Ride Fare Classification

Department of Computer Science and Engineering

University of Moratuwa
Academic Year 2019

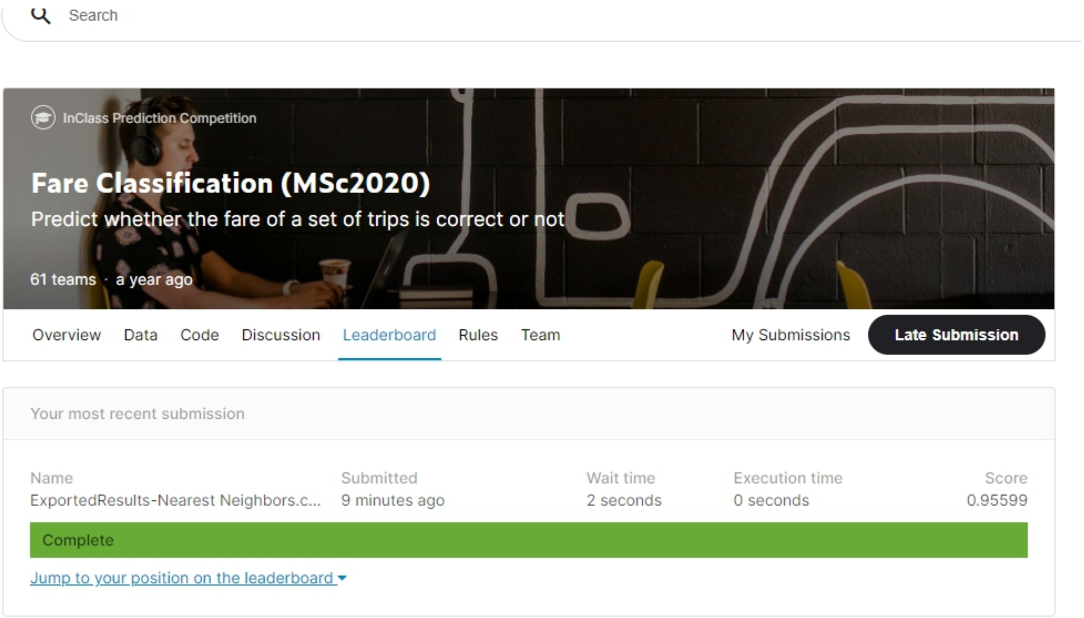
W.A.D.N.S. Wijesuriya
199373A

CS5621 - Machine Learning

Dr. Uthayasanker Thayasivam

Ride fare classification

Kaggle User Id : nicumwijesuriya
Kaggle Score : 0.95599
Public leader board rank: Not avaialable
Private leader board rank: Not avaialable
Link to the solution : https://www.kaggle.com/submissions/21552762/21552762.raw



Fare Classification (MSc2020)

Predict whether the fare of a set of trips is correct or not

61 teams · a year ago

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
ExportedResults-Nearest Neighbors.c...	9 minutes ago	2 seconds	0 seconds	0.95599

Complete

Jump to your position on the leaderboard

Introduction

This solution contains a classifier to classify whether a mentioned fare for a taxi fare is correct or not. Training and test sets were provided by the kaggle competition.

Feature engineering

From the given data GPS locations were not useful at all as raw data. So they were transformed using “Harversine formula” to get the distance between the mentioned locations.

Trip start and end times were combined to get the actual time taken to complete the trip. Assuming the system used in the taxi company is correct.

Therefore in this solution feature reduction was used.

If values were not available for any feature, these records were not used in training the classifier.

Classification techniques used

In this solution following classifiers were tried :

- Nearest Neighbors
- Linear SVM
- RBF SVM
- Decision Tree
- Random Forest
- Neural Net
- AdaBoost
- Naive Bayes
- QDA

Following are the results for each classifier:

Classifier : Nearest Neighbors
Accuracy : 0.8979191205339615
Precision : 0.914763458401305
Recall : 0.9777680906713164
F1 score : 0.9452170248630425

Classifier : Linear SVM
Accuracy : 0.8994895956026698
Precision : 0.9076
Recall : 0.9891020052310375
F1 score : 0.9465999165623696

Classifier : RBF SVM
Accuracy : 0.8979191205339615
Precision : 0.9010252365930599
Recall : 0.9960767218831735
F1 score : 0.9461697722567288

Classifier : Decision Tree
Accuracy : 0.9057714958775029
Precision : 0.9137792103142627
Recall : 0.988666085440279
F1 score : 0.949748743718593

Classifier : Random Forest
Accuracy : 0.90616411464468
Precision : 0.9085487077534792
Recall : 0.9960767218831735
F1 score : 0.9503015179871076

Classifier : Neural Net
Accuracy : 0.8924224577934825
Precision : 0.9092382495948136
Recall : 0.978204010462075
F1 score : 0.9424611507769843

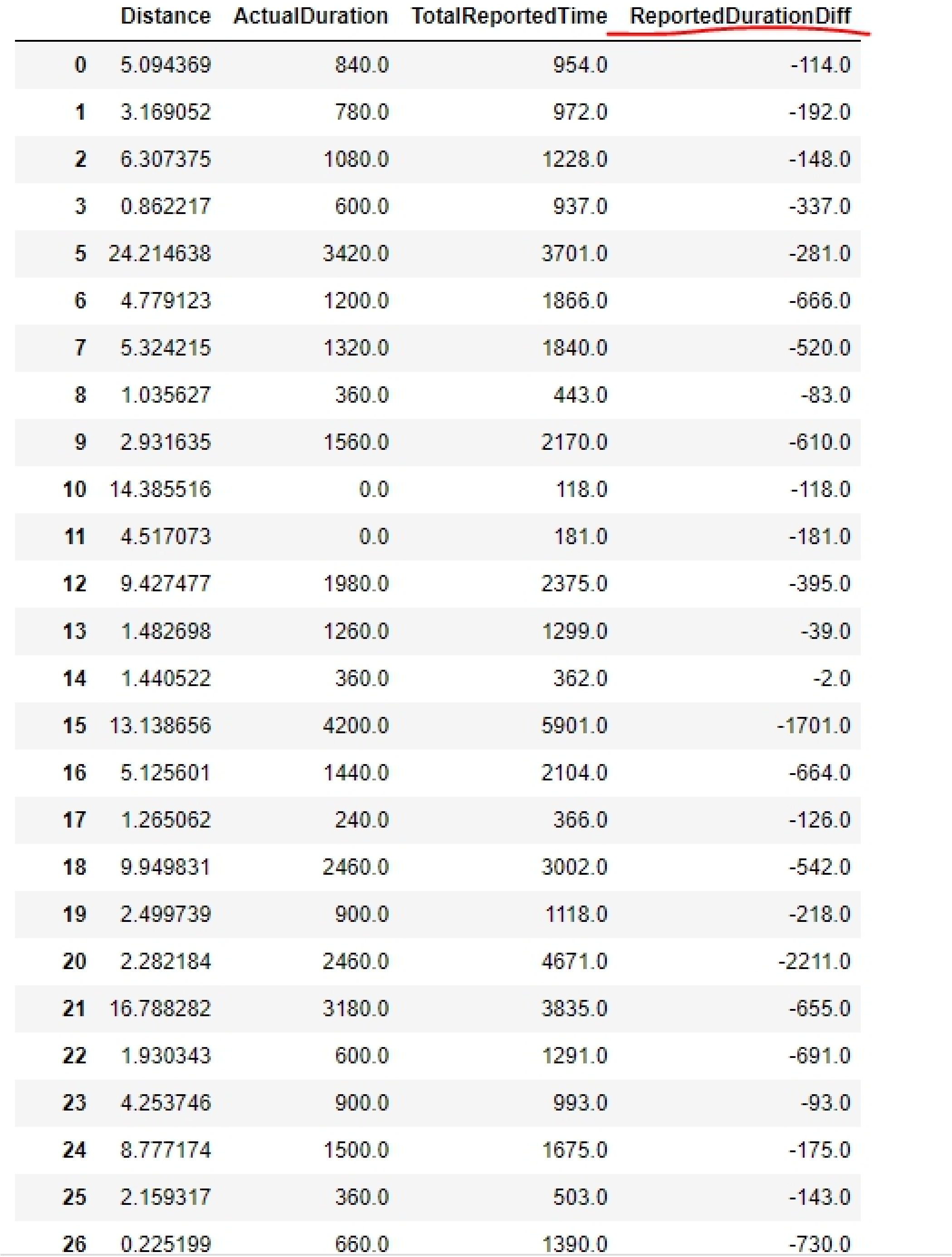
```
Classifier : AdaBoost
Accuracy : 0.9002748331370239
Precision : 0.9054054054054054
Recall : 0.993025283347864
F1 score : 0.9471933471933472

Classifier : Naive Bayes
Accuracy : 0.9014526894385552
Precision : 0.9055180627233029
Recall : 0.9943330427201394
F1 score : 0.9478495740702264

Classifier : QDA
Accuracy : 0.894778170396545
Precision : 0.9068273092369478
Recall : 0.984306887532694
F1 score : 0.9439799331103679
```

Sampling techniques used

There was a high imbalance between correct and incorrect labels. Therefore oversampling was used to add more records containing label “incorrect”. Randomly selected “incorrect” labeled records were duplicated in the dataset.

Noteworthy observations:

Once difference between actual time and reported time was calculated, for most of the records reported time was greater than actual time calculated from the system. This was observed for records labeled as “correct” as well.



	Distance	ActualDuration	TotalReportedTime	ReportedDurationDiff
0	5.094369	840.0	954.0	-114.0
1	3.169052	780.0	972.0	-192.0
2	6.307375	1080.0	1228.0	-148.0
3	0.862217	600.0	937.0	-337.0
5	24.214638	3420.0	3701.0	-281.0
6	4.779123	1200.0	1866.0	-666.0
7	5.324215	1320.0	1840.0	-520.0
8	1.035627	360.0	443.0	-83.0
9	2.931635	1560.0	2170.0	-610.0
10	14.385516	0.0	118.0	-118.0
11	4.517073	0.0	181.0	-181.0
12	9.427477	1980.0	2375.0	-395.0
13	1.482698	1260.0	1299.0	-39.0
14	1.440522	360.0	362.0	-2.0
15	13.138656	4200.0	5901.0	-1701.0
16	5.125601	1440.0	2104.0	-664.0
17	1.265062	240.0	366.0	-126.0
18	9.949831	2460.0	3002.0	-542.0
19	2.499739	900.0	1118.0	-218.0
20	2.282184	2460.0	4671.0	-2211.0
21	16.788282	3180.0	3835.0	-655.0
22	1.930343	600.0	1291.0	-691.0
23	4.253746	900.0	993.0	-93.0
24	8.777174	1500.0	1675.0	-175.0
25	2.159317	360.0	503.0	-143.0
26	0.225199	660.0	1390.0	-730.0