PATTERN RECOGNITION AND MACHINE LEARNING CS5691

DATA CONTEST REPORT

Team Name : we-are-groot

Gunavardhan - CH18B035

Narendhiran - CH18B015

JANUARY 2021

Contents

INTRODUCTION 2
Overview 2
Initial Ideas 2
Data Processing 2
Feature Engineering 3
    Correlation Matrix 5
    Feature Importance Visualisation 6
Model Building 7
    Training 7
    Hyper parameter tuning 7
    Best settings 7
    Evaluation on Validation Data 8
    Generating Submissions 8
Observations 8
Conclusions 10

INTRODUCTION

Bikers Ranking Model we developed is based on a fast, distributed, high performance gradient boosting framework called LightGBM. We achieved 0.766 mean average precision using a validation split of 0.25 on the given data.

The accuracy on Kaggle was **0.734**.The final accuracy on test data was **0.711**. This report briefs the procedure we followed.

OVERVIEW

Kaggle data contest asks for a model that can match tours to bikers given biker and tour metadata and some demographic information. Basically a Tour recommendation model (decreasing order in which biker likes a tour) for bikers based on the information available with us.

INITIAL IDEAS

Collaborative filtering, which we considered might not be the best model as it is heavily based on the assumption that an effective similarity can be built on the biker/tour groups. Meaning for this model to be useful there should be significant overlap between the bikers and tours but that's not the case here. Also recommendations made by collaborative filtering models are usually on open-lists i.e., the candidates are from the entire set, but here we are asked to rank closed-set of tours for a certain biker. Hence might not be the best model.

Next immediate idea is **classification** as the output is binary whether the biker likes the event or dislikes(or maybe) the event. We can use this data to predict the probability with which he might like the tour, and rank them on it. Hence this is the idea we proceeded with.

DATA PROCESSING

Initial analysis of data showed us that the data matrices are very much sparse, so we had to make reasonable assumptions to complete the data matrix.

- In the train data set given on bikers and tour pairs there are 3370 likes and 10496 dislikes or maybes. Implies the ratio of likes : dislike/maybe of the data set is 0.321. This could be used for setting the prior.

- There were 1492 bikers whose bornin year isn't mentioned which is assumed to be median year 1952 from the rest of the 36717 biker entries. The 109 entries where gender isn't mentioned is taken as male as 61 in 100 of the bikers are males in majority. And the single entry whose join date in bikers club(member since) isn't available is taken to be '30-10-2012' as that is the day with major joiners into the club.
- Bikers area address is used to get latitude and longitude of the location and if not present is replaced with median values.
- Coming to tour data which has demographics on 3137972 tours is too sparse and time consuming to work on. But the information we need is just on the tours in train and test sets which are 7985 and 1996 respectively. Hence we first draw the demographics on the tours required which are in train and test.
- Then for each tourid based on the given address we find the latitude and longitude if not already present else, in case both are absent we replace it with median latitude and longitude from the tour set.

After processing each dataset as explained above they are merged with the train and test to form main dataset for each biker/tour pair, on which our model is built on. On it few features have been extracted which are described in the section below.

FEATURE ENGINEERING

The 'text' values in the bikers data set which are 'location id', 'language id', 'gender' in each column unique values are given different labels using label encoder to obtain data in numerical.

The 'member since' feature present in bikers and 'tour date' feature in tours which are in 'dd-mm-yyyy' format has been changed to three separate columns each (days months and years).

Features extracted:

Based on intuition, factors we consider when planning a tour are generally the number of friends who decide to go or not go, the distance between me and the tour location, time gap or interval between the tour date and time I received the notification, previous popularity of the tour.

Accordingly we can extract features from tour convoy and bikers network, the circle of friends who are going, not going, maybe and invited, also the **popularity** of the tour which in a sense can be quantified by total bikers going / (not going+maybe+invited), this might result in some 0/0 cases, which is taken care by replacing with zeros.

Distance between tour and biker location can be calculated using the latitude and longitude of tour and biker. **Time interval** in the number of days between tour date and notification can be calculated using the timestamp from the train data.

Other features like the number of friends extracted from bikers network, can be justified from observation that people with more friends are more outgoing, and hence are more likely to go on tours. Age feature of bikers can be justified as in general youth prefer to be more active and exploring with their friends.

Extra features which we added are on the intuition of popularity of tours which we thought can be quantified by total likes or total dislikes based on the train data are removed at the end, reason is mentioned in the observations section.

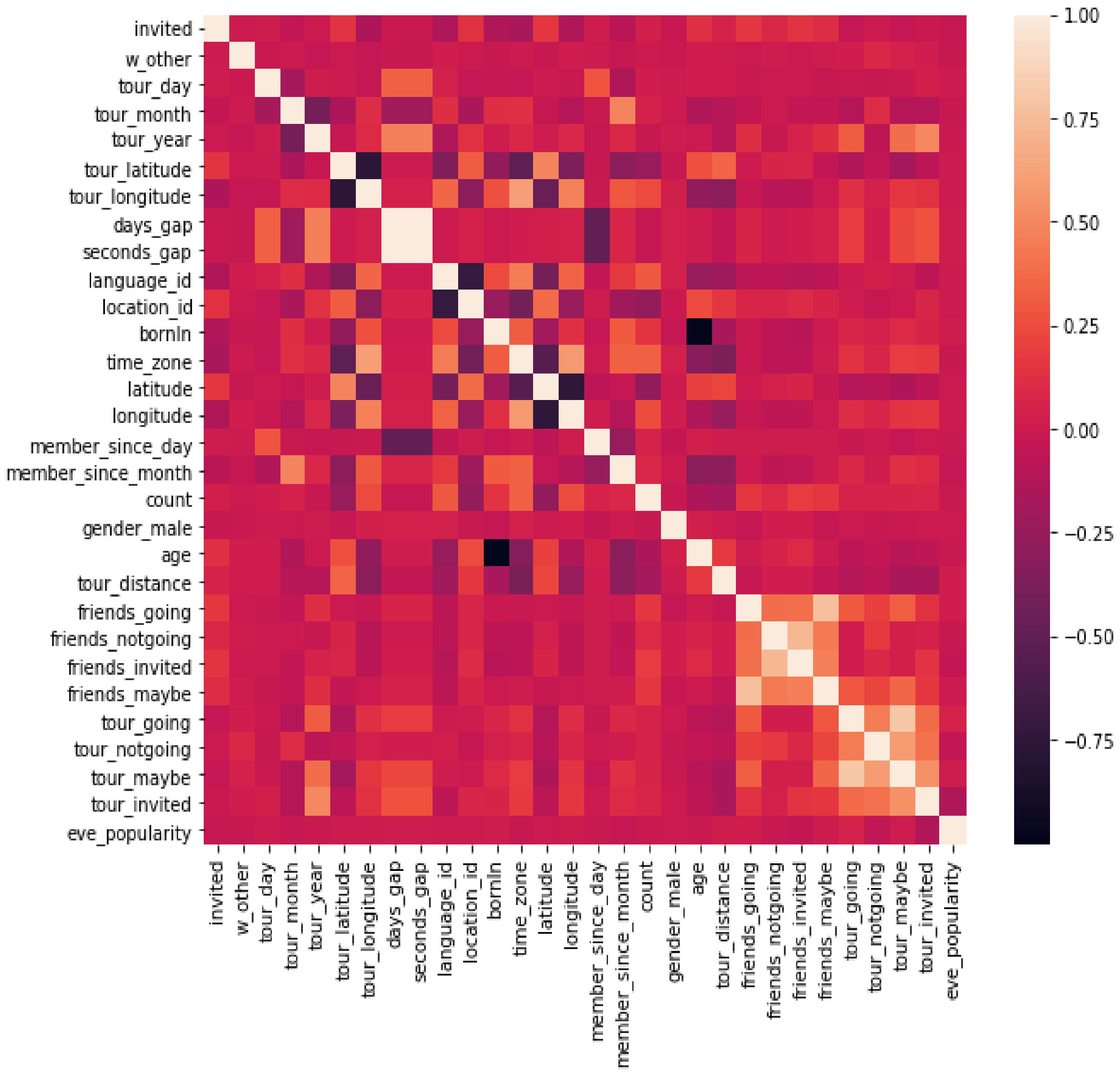
Hence a total of **132** features are extracted from the given datasets. Which are extracted using Python Dictionary data structure and pandas library tools, and missing values are taken care as mentioned above, a summary of those features are:

1.Number of days gap between invite and tour date
2.Distance between tour and biker location
3.Total number of friends in the bikers network
4.Tour popularity (extracted as described above)
5. going, invited, not going, maybe
6.Day, month, year of tour date and member since date
7.Total bikers going, maybe, not going, invited to a particular tour
8.Latitude and longitude of tour and biker
9.Year of birth of biker
10.Language Id of the biker
11.Location Id of the biker
12.Gender of the biker
13.Time zone of the biker
14.Age of the biker
15.w1, w2, ... w100, wother: Same as mentioned in tour data set

There are few auxillary features which are taken as a product of above features

Correlation Matrix

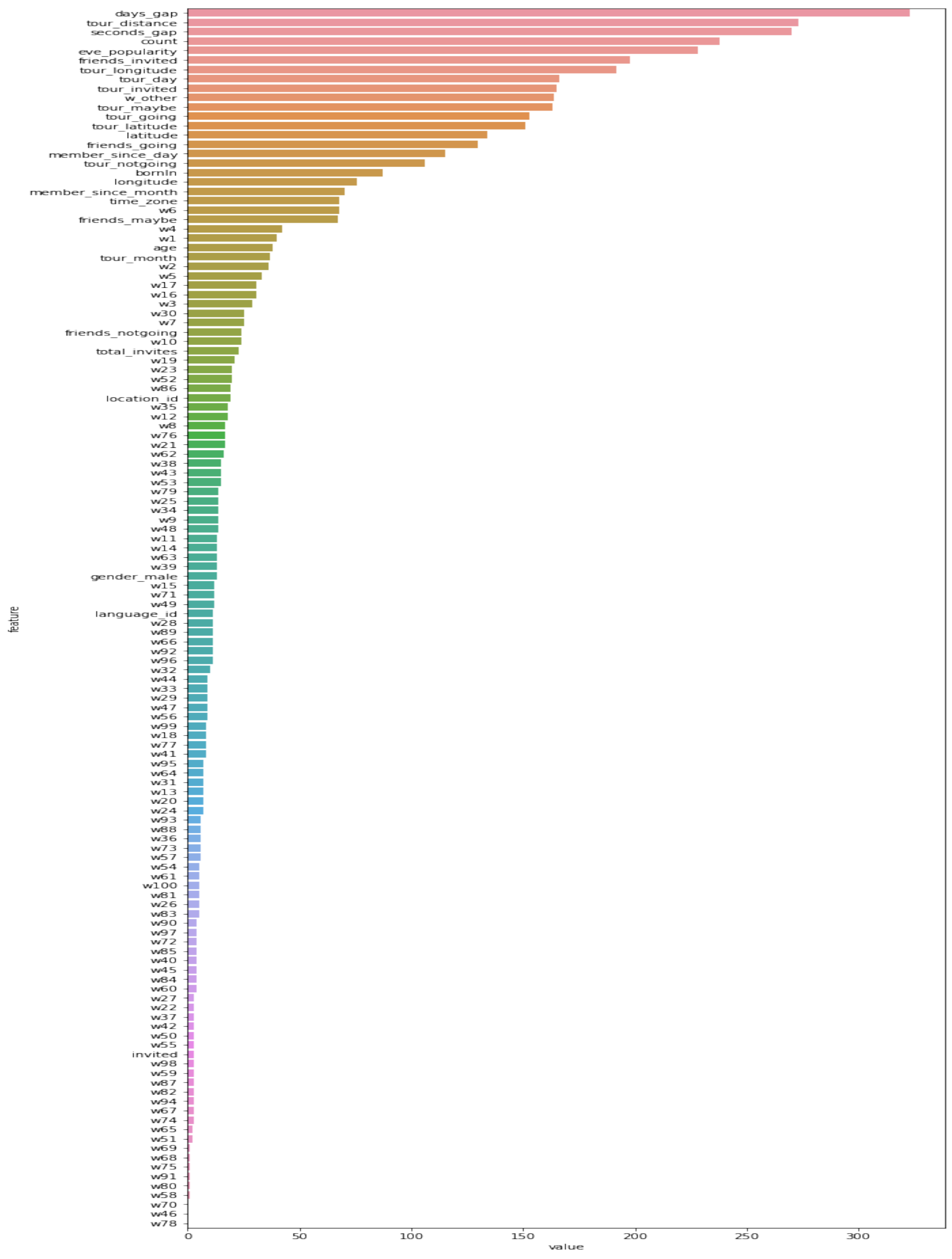
Following is the correlation matrix between all the features(except w1..100) which was useful in selecting and modifying the features.



(a) Correlation Matrix

Feature Importance Visualisation

Following the feature importance matrix, we can notice that features extracted days-gap, tour distance, count(total number of friends) were more important



(a) Feature Importance Plot

MODEL BUILDING

The dataset consists of a noisy mixture of continuous valued categorical columns. Gradient boosting machines are especially helpful in this scenario since they are agnostic to the type of the features to some extent and can learn to classify linearly inseparable data better than other models in the same scenario. Hence, we decided to use **XGBoost** and **LightGBM** as our models.

Training

Feature engineered training data matrix was obtained by removing likes, dislikes and some other columns which were not necessary and likes column was used as target label because using dislike as target label produces highly imbalanced dataset resulting in poor model. From this dataset **one-fourth** was kept aside to evaluate model after tuning and remaining data was used for model training as this will reduce overfitting.To choose the model and hyper-parameter combination we performed stratified k-fold cross-validation with **n-splits=3**.

Hyper parameter tuning

For base estimators XGBoost and LightGBM we experimented with several values for parameters learning rate, num leaves, n-estimators, depth etc...Optimal combination was found out using gridsearchcv along with stratified k-fold cross-validation with n splits=3. Since the dataset we have is an imbalanced one , using accuracy as performance metric is not sufficient. We used average precision as scoring criteria for fitting the gridsearch algorithm.

Best Settings

We have summarised the parameter settings we got through fitting of gridsearch model with stratified k-fold validation in below table.

Hyperparameters	XGBoost	LightGBM
N estimators	120	200
Learning rate	0.05	0.05
Depth	8	16
Num leaves	16	25

Table 1: Tuned Model

Evaluation on Validation Data

Models trained with above hyper-parameter values were evaluated on **one-fourth** of data we set aside for this very purpose.We computed model accuracy, **MAP** and **AUC ROC** score over validation data. Results are presented below.

Metric	XGBoost	LightGBM
Accuracy	0.734	0.766
MAP score	0.781	0.736
AUC ROC score	0.776	0.765

Table 2: Tuned Model

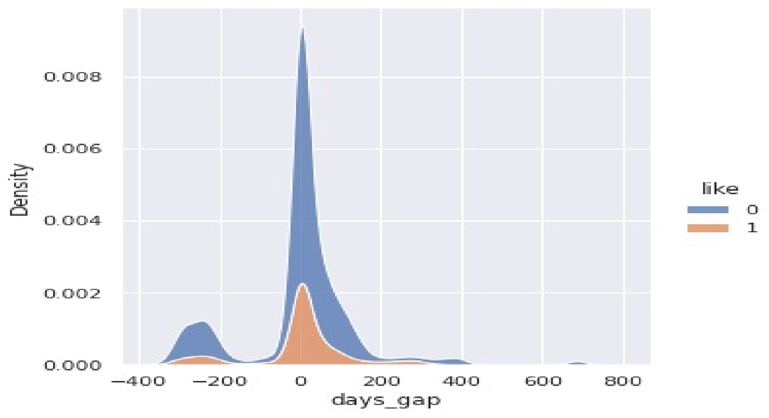
Generating Submissions

While testing 3 models are generated as we are using 3-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 three probability scores are averaged and then the tour ids are arranged in descending order. The results are written in a csv files and submitted.

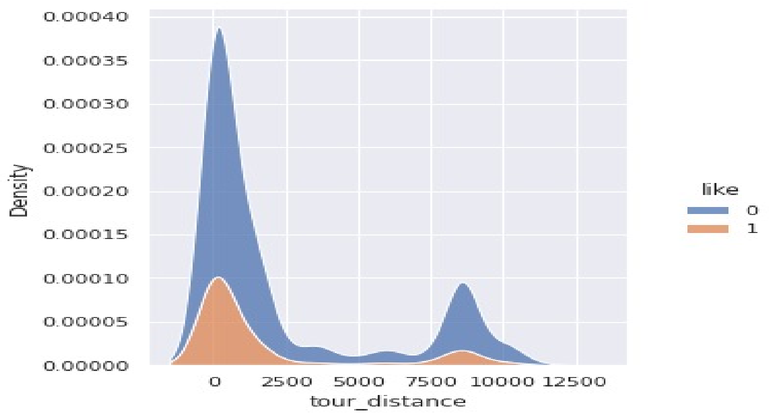
OBSERVATIONS

- The extra features extracted total likes, total dislikes given and mean average precision of 0.95 on the one fourth validation size, but this model performs poorly on the test data.
- The reason can be that the tourId in test data might be completely different from the tourIds in train data, in which case such rating system overfits for train and doesn't improve on test.
- Tour popularity feature which was used gave us a jump to 0.75 mean average precision compared to the model without it which was at 0.73 on 30 percent test data, but on full test data popularity model gave 0.69 whereas model without,gave us 0.71 mean average precision.
- This makes sense because this feature was too refined which increased the risk of overfitting the data. But in reality its difficult to make the conclusion unless we see the entire dataset.

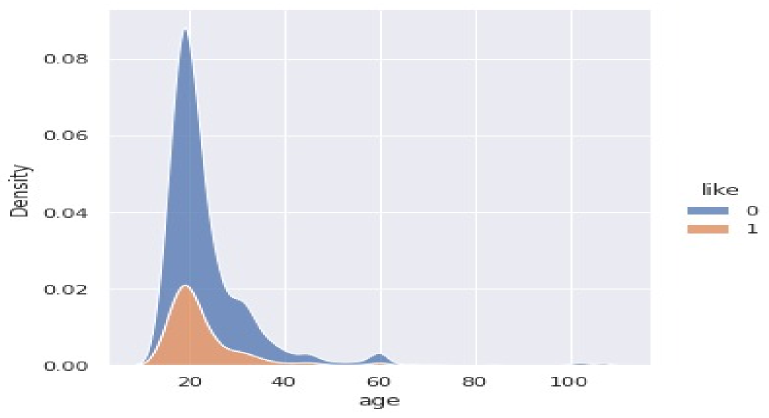
Few density plots are shown below to get intuitive feeling and to decide its importance.



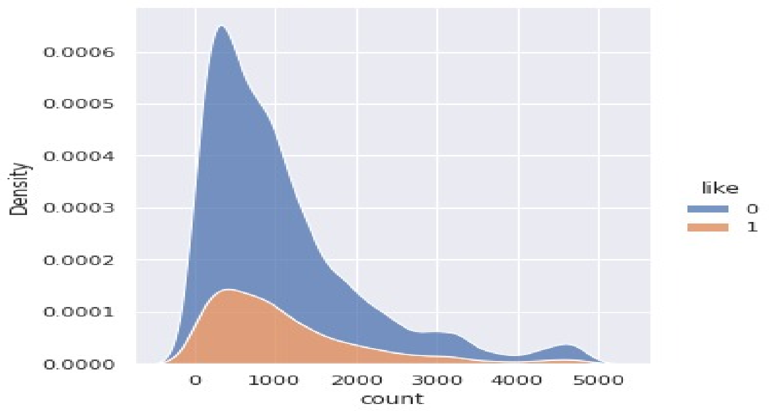
(a) days gap density



(a) tour distance density



(a) bikers age density



(a) No. of friends density

Conclusions

Through this data contest we were able to get a better understanding of the applications part of the concepts taught in class. We gained experience in searching for various models that can be used to solve a real world problem. We realized the importance of ensemble models in boosting their predictive performance.

From the change in public to private leaderboard experience and from our own observations we understood that extracting complex features or making complex models can increase the risk of overfitting the data and deriving too many features from a set of features can result in decreasing its importance (which can potentially enhance our model).

Through feature engineering, building model and tuning hyperparameters we learnt to resolve the issues of overfitting, deriving useful features and class imbalance which are generally observed in real world problems.

The performance could have definitely been improved with more creative features and models which we aspire to learn from other models and implement in future problems.