Airbnb New User Bookings

Udith Sai Manda (IMT2018081) Pranav Kumar (IMT2018020) Naveen Kumar (IMT2017029)

I. Introduction

In this busy world, everyone is busy with their own works and wants to take break. So many are planning to go for trips but many lack time or unable to decide the destination. And Airbnb wants to predict the region of booking based on previous data available. This helps Airbnb in terms of more bookings as content will be in favour of the customer interests. New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. Predicting the region of booking will help provide better content to the users with regard to their choice of destination, decrease the average time to first booking, and better forecast demand, thus benefiting both the customers and Airbnb.

II. Dataset

In this challenge, we are given a list of users along with their demographics, web session records, and some summary statistics. You are asked to predict which country a new user's first booking destination will be. All the users in this dataset are from the USA.There are 12 possible outcomes of the destination country: 'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL','DE', 'AU', 'NDF' (no destination found), and 'other'. Note that 'NDF' is different from 'other' because 'other' means there was a booking, but is to a country not included in the list, while 'NDF' means there wasn't a booking. The data in the train set dates back to 2010, while the test set consists of users joining after 01/04/2014. The sessions data dates back to 2014. There are 5 csv files data. Those are age gender bkts.csv, countries.csv, sample submission NDGC.csv, sessions.csv, train.csv and test.csv. In Train data, there are 16 columns and 170138 datapoints. In test data, there are 15 columns and 43315 data points.

III. Pre-Processing

After some observations through various plots and tables, we got an idea about the data structure. There are many null values in some of the columns. Those are date first booking, gender, age, first affiliate tracked, first browser. There is no use of date first booking as we need to predict the destination. For gender and first browser 'unknown' values, we changed that to NaN values. Then for age, the minimum and maximum are 1 and 2014 respectively so we filtered the age between 18 and 100. Then for unknown values we changed to mean of the age.

IV. Feature-Engineering

There are 16 columns in training data. Those are id, date account created, timestamp first active, date first booking, gender, age, sign-up method, sign-up flow, language, affiliate channel, affiliate provider, first affiliate tracked, sign-up app, first device type, first browser. In these features, some features are not necessary for our prediction of destination like date-first booking and country destination. As date-first booking has more percentage of NULL values which show that many who checked the Airbnb did not book so for prediction this feature has less influence. And we need to predict so country destination so we will separate target column in a different data frame.

We have two dates columns. Those are date account created, timestamp first active. For better classification we divided the dates into day,month and year for both. For date account created, dac day, dac month, dac year. For timestamp first active, tfa day, tfa month, tfa year.

From the analysis of data, we got new features. Those are session count, average seconds, long sessions count. Session count gives us the idea of how many sessions are there and with this we can use this for classification. Average seconds shows the average of time spent in each session of each member and this shows the interest of each interest in finding a destination. We can count this instead of user's one time visit. long sessions count are the number of sessions a user is present with more than 30min of duration. This shows how the user is more interested to find destination.

V. Encoding

After the necessary data pre-processing and feature engineering is done, we split xtrain and y to xtrain xtest and ytrain, ytest to test the different classifier and select the best one. For both X train and X test we applied Standard Scalar for standardizing the features i.e subtracting the mean and then scaling to unit variance.

We identified what are the categorical features in all the features. The categorical features are 'affiliate channel','affiliate provider','first affiliate tracked','first browser','first device type','gender','language',sign-up app',sign-up method',sign up flow'. We applied one hot encoding for these features so that we can have better classification at last. Then for our target feature i.e country destination, for our better classification we did label encoding for the country destination feature.

VI. Model Selection

We experimented with quite a few models , which were mostly tree-based algorithms because in the PCA analysis we could see that there was no clear linear boundary in our target column. We need to select the best classifier, We think

Ensemble methods work best. Some of the models that we tried for this project:

1) Logistic Regression: The first model we used is logistic regression classifier.Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam,..etc. Logistic regression transforms its output using the logistic sigmoid function to return a probability value. With this classifier, we got a accuracy score of 0.5781121429411074. But we weren't so satisfied with the score.
2) Random Forest Classifier: From LR, we moved on to Random Forest Classifier. Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity.Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. With this model we got a score of 0.6210473727518514 and we still want to improve the accuracy score.
3) CatBoost Classifier: CatBoost is a recently open-sourced machine learning algorithm from Yandex. It can easily integrate with deep learning frameworks like Google's Tensor Flow and Apple's Core ML. It can work with diverse data types to help solve a wide range of problems that businesses face today. To top it up, it provides best-in-class accuracy. Compared to CatBoost we got better accuracy score with XGBoost. We got accuracy score as 0.67 but overall score was just 0.70. So we went with XGBoost model.
4) XGBoost Classifier: At the end this was model through which we got best score. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now. With this model we got accuracy score of 0.6263077465616551.

We tried even other classifiers like Light gradient Boosting machine,Decision tree.. etc. But at last we got better result with XGBoost after hypertuning.

VII. HyperParameter Tuning

After selecting model, Hyper tuning of model is a impaortant task in getting more accuracy because there are many parameters for each algorithm. And every algorithm is a very complex algorithm so we need to get perfect parameters.

For hypertuning, first we tried changing parameters manually with different methods like XgBoost, CatBoost, LgBoost... But the score varied only between 0.71 and 0.80 but with XgBoost we got 0.81 as score.

Then with the help of GridSearchCV, we did the hypertuning automatic with the help of this. GridSearchCV is a function that comes in Scikit-learn's(or SK-learn) model selection package.This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters. GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the Cross-Validation method. Hence after using this function we get accuracy/loss for every combination of hyper parameters and we can choose the one with the best performance.

After using GridSearchCv for XgBoost, we got the tuned parameters:

XGBClassifier(base$_score = 0.5, booster =' gbtree', colsample_bylevel = 1, colsample_bynode = 1, colsample_bytree = 1, gamma = 0, gpu_id = -1, importance_type =' gain', interaction_constraints ='' , learning_rate = 1, max_delta_step = 0, max_depth = 1, min_child_weight = 1, missing = nan, monotone_constraints =' ()', n_estimators = 10, n_jobs = 0, num_parallel_tree = 1, objective =' multi : softprob', random_state = 0, reg_alpha = 0, reg_lambda = 1, scale_pos_weight = None, subsample = 1, tree_method =' exact', validate_parameters = 1, verbosity = None)$

And the accuracy score is 0.72.

VIII. Results

- The accuracy score for XgBoost Classifier after hyper tuning is 0.615.
- The overall Score in public leader board is 0.82128.
- The overall Score in private leader board is 0.82339.

IX. Conclusion

We would like to conclude that we were able to come up with an efficient model to predict a country destination. Such projects have a potential scope to make the Airbnb more interesting and helpful for both Airbnb and customer. In the present busy world, prediction machines are so helpful.

Acknowledgment

We would like to thank our professors Professor GS Raghavan and Professor Neelam Sinha for imparting us the knowledge in the field of Machine Learning and giving us a great headstart in this world of Machine Learning. Next we would also like to thank our amazing TA's who have been teaching us all the things that we have needed for the course and also for our project and the assignments and we would especially like to thank Tanmay Jain for his constant guidance and help in clearing all the doubts that we faced throughout the process.

REFERENCES

Here are the list of references which helped us in the process of doing project with information and syntax. We learnt a lot with the help of the these references and applied it on project.

REFERENCES

[1] https://catboost.ai/docs/concepts/r-training-parameters.html
[2] https://stats.stackexchange.com/questions/237523/how-to-threshold-multiclass-probability-prediction-to-get-confusion-matrix
[3] https://towardsdatascience.com/how-to-handle-multiclass-imbalanced-data-say-no-to-smote-e9a7f393c310
[4] https://builtin.com/data-science/random-forest-algorithm
[5] https://www.mygreatlearning.com/blog/gridsearchcv/
[6] https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/