Report - Team Alpha Boost

CS18B032 - P Girinath

CS18B050 - Aniswar Srivatsa Krishnan

Steps of the Pipeline

- Preprocessing
- Feature Engineering
- Model
- Ensembling

Preprocessing

- The given data files are very large. E.g., tours.csv is 1.05GB.
- Hence working with the entire files in the RAM will lead to very slow operations or crashing of machines.
- Most of the data in the files is not required for most of the features.
- We trim the data in each file to just the data which contains the bikers and tours in train.csv+test.csv.

Location Data for Bikers

- The latitude and longitude for the bikers are not given explicitly. However, the area/address of the bikers are given.
- We use the geopy library to convert the area of the bikers into latitude and longitude.
- Since, geopy involves a network api, it might be very slow and slightly unstable.
- Hence, we load the locations from an offline file already created (as allowed)

Imputation of Location Values

- For biker locations, we use the mode of the locations existing in a particular timezone. Timezone seemed to work slightly better than taking location_id of bikers.
- For tour locations, the mode of the locations of the bikers who are invited + going to the tour are taken.
- For locations still missing, we take the mode of the locations of the friends of the tour organizer.
- We use median to impute the rest of the missing values.

Features

- We merge the train data with bikers and tours to create a dataframe with all the given attributes.
- We create 85 additional features, by using the given data in interesting ways.
- The list of the features are as follows:
    - Time difference between the timestamp and the tour date
    - Fraction of the invited people who are actually going
    - Total number of going, invited, maybe and not going to a particular tour
    - The distance between the biker and the tour

- The total number of friends of the biker who are going, invited, maybe, not going
- The ratio of the minimum difference between the timestamps and the tour dates of the tours presented to the biker and the current difference
- The country of the bikers with Frequency Rank encoding.
- The cluster number of the tour, with clusters formed by the word vectors of the tours
- The count of the biker in the train file
- The count of the biker in the test file
- The difference between the timestamp and the joining date of the biker
- The difference between the tour date and the joining date of the biker
- Whether the biker is a friend of the tour organiser
- Age of the biker
- Year, month, date and day of the week of the three dates, i.e, timestamp, member since and the tour date
- Whether each of the above dates falls on a weekend
- The quarter of the year in which each of the above dates fall
- The number of likes given by the biker
- The number of dislikes given by the biker
- The count of the tour in the train set

- The count of the tour in the test set
- The total number of going, invited, maybe and not going for the biker
- The similarity between the current tour's word vector and the average tour vector of the tours of going, notgoing, maybe and invited, for the biker.
- The similarity between the current tour's word vector and the average tour vector of the tours of going, notgoing, maybe and invited, for the biker's friends.
- Whether the tour falls in the month of december
- The tours are clustered by the word vector.
- The number of tours in the current tour's cluster to which the biker is going, invited, maybe, not going.
- The number of similar tours going, notgoing, maybe and invited of the people of the same language as of the biker, based on cluster
- The number of similar tours going, notgoing, maybe and invited of the biker, based on cluster.
- The number of people going, notgoing, maybe and invited for the current tour, among bikers friends
  The number of tours going, notgoing, maybe and invited of the people of the same language as of the biker, based on cluster

Model

- The given problem is modeled as a binary classification problem where the like=1 is considered as the positive class and like=0 is considered as the negative class.
- A common phenomenon which was observed was that considering dislike column resulted in reduction of score.
- The predicted probabilities of a tour being liked is used as the basis for ranking the tours of a particular biker in test set.
- A lightgbm classifier is used as the base model. Three lightgbm classifiers which differ in the number of trees are taken and averaged.
- Hyperparameter tuning is performed by KFold Cross validation error with log loss as the metric, as log loss correlates with the MAPk metric which is used for actual evaluation. The parameters, num_iterations, num_leaves and learning rate are tuned.
- The model gave a 5-Fold cross validation neg_logloss of around 0.44 .

Ensembling

- Each of the three classifiers mentioned above are bagged with 175 estimators.
- For each estimator, a random sample consisting of 90% of the training data is given as the training set.
- The outputs (predicted probabilities) of each of the estimators is aggregated by averaging.

Observations

- Location feature and imputation is important.
- Aggregate features involving tour and friends of a biker are crucial.
- Ensembling with bagging and or multiple classifiers improves performance.
- Using a faster classifier like lgbm in the initial stages makes playing with features easier and quicker.
- Sometimes reducing features can help increase score.
- Recommendation system based approaches tend to work poorly in this dataset, which maybe due to the cold start problem.
- The test data seems to be from a slightly different distribution than the train set and also the public and private sets might have different distributions and models

fitted to cross validation on the train set as well as the public leaderboard tend to have a drastic decrease in the private leaderboard.

- Generally trusting the cross validation error is better than the public leaderboard score, as the former is a better indicator of the private leaderboard score.

Submission_2

- Our second submission involves removing 35 features from the above list and averaging 10 lightgbms with two different feature_fractions.