Capstone Project

Machine Learning Engineer Nanodegree

Neethu Wilson
December 6th, 2018

Definition

Project Overview

Airbnb is an online home rental platform based in San Francisco that lets people list, find, and rent short-term lodging. In recent years, Airbnb has evolved its peer-to-peer model to give hosts the technology tools they need to run a seamless, sophisticated operation. Using Machine Learning on the available data, can be used to generate new insights to improve customer satisfaction, adding new customers etc. The project that I will be working is posted by Airbnb in Kaggle to predict New User Bookings.

Problem Statement

New users on Airbnb can book a place to stay in 34,000+ cities, across 190+ countries. By accurately predicting where a new user will book their first travel experience, Airbnb can share more personalized content with their community, decrease the average time to first booking and better forecast demand.

In this project I will be applying Machine Learning classification models to predict the new user bookings using the data provided here https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings.

Metrics

As the problem is taken from Kaggle, evaluation metric is already known. The evaluation metric for this competition is NDCG (Normalized discounted cumulative gain) @k where k=5.
NDCG is calculated as:

$$DCG(k) = \sum_{i=1}^{k} \frac{2^{rel(i)} - 1}{\log 2(i+1)}$$

$$nDCGk = \frac{DCGk}{DCGk}$$

where reli is the relevance of the result at position i.
IDCGk is the maximum possible (ideal) DCG for a given set of queries.
All NDCG calculations are relative values on the interval 0.0 to 1.0.

For each new user, you are to make a maximum of 5 predictions on the country of the first booking. The ground truth country is marked with relevance = 1, while the rest have relevance = 0.
For example, if for a particular user the destination is FR, then the predictions become:

- [FR] gives a NDCG=21−1log2(1+1)=1.0
- [US, FR] gives a DCG=20−1log2(1+1)+21−1log2(2+1)=11.58496=0.6309

Analysis

Data Exploration

We will be using the below 2 files provided by Kaggle.

- train_users.csv : users Imformation.
- sessions.csv : web sessions log for users

There are 213,451 records in train_users file and 10,567,737 records in sessions.csv. Our dependent variable is Country_Destnation and is present in Train_user file. So both the files could be merged using the **'id'** from **train_users** and **'User_id" from Session file.**

First let us find the null values in each files.

train_users:

id	0
date_account_created	0
timestamp_first_active	0
date_first_booking	124543
gender	0
age	87990
signup_method	0
signup_flow	0
language	0
affiliate_channel	0
affiliate_provider	0
first_affiliate_tracked	6065
signup_app	0
first_device_type	0
first_browser	0
country_destination	0

sessions:

user_id	34496
action	79626
action_type	1126204
action_detail	1126204
device_type	0
secs_elapsed	136031

Session file has many records with no user_id, So they can be removed right away. When we are predicting 'New booking', "date_first_booking" from train_user will always be null for the test data. So that column can also be dropped.

Lets get the basic information of Age columns. Min and Max value of Age is 1 and 2014, so there are some invalid data in age column. Someone with age 1 or 2014 (looks more like a year, than age) cannot book a destination. So the outliers have to be removed.

	age
count	125461.000000
mean	49.668335
std	155.666612
min	1.000000
25%	28.000000
50%	34.000000
75%	43.000000
max	2014.000000

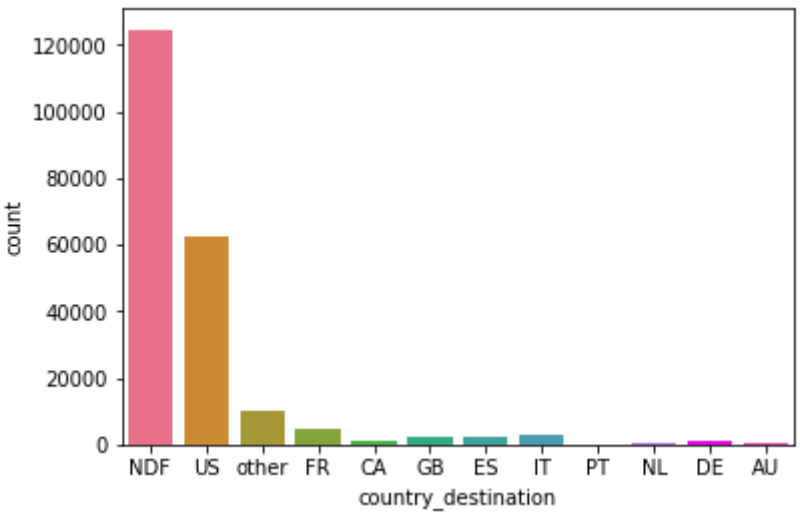
Further, the correlation between 'date_account_created' and 'timestamp_first_active' is 99%.So we can drop one of the 2 columns.

As there the competion is closed now. We are using only the train_dataset for the project. A part of training data is kept aside for testing. Also the evaluation using 'NDCG' is done as part of the coding itself. The function of NDCG was taken form the Kaggle : 'https://www.kaggle.com/davidgasquez/ndcg-scorer'

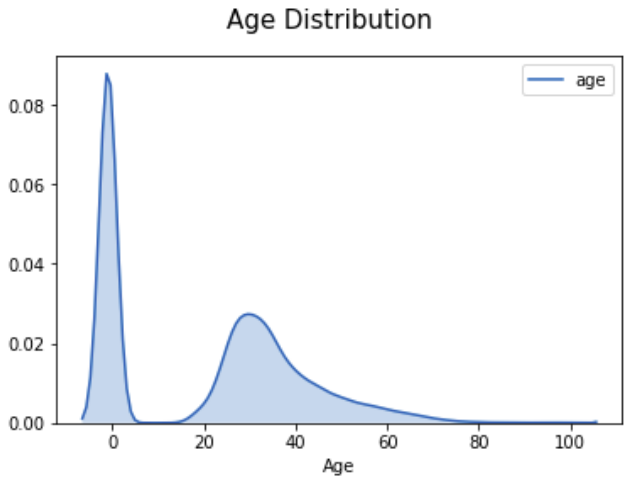
Exploratory Visualization

country_destination distributon

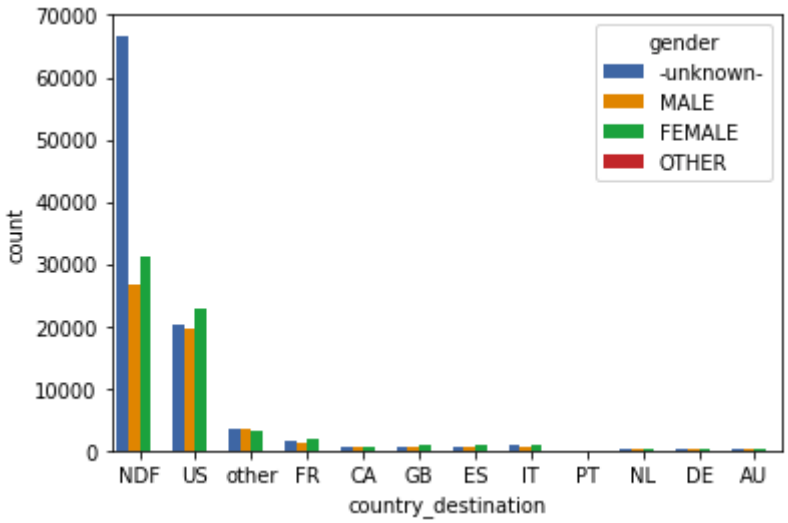
Around 50% of the people have not booked a destination.



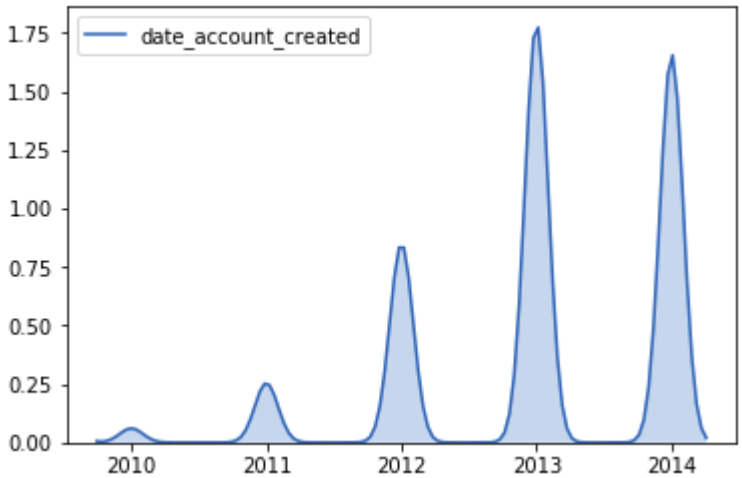
Around 40% of the data has Invalid/missing data. All invalid ages and missing data is replaced by '-1'.



It looks like people who have not provided Gender are not likely to book a destination.



Number of account creation has increased from 2010 to 2014



Algorithms and Techniques

As per the given problem statement, the model needs to predict the first locations where a new user of Airbnb will book. The location will be one among the 12 countries provided. So this is a multi-class classification problem.

Decision trees are best suited for a multiclass supervised learning.It breaks down the data into small datasets using decision rules.They can handle numerical and categorical data, missing data, along with multiple target classes

Rather than using a simple decision tree, we will be using ensemble learning methods to produce improved results. So here we will be using random Forest and XgBoost using GridSearchCV.

Random Forest

Random forest uses bagging ensemble technique. Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.A single decision tree may be prone to a noise, but aggregate of many decision trees reduce the effect of noise giving more accurate results.

XGBoost with GridSearchCV

Extreme Gradient Boosting (xgboost) uses Boosting ensemble technique. In this algorithm also, many weaker decision tree are are created before finalazing the best tree.
In Boosting the tree are made sequentially. The new trees will learn from the mistakes made by the previous ones making it better during each subsiquent run.
It uses gradient boosting and is a regularized model formalization to control over-fitting, which gives it better performance.

Methodology

Data Preprocessing

Train_User Processing

- date_account_created and timestamp_first_active
  - As we can see in the data exploration,the corelation between these 2 feilds are 99%. So I have decided to go with only date_account_created . But as we dont have timestamp in date_account_created, hours is taken from timestamp_first_active. The data was then split into year, month,day etc
- Age
  - This column had a lot of outliers and missing values. So I have decided to update ann the Nan as -1. Also all ages less than 14 and greater than 100 is marked as -1.
- Missing Age and Gender
  - From the dataexploration, t was found that records with Missing Age and Gender had a corelation to country destination. So I have created a new column,Missing_Age_gen, which is marked as 1 if the data is missing, else 0.

Other Categorical column were encoded, so that we could apply the model.

Session Processing

- user_id
  - In session there were many records with user_id as ‘Nan’ and all of them were removed. So once it is removed, the remaining userid’s were only 40% of the number of user_id in Train_user file.
- secs_elapsed
  - secs_elapsed has a wide range of values. Some values even extend to days, so there is good chance that it might be an outlier. So a new column was created to cap the max value of secs_elapsed. The cap was set to 2nd quartile. secs_elapsed and secs_elapsed50 were then aggregated at user_id level. I also add a column count, which had count for each visit/action on airbnb site by an user.

Now these 2 files were joined on user_id and the values were normalized using MinMaxScaler. The total number of column after data preparation is 50.

Implementation

Random Forest Classifier

The parameter used are estimators =10, min_samples_split=50, criterion = 'entropy'. I have stopped the min_sample_split at 50 to avoid overfitting. I validated the data using validation subset that was taken from the data. Once model tuning is done, I ran the same in the test subset taken from data set. Below are the scores obtained.

```
Valid       :  0.8230682027437065
Test        :  0.8244925060328597
```

XG Boost Classfier with GridSearchCV

The Data was trained using parameters max_depth of 8, learning rate of 0.1 , objective ‘ndcg@5’ etc. GridSearchCV was run on the model with 10 folds running 2 jobs in parallel. A custom scoring,ndcg, which is based on the ‘evaluation metrics of the problem’ was ussed. After training and validation below scores were obtained.

```
Valid :  0.8274657737461266
Test :   0.8286871702696362
```

Results

Below is the final Score for the Project.

```
Valid :  0.8274657737461266
Test :   0.8286871702696362
```

In the proposal it was suggested that we benchmark the results by uploading the results on the Kaggle leaderboard. During the course of the project it was discovered that Kaggle had stopped accepting further submissions to the leader board. Therefore, we are unable to conclusively rank our project.

However, by comparing the current scores manually with what have been uploaded it is possible to ascertain that the results are in the same ballpark of the stated objectives.

Justification

The major requirment of any algorithm is that it should train right, so that the score does not goes down when we run on a testing data. In this project, both testing and the training scores looks pretty similar. In that way the algorithms used were good.

Conclusion

Summary

We have Completed the project using XGBoost and Grid SearchCV. The model has been trained on the training dataset and gave a decent result for both the validation score and Training score.

Learnings

I learnt how make sense out data, how to handle missing data and outliers etc. Also learned the importance of using visualizations, as they could provide better insight of data. This also helped me in trying new alogorithm and try hyper parameter tuning for XGBoot and GridSearchCv

Limitations

The training would have been better, if we could have used some more information from session file. But there are more than 300 actions and my system was not cable of handling such huge data.