CS5691:Pattern Recognition and Machine Learning
Data Contest Report

ME1B122 ME1B156
Team Totem Twins

Saturday 6th February, 2021

Contents

1 Introduction

The task focuses on building prediction models for members of Biker-Interest-Group and trying to predict which bike-tours will be of interest to bikers. Extra information such as biker's previous interests, his/her demographic details, past tours, friend circle etc, are provided for the training of the model.

To learn about the bikers, one can access their feature information and friends-network. Information regarding tours and its participants is also available in tour-convoy. Supervised data is available about biker's preference is available in train file. The files provided in the training set and their purpose are listed in Table 1.

File name	**File content**
Bikers	Feature information about bikers
Tours	Feature information about the tours
Bikers network	Social networks of the bikers
Tour convoy	List of bikers who showed interest in a particular tour
Train	Tours shown to a biker and about whether he liked it or not
Test	Tours shown to a biker

Table 1: Files in the dataset provided

2 Feature engineering

For successful training of the model, it is important to have important and trainable features(in float data type) to get good performance. A series of processes are performed to ensure such a training and test set is prepared.

2.1 Pre-processing existing features

For training the model, we need to ensure the features in the training dataset are of type float or int. So, missing and incompatible features are modified to obtain trainable features. In Table 2, we have listed all the modified features used by our model for training and testing.

File	**Feature**	**Preprocessing step carried out**
train.csv	timestamp	The timestamp feature is split into seconds, minutes, hours, day, month and year from **DD-MM-YYYY HH:MM:SS** format using **datetime** library
bikers.csv	member_since	The member since feature is split into day, month and year from **DD-MM-YYYY** format using **datetime** library
	time_zone	When area or location_id information is provided for a biker while the time_zone information is missing, these are used to compute the time zone using **Nominatim module of geopy** library

bikers.csv	bornIn	If bornIn feature is missing for a biker, it is filled with the minimum bornIn in the whole dataset, i.e., 1952
	gender	**One-hot encoding** is performed to generate two features namely, gender_male and gender_female
tour.csv	latitude	When the latitude feature is missing for a biker, the information about city, state and country is used to compute the latitude using **Nominatim module of geopy** library. In case if the information about city, state and country is also missing the latitude is filled with value 0
	longitude	When the longitude feature is missing for a biker, the information about city, state and country is used to compute the longitude using **Nominatim module of geopy** library. In case if the information about city, state and country is also missing the longitude is filled with value 0
	date	The tour date feature is split into day, month and year from **DD-MM-YYYY** format using **datetime** library

Table 2: Feature preprocessing

2.2 Generated features

For the model to perform better, it is important to provide good number of important features which can help in classification. So new features are developed from the information given in the biker, biker_network, tour and tour_convoy files. The list of generated features and their description is added in table Table 3.

Feature	**Description**
biker_latitude	The latitude of the biker's location is computed using **Nominatim module of geopy** library from the area information given for the biker. If the area feature is missing, the location_id is used to calculate the latitude
biker_longitude	The longitude of the biker's location is computed using **Nominatim module of geopy** library from the area information given for the biker. If the area feature is missing, the location_id is used to calculate the longitude
latitude_diff	Difference between the latitude of the biker's and the location and tour location
longitude_diff	Difference between the longitude of the biker's and the location and tour location
member_how_long	How long has the biker been a member of the biker community on the date of tour
biker_age_while_tour	The age of the biker on the date of tour

num_friends_going, num_friends_not_going, num_friends_maybe, num_friends_invited	The number of friends of the biker(obtained form biker network) who are going to the tour, not going to the tour, maybe going to the tour and invited to the tour respectively.
perc_friends_going, perc_friends_not_going, perc_friends_maybe, perc_friends_invited	The percentage of friends of the biker(obtained form biker network) who are going to the tour, not going to the tour, maybe going to the tour and invited to the tour respectively.
tour_going, tour_not_going, tour_maybe	The number of tours the given biker is going to, not going to and maybe going to respectively.
tour_popularity	The difference between the number of bikers going to a tour and not going to a tour. Can be used as a metric for tour comparison.
is_host_friend	Checks if host is a friend of the biker.
host_num_tours_going, host_num_tours_not_going, host_num_tours_maybe	The number of tours hosted by the particular host to which the biker will be going, not going and maybe going.
host_perc_tours_going, host_perc_tours_not_going, host_perc_tours_maybe	The percentage of tours hosted by the particular host to which the biker will be going, not going and maybe going.

Table 3: Developed features from the dataset

After these features are added, some features with redundant information in the the dataset are removed to generate a trainable dataset. Box plots for the generated features are in Figure 1, Figure 2, Figure 3, Figure 4.

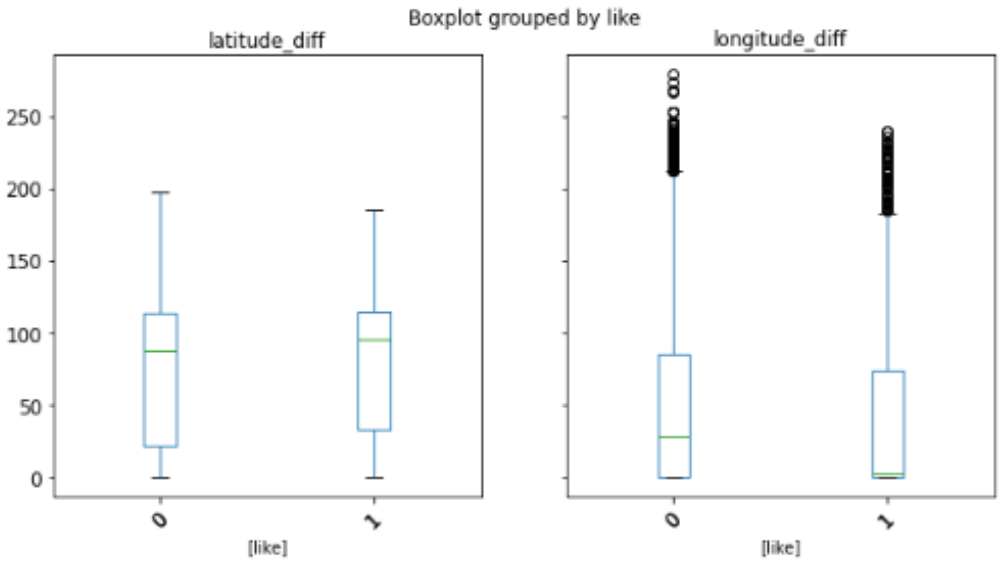


Figure 1: Box plot for latitude and longitude difference

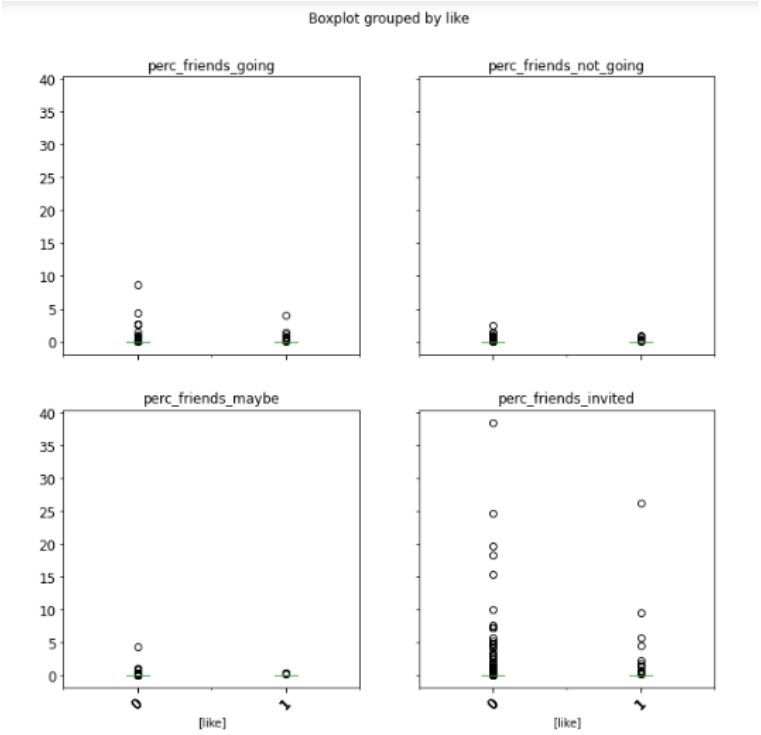


Figure 2: Box plot for percentage of friends going, not going, maybe going and invited to a given tour

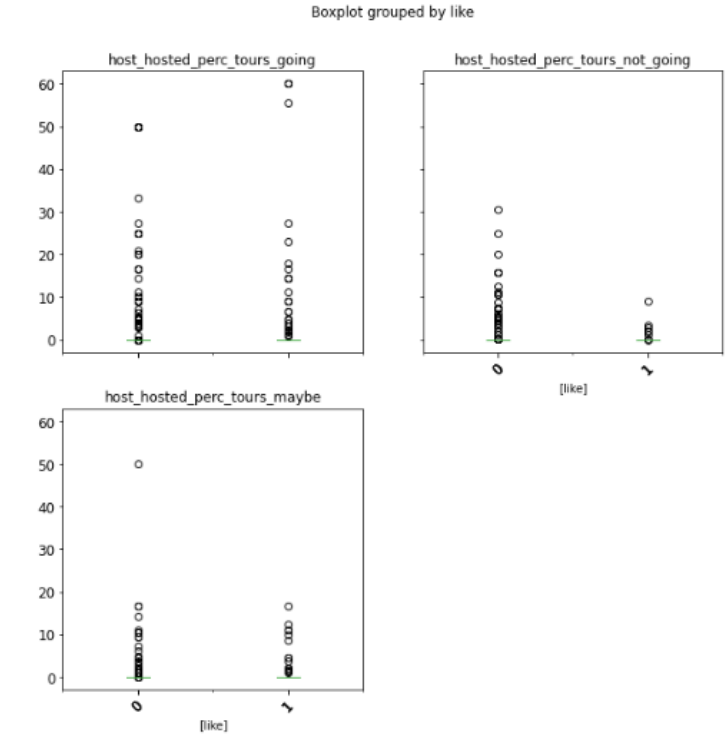


Figure 3: Box plot for tours hosted by host for which the biker is going, not going and maybe going

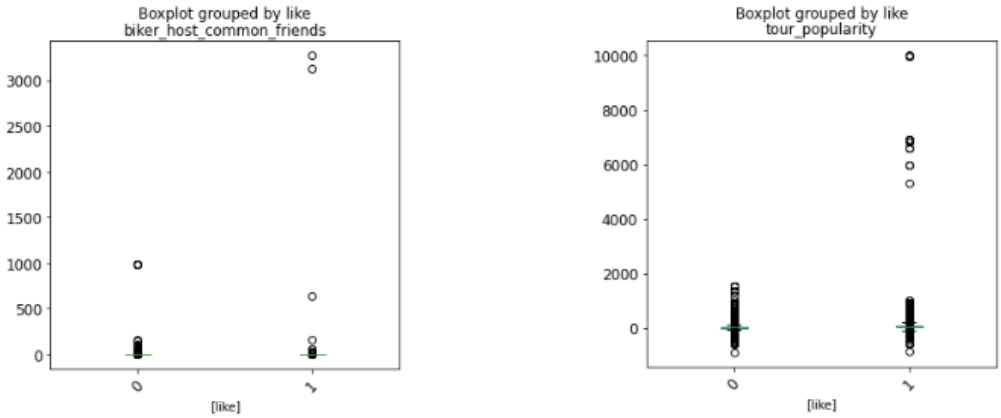


Figure 4: Box plot for biker host common friends and tour popularity

3 Model

The dataset consists of a mixture of continuous valued categorical columns. Gradient boosting machines are especially helpful in this scenario since boosting methods solve the class imbalance problem and improve the performance in test set. Also, modern GBM libraries come with in-built mechanisms to fit the model without over-fitting. Considering these, three classifiers namely XGBoost, CatBoost and LightGBM are used mainly.

3.1 Training Procedure

The generated features are used as training features to train the model with the like column being used as the label for the data points. We used only like column to train the model as we observed that the using dislike feature and learning a multi-class problem leads to reduced performance in the validation set. To choose the model and hyper-parameter combination we performed 5-fold cross-validation. The splitting of the dataset is carried out in such a way that the training set contains 80% of the biker ids while validation set consists of the remaining 20% biker ids. This is done to make sure that the model is tested on points that are different from the training set to ensure generalization of the model. While training the model, the 20% of the validation set is given to the model, so that the model stops training when the performance in validation set reduces(early stopping) to avoid over-fitting the training set.

3.2 Testing Procedure

While testing 5 models are generated as we are using 5-fold cross-validation. The models are then used to predict the probability score for each of the biker id-tour id pair given in test set. The five probability scores are averaged and then the tour ids are arranged in descending order. The results are written in a csv files and submitted.

3.3 Performance

We noticed that out of all the models, LightGBM classifier performed best. The two best models with their performance is given in Figure 5. The 5-fold cross-validation

Model	Model1	Model2
Objective	Binary log-loss	Binary log-loss
Num leaves	32	32
Subsample(% of samples)	0.6	0.5
colsample_bytree(% of features)	0.6	0.5
Learning rate	0.05	0.05
Metric(to avoid over-fitting)	F1 score	F1 score
Cross validation(F1 score)	0.4484	0.4762

Figure 5: Two best models

Fold Number	Validation Accuracy	Validation F1 score
1	0.747	0.451
2	0.747	0.410
3	0.761	0.499
4	0.748	0.425
5	0.764	0.457

Figure 6: Cross Validation scores of best models

scores for the best model is given in Figure 6

4 Technical Insights

Note: The importance of various features used by the model is given in Figure 7. The importance scores are normalized.

- From Figure 7, we can infer that **member_how_long_days, longitude_diff, tours_going, tour_popularity, host_hosted_num_tours_going** are a few important features. In the following points importance of these features will be analysed based on the density map for these features.
- **tours_going:** From Figure 8(a), we can see that the density for both like and dislike peak close to 0. But, the like distribution has significant probability of taking higher values, i.e., higher values of tours_going imply high probability of biker liking tour. Hence, this is also an important feature and is rightly identified by the model.
- **host_hosted_num_tours_going:** From Figure 8(c), we can see that the density for both like and dislike peak close to 0. But, the like distribution has significant probability of taking higher values, i.e., higher values of host_hosted_num_tours_going

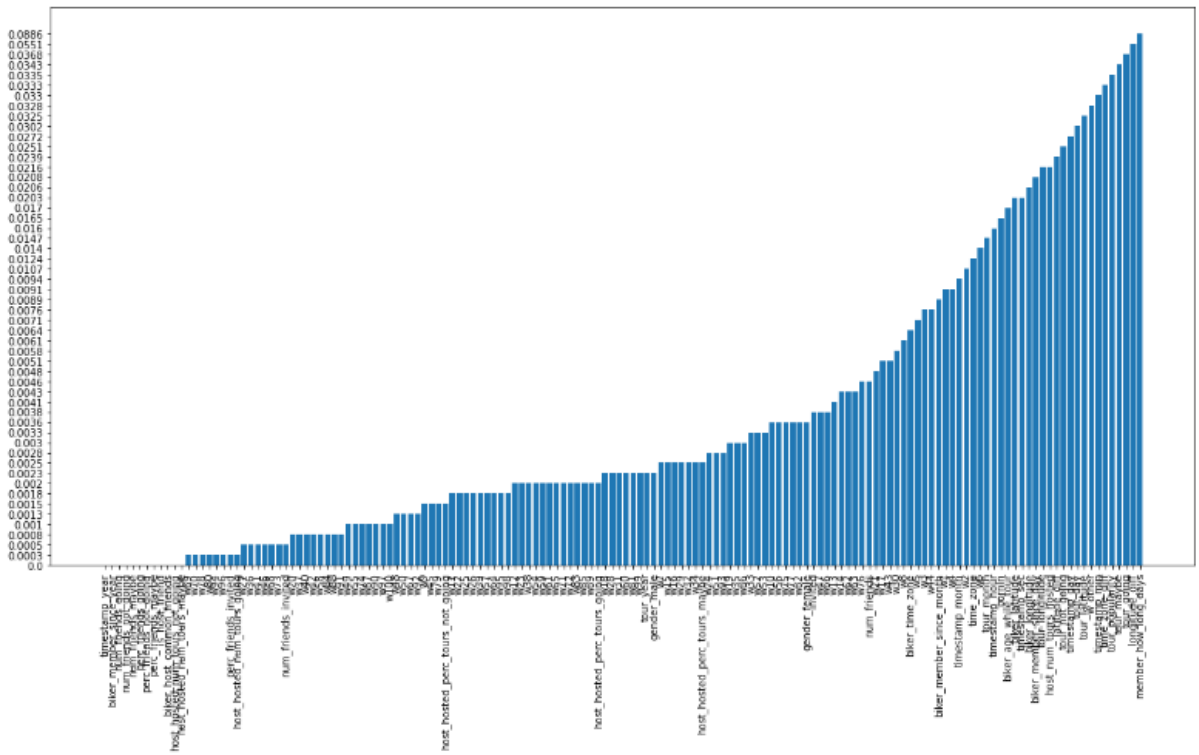


Figure 7: Importance of features

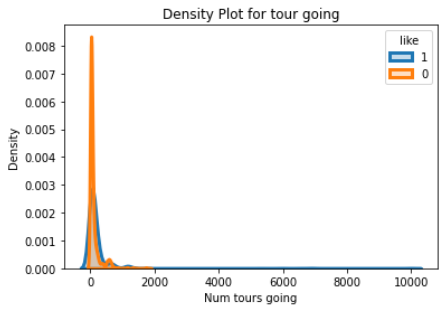
imply high probability of biker liking tour. Hence, this is also an important feature and is rightly identified by the model.

- **member_how_long_days:** From Figure 8(f), we can see that the density for both like and dislike peak close to 0. But, the like distribution has significant probability of taking higher values, i.e., higher values of member_how_long_days imply high probability of biker liking tour. Hence, this is also an important feature and is rightly identified by the model.

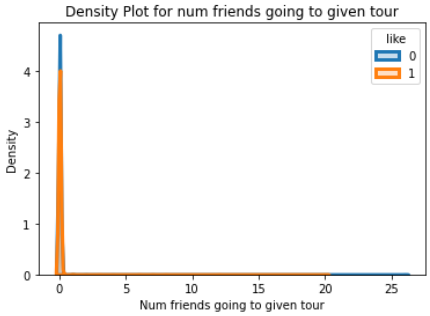
- **tour_popularity:** From Figure 8(h), we can see that the density for both like and dislike peak close to 0. But, the like distribution has significant probability of taking higher values, i.e., higher values of tour_popularity imply high probability of biker liking tour. Hence, this is also an important feature and is rightly identified by the model.

- **longitude_diff:** From Figure 8(e), we can see that the density for both like and dislike peak close to 0 and 80. But, the like distribution holds more probability of taking values close to 0 than the dislike distribution. Hence, this feature can be considered reliable up-to an extent even though it cannot be individually used for classification.

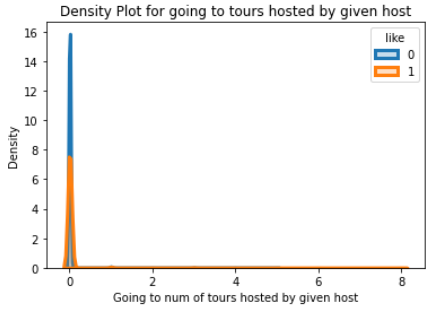
- **num_friends_going:** From Figure 8(b), we can see that the distribution is similar for both like and dislike label. Hence, num_friends_going feature is hence not a good feature for classification and the model also gives low importance score for the feature. The model correctly identifies the feature as a low importance one.



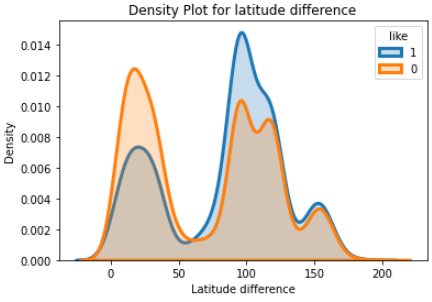
(a)



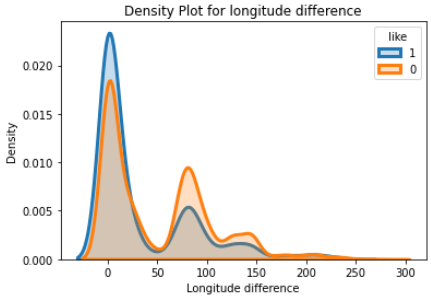
(b)



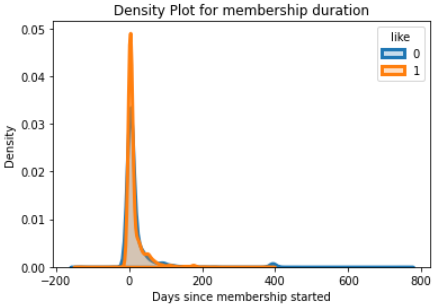
(c)



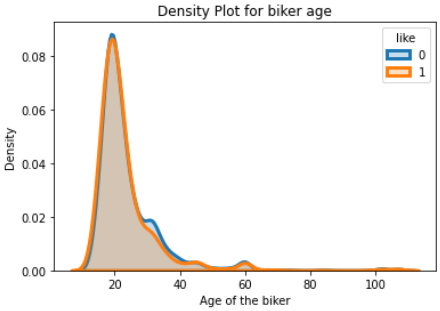
(d)



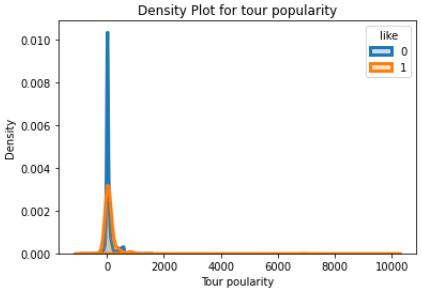
(e)



(f)



(g)



(h)

Figure 8: Density plot for various features

- **biker_age:** From Figure 8(g), we can see that the distribution is similar for both like and dislike label. Hence, biker_age feature is hence not a good feature for classification and the model also gives low importance score for the feature. The model correctly identifies the feature as a low importance one.

5 Conclusion

The project was very helpful in understanding the applications of various concepts learnt during the course. The feature engineering step followed in this project was very helpful to improve the skill to link dataset in order to understand the core problem and solve it. We believe that parts of project such as hyper-parameter fine-tuning, searching for various models helped increase our knowledge towards solving real world problems. We also learnt about how to resolve over-fitting, class imbalance and other constraints that are generally observed in real-world datasets. The given problem set also increased our understanding of recommendation systems.