CS5691 - Pattern recognition and machine learning

Sanjana Prabhu EE17B072

February 4, 2021

1 Introduction

The problem given to us was to predict whether a certain biker is interested in a certain tour or not. We were given a test file containing a list of bikers with a number of tours assigned to each biker and our task was to arrange these tours in our predicted order of preference. Demographic details for the bikers and tours, as well as information about the bikers friend network and previous tour-attendee information was provided. We were expected to leverage machine learning techniques to build the prediction models.

2 Our approach

2.1 To generate the feature vector

We were given four .csv files which contained information about the bikers and tours - bikers.csv and tours.csv contained demographic details about the bikers and tours respectively, tour-convoy.csv contained a list of bikers who were going to, maybe going to and not going to a particular tour, bikers-network.csv contained a list of friends for each biker. Further, we were given supervised data in the form of a list of bikers and whether or not they had liked the tour assigned to them, i.e., every entry in the train.csv file had a tuple (biker ID, tour ID) and 1 or 0 corresponding to whether the biker likes the tour or not and the same for dislike. We considered the tuple (biker ID, tour ID) to represent the data input to a classifier, which takes in the features that we give for each tuple and classifies it into a 1 or a 0, which represents whether the biker liked the tour or not, respectively. So, the input feature vector to the classifier would consist of features of the biker corresponding to the biker ID, features of the tour corresponding to the tour ID, some features representing correlation between biker and tour features, some features conveying information about the behaviour of the friends of that particular biker and lastly some features from the tour-convoy of the particular tour. Ultimately, we ended up using only the like classifier to order the tours, because the dislike vector in train.csv was so sparse, that

even classifying everything as 0 was giving a very high accuracy and hence the dislike classifier was not useful. The probability of the predicted class being 1, as given by our like classifier was used to order the tours for each biker.
Therefore, for every tuple (biker ID, tour ID) present in the train.csv and test.csv files, the feature vector was generated in the following fashion :

- Biker features - We used all the features given for the biker, namely language, location, birth year, gender, date of joining, area and timezone. Categorical features such as language, location, area and gender were encoded using integers. Missing values were replaced by NaN, which were automatically converted into '-1' by our encoding function(pandas.factorize). Our approach towards missing features was to put all of them in a new category altogether, so this worked out well as '-1' formed a new category. The date/time features(date of joining in this case) were mapped onto unique integers, that is by multiplying the year, month and day by suitable numbers, we ensured that no two dates could output the same integer. Also, the areas were used to extract the coordinates(latitudes and longitudes) for each biker. If the area was missing, then the location IDs were used to find the latitude and longitude. This was done using the geopy library and the reason why we extracted latitude and longitude will be explained later on.

- Tour features - For the tours, biker ID, tour date, city, state, country, pin code, latitude, longitude, all 100 w values, w other were used. Similar to the bikers, all dates were mapped to unique integers and the categorical features were integer encoded. Missing values were replaced by NaN, so that our ML model could handle it.

- Information from tour convoy - This was used in two ways : firstly, as a tour feature and secondly, in conjunction with friends network(as a correlation feature). First, we computed the fraction of bikers going to the particular tour out of the total number of bikers invited to that tour. Both these lists are given in the tour-convoy for all the tours. Similarly, we computed the fraction of bikers who said maybe and who are not going to that tour. These fractions were added as features.

- We added the number of friends of a biker as a feature.

- Day of the week : the day of the week of the tour was also added.

- Sum of word count : the sum of all the word counts (w1 to w100 and wother) was added as it signifies the length of the description of a tour.

- We also added the number of times a biker has appeared in the train/test set as a feature as this represents the number of tours a biker is getting invited to, roughly.

- Further, we added some features which represent the correlation between the bikers and tours, such as :
  - Distance : We had computed the latitudes and longitudes for the bikers previously. Since we have been given the latitude and longitudes of the tours, we used a simple distance formula to compute the distance between the biker and the tour.
  - Time differences : We computed three types of time differences, namely, difference between the tour date and date of joining the bikers interest group for the biker, difference between the tour date and the timestamp(the time when the biker was informed about the tour) and the difference between the timestamp and the date of joining for the biker. Eventually, these features were among the most important ones(as given by the feature importance attribute of the classifier used).
  - We found out the intersection set of the bikers going to the tour and the set of friends of the biker. The size of this set is also added as a feature. Similarly the intersection between the friends set and the set of bikers who said maybe and not going to that tour can also be computed and these sizes are also added as features. To reiterate, the biker and tour corresponds to the biker ID and tour ID in the tuple for which we are computing a feature vector.

Therefore, we ended up forming a 135 dimensional feature vector for each tuple, which, along with the vector indicating likes, were used to train our model.

2.2 To train the model

Since a lot of features were categorical, we directly used LightGBM to train our model. LightGBM directly handles categorical features without any requirement of one hot coding etc. For further model ensembling, we trained our LightGBM classifier with a random fraction of the training data(0.8-0.9) and averaged the results of a number of such classifiers to obtain our final prediction probability. Using this probability, we ordered the tours. We tuned parameters like number of leaves, learning rate and number of boosting rounds using hyper-parameter tuning using a grid search. Our best performing model gave an accuracy of 75.846 on the public dataset. However, we obtained an accuracy of 68.924 on the private dataset(this model gave 75.7 on the public dataset).

3 Feature importance and inference

Using the feature importance attribute of the LightGBM classifier, we determined the most important features, they were:

- Time difference between the joining date of the biker and the tour date : intuitively, this plays a very important role in determining whether the biker goes for/likes a tour or not. If it is too high, it is possible that the biker has joined after the tour date and he/she would not go.
- Biker count : Surprisingly, the number of times that particular biker occurs in the train/test files forms a very important feature. We assume this could be because a biker who generally goes to many tours would go to tours very often and therefore would be invited to many as well, therefore making biker count an important feature.
- Area : Again, this is not an obviously important feature, but, we can infer that the leniency of a biker towards going to/liking tours might depend on the area of the biker. For example, the tours culture in say, Indonesia, may be more prevalent than in other places.
- Distance : This is an obvious important feature. If the distance between the biker and the tour is very large, the biker would probably not prefer going.
- Ratio of not going to invited in tours-convoy : If the fraction of bikers not going to a tour out of those invited is high, it is probably not a very exciting tour and therefore other bikers may not go too. Hence it is an important feature.

4 Public vs Private leaderboard

There was a nearly 7 percent drop in the accuracy from the public to private leaderboard. Some reasons :

- Towards the end, I was trying to tune the parameters quite repeatedly, including the number of boosting iterations and the number of ensemble models, this probably led to overfitting. One of my previous submissions which gave 75.4 accuracy on the public dataset, gave 69.3 on the private dataset(as shown in Fig 1), which makes it obvious that the submission that performed best on the public dataset was slightly overfitted to it.
- Secondly, it could have been because the private dataset was quite different in terms of distribution of features from the public dataset.

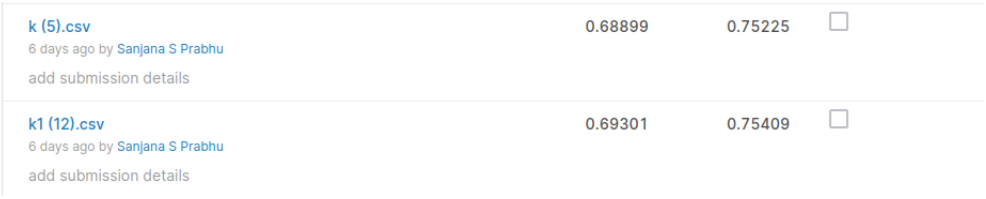

Figure 1: This model(k1(12).csv) is the best performing on the private dataset. It had slightly different hyperparameters from the best performing model on the public dataset.

This is evident from the fact that the highest accuracy on the private dataset is nearly 6 percent less than that of the public dataset. Similarly even the baseline accuracies reduced by nearly 6 percent.

5 In hindsight..

The first approach that came into my mind while reading the problem statement for the data contest was collaborative filtering. But, initially I was not able to find a suitable library to implement collaborative filtering, so I decided to generate a feature vector and use a boosted decision tree such as CatBoost or LightGBM. However, I eventually realised that the models were giving higher accuracies when I added a few correlation features such as distance, time differences etc. which turned out to be among the most important features as well. So, in hindsight, the model is similar to collaborative filtering as it has a subset of the feature vector corresponding to the biker's features, a subset corresponding to the tour's features and the rest of the features correspond to correlation or some sort of an inner product function between the biker features and tour features, which is the concept of collaborative filtering.