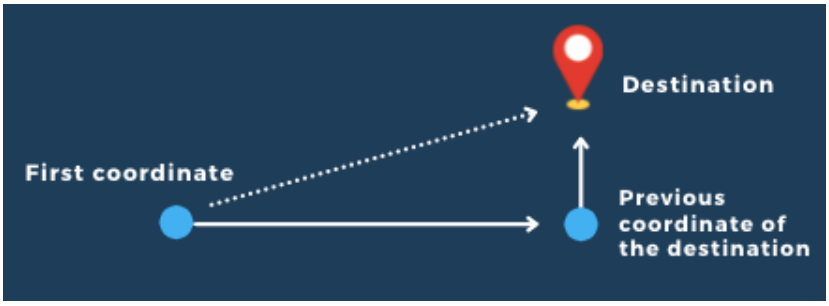
x_vector means “POLYLINE_final_x1 - POLYLINE_first_x1”
y_vector means “POLYLINE_final_y1 - POLYLINE_first_y1”





We found that in the first version, there are many features that are not so important, such as the beginning coordinates and time. It cannot predict accurately and also wastes training time.And we don't think the last coordinate can pinpoint exactly where the destination is, because many tracks turn or go back in the last 15 seconds. A better way is: compute a large vector in the x and y directions. In this way, we can know the final direction of the destination, and then put in the initial

coordinates to predict. Then, we delete the data that " |x_vector| > 0.1| , |y_vector| > 0.1“



Algorithm

1. X G B O O S T : (eXtreme Gradient Boosting)

- Based on Gradient Boosting
- Using the technique of random sampling of features, like random forest, the features are randomly selected when each tree is generated

2. K-Means :

• Clustering Algorithm

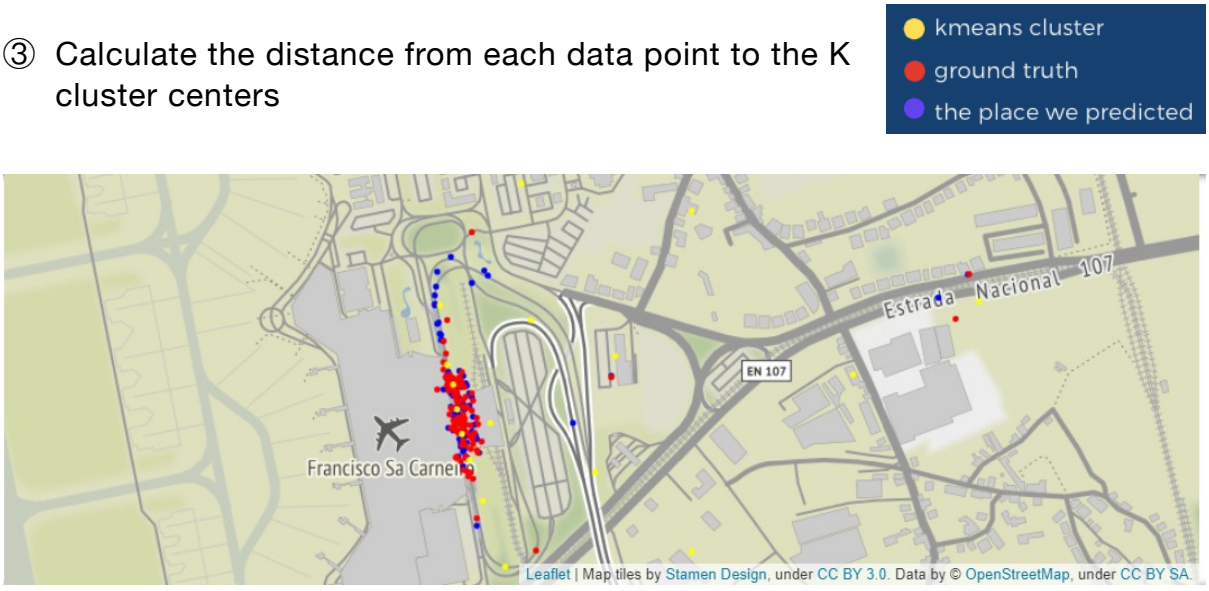
• Unsupervised Learning

Step:

① set the k

② randomly set K cluster centers in the feature space

③ Calculate the distance from each data point to the K cluster centers



The reason for using K-means, although Xgboost already has good results, K-means can help us classify some unclassified destinations to more common places.

Result on Kaggle

- First version

public ranking : 170
private ranking : 122

- Final version

public ranking : 51
private ranking : 84

Reference

1. https://ithelp.ithome.com.tw/articles/10273094

2. https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/635146/

3. https://chih-sheng-huang821.medium.com/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E9%9B%86%E7%BE%A4%E5%88%86%E6%9E%90-k-means-clustering-e608a7fe1b43

4. https://alankrantas.medium.com/kmeans-%E8%83%BD%E5%BE%9E%E8%B3%87%E6%96%99%E4%B8%AD%E6%89%BE%E5%87%BA-k-%E5%80%8B%E5%88%86%E9%A1%9E%E7%9A%84%E9%9D%9E%E7%9B%A3%E7%9D%A3%E5%BC%8F%E6%A9%9F%E5%99%A8%E5%

AD%B8%E7%BF%92%E6%BC%94%E7%AE%97%E6%B3%95-%E6%89%80%E4%BB%A5%E5%AE%83%E5%88%B0%E5%BA%95%E6%9C%89%E5%95%A5%E7%94%A8-%E4%BD%BF%E7%94%A8-scikit-learn-%E8%88%87-python-5dd8c0c8b167