

Project report: Airbnb New User Booking Destination Prediction

CEBD 1260 Winter 2021 Michal Velharticky

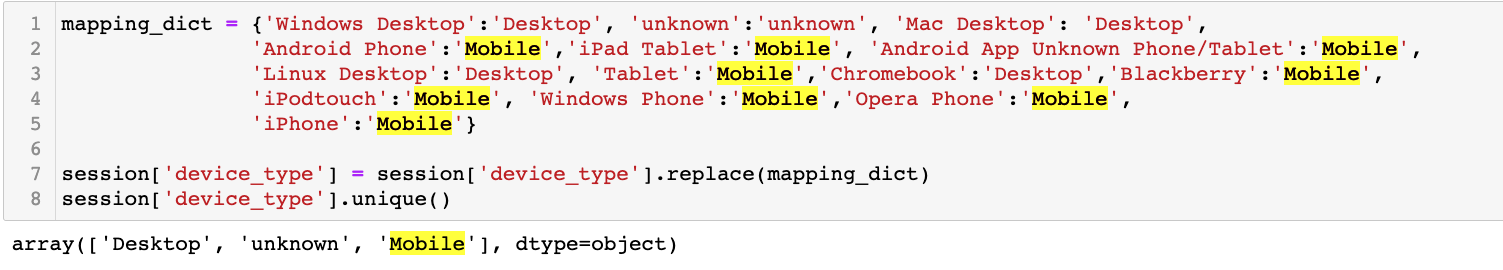
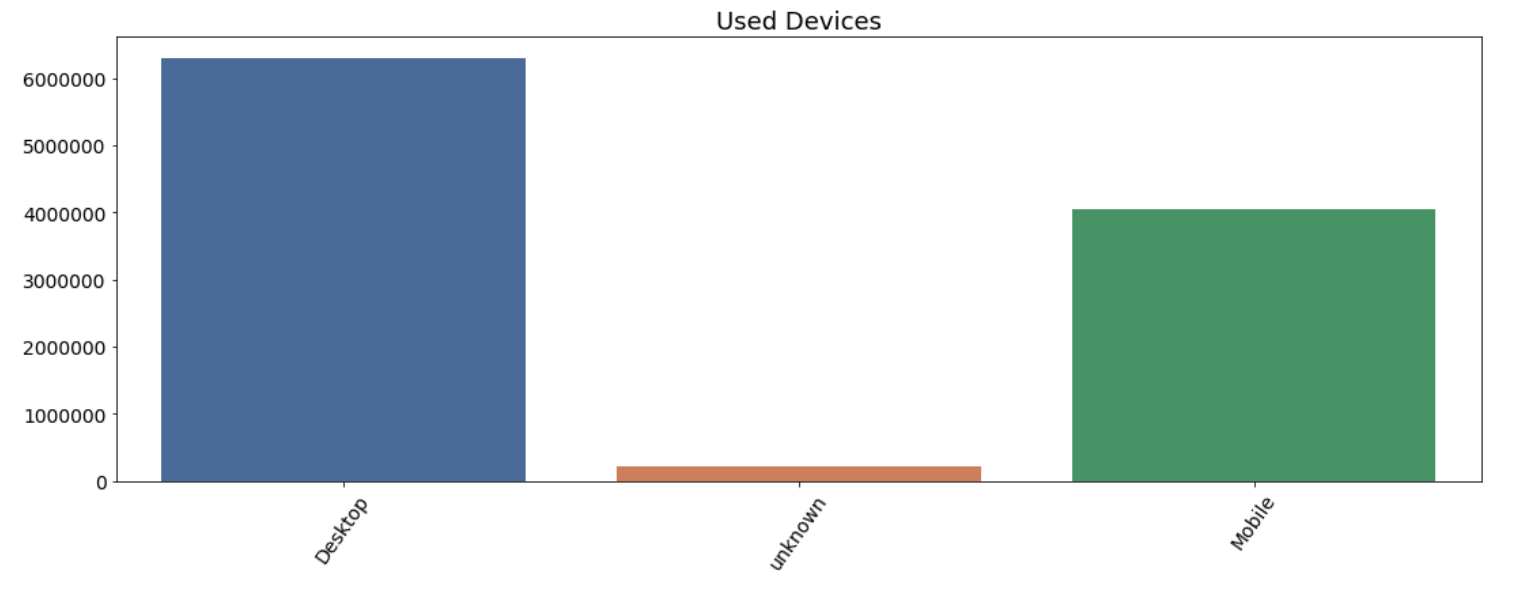
In this project, the goal is to build a model to predict which country a new user is most likely to book. The data provided for the project are from Airbnb online marketplace. Airbnb is the world's largest online platform for booking a vacation stay at a location of choice. It is accessible through a website or a mobile app. Its versatility has become very popular in recent years. It includes both from property providers to renters. Not only holiday seekers but travelers for business alike have the possibilities to choose from an endless variety of properties to choose from. The list includes options from basics like rooms, apartments, and goes all the way to cottages and whole properties. The company itself does not own any rental property. The company’s model is built on collecting provisions from stays at the renter’s property in exchange for listing their place, its availability, and also the online services such as payments and renters profiles for security measures.

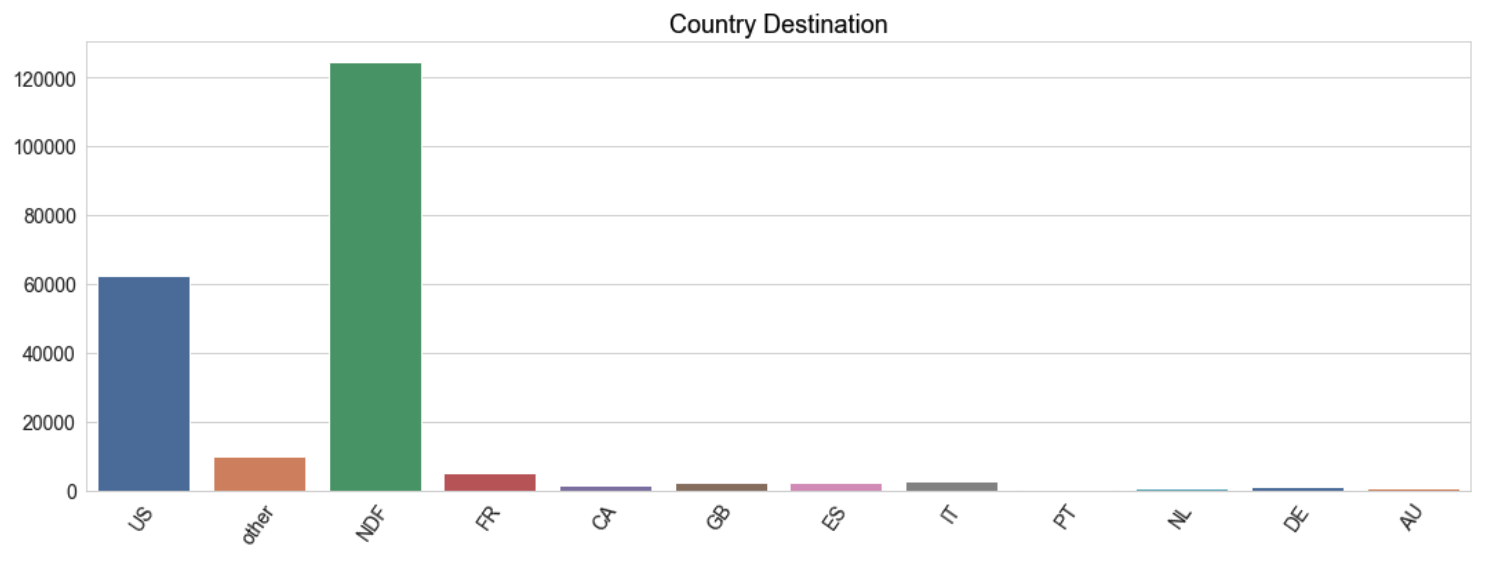
* Results   
  The early runs of the model (using LGBMClassifier) were at multi\_logloss of 0.785962, and valid\_1's multi\_logloss of 0.945887. With changes in the parameters, increasing the number of estimators to 10000 and min data in leaf to 1000, the score improved to training's multi\_logloss of 0.854425 valid\_1's multi\_logloss of 0.951687. 

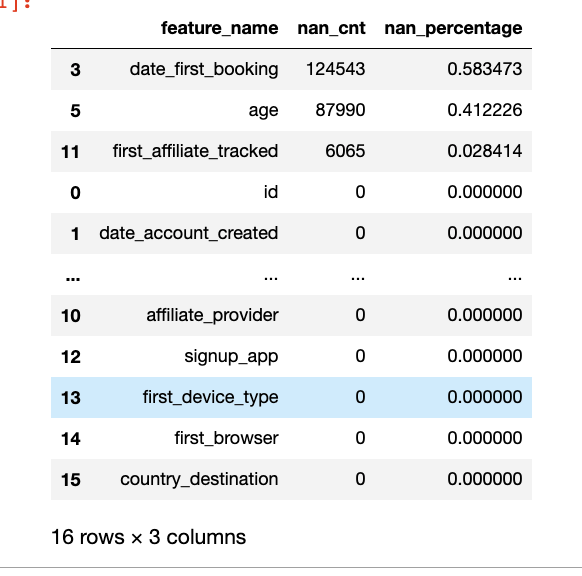
The final score of multi log loss improved to 0.943668 with valid\_1's multi\_logloss at 0.992049 and Coefficient of determination R^2: 0.8708503344090663

* Findings  
  Increasing the number of estimators and min data in the leaf really boosted the performance, while the number of leaves was kept at considerably low and not going over the recommended num\_leaves = 2^(max\_depth), which could lead to overfitting.

**System pipeline**

* Data preprocessing 

The data consist of two files, ‘session’ and ‘booking’. The session file contains data that is related to users' activity within the marketplace, that was not very clean and required a decent amount of work. In the booking file, there is data that was fairly clean but contained some records that were not corresponding to reality with other data in the file. There are mainly categorical values, three features with dates and elapsed time.   
The target columns are also categorical values and contain 12 country destinations.

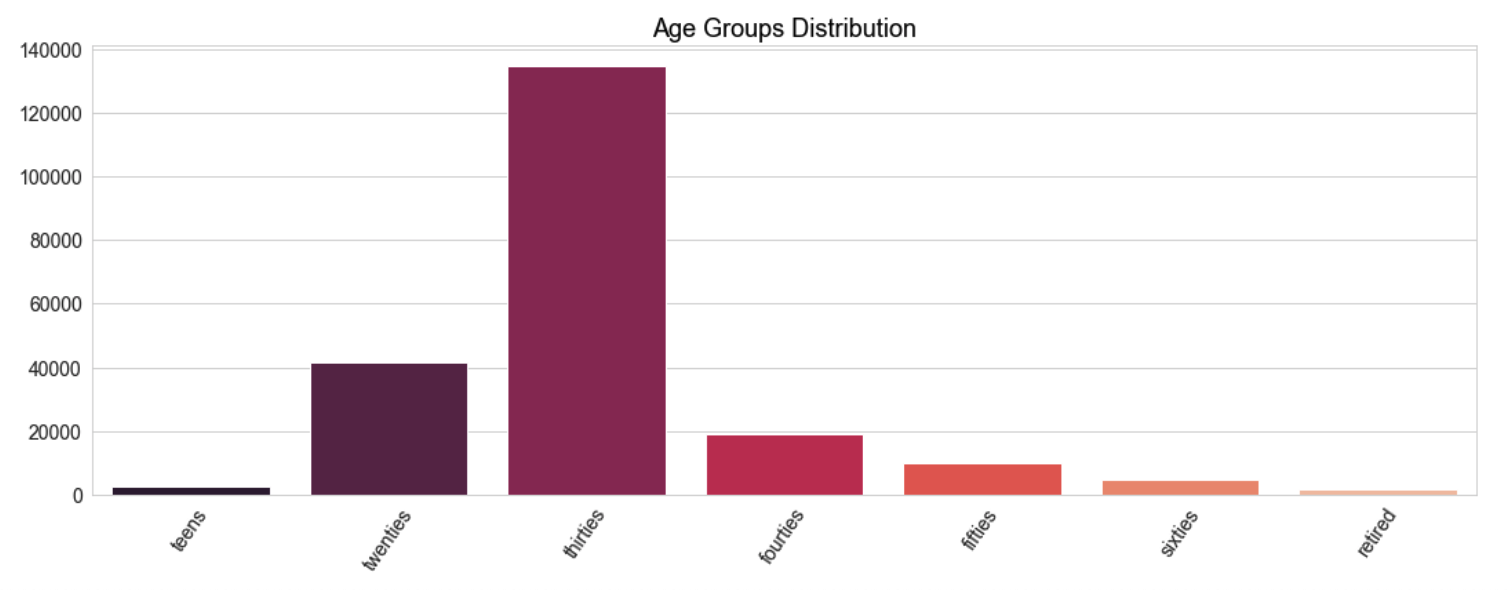
  
To start with the booking file, it is consisting of 213451 rows and 16 columns. As seen in the image we can see the feature “DATE FIRST BOOKING” was the column with the highest amounts of missing values. Those were filled with zeros to be later filled by using a function interpolate(“pad”), which fills the missing value with the closest existing value. This I thought worked perfectly where in this case it can provide some level of continuity.

The “AGE” feature contained some data that was unrealistically high, in thousands. Some values were under the legal age of 18 and those over 100 years were filled with zero values. I decided to fill those with median values rather than the traditional mean. My focus was on keeping the age data distribution as much without a change as possible.   
  
The values of “DATE ACCOUNT CREATED” and “TIME FIRST ACTIVE” are in most cases identical.   
The remaining features in this file are categorical and were mainly subject to simplifying but also cleaning. This would be done by creating a dictionary, which would also handle the missing values where they would fall under the unknown category. It also helped to reduce the amount of less significant categories. Which is either set as unknown or put under one type/category.   
This was decided by rationale and according to the type of features. It allowed for a cleaner representation of values.

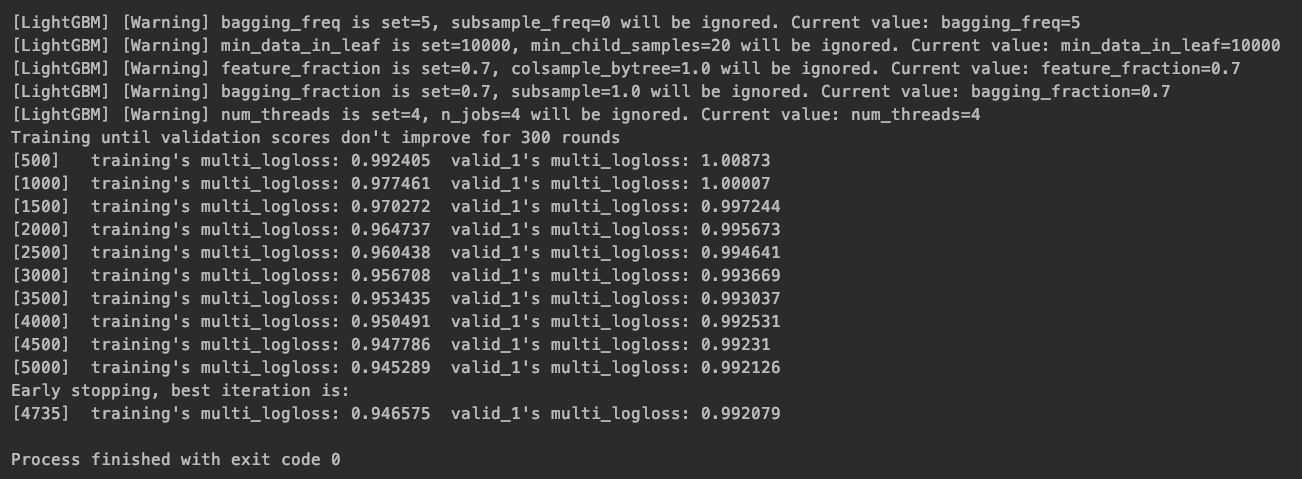
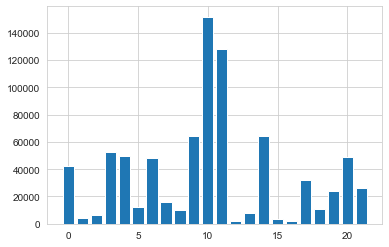
The “session” has 10567737 and 6 columns. The file was mainly handled in the same way. Missing values were filled with an unknown value that was in most cases already part of the data. It could perhaps help if the data generation for features “ACTION TYPE” and “ACTION DETAIL” was done in a more structured way and allowed for easier handling. This could increase the importance of this data and could provide additional understanding of users' activity. We need to consider if we need this level of detail in the case of the mentioned two features. The final steps were grouping the data by “ID”, aggregating numbers of unique action types and used devices, and creating calculations on “ELAPSED TIME”. Which worked as preparation for joining the main file but indeed as feature generation. During the cleaning process analysis on each feature is done and followed by visualization for a better feel of the data.

Both files are joined using an inner join method which unfortunately leads to a loss of rows in large volume. Where the booking was at (213449, 33) and session (135483, 11). This is due to many IDs not matching in both files. The main file ends up with 73815 rows and 43 columns for model processing.

In total there are 12 columns with categorical features that are encoded using LabelEncoder() to give the date significance rather than using the method of one hot encoding which I thought is not suitable for this case for the high number of categorical features. The date data types are dropped and the file is at 39 features for the model.

* Feature engineering  
    
  The idea of calculating usable data from the date values was not possible, it created a lot of negative values.  
    
  Taking into account trending locations or a search for a particular activity, for different age groups, I binned different ages into their decade categories. Activities or locations are linked with the seasons of the year   
    
  I used the data to generate days of the week, day of the month, day of the year, the month of the year and finally year. Where in the winter weekend might be the more likely time to book at a warm location. This also gave an idea to add seasons of the year, for its significance in booking a holiday stay.   
  
* Algorithms(models):  
  When selecting the assignment and working on the first theoretical assignment I was looking for which model to use and my consideration was XGboost as the web was full of its glory by people using it. I was convinced that it would definitely be the way to go especially since dealing with multi-class classification. During the class presentation of LightGBM, I changed my mind and went with LightGBM.

The performance was convincing so I stayed with it throughout the project and tuned parameters for higher performance.

* Methodologies  
  I split the data 65% of data for training, 25% for validation, and 10% for testing. The model performed better than the commonly suggested rule 70/20/10. Multi-class logarithmic loss function per class was used as the evaluation metric with the early stopping of 300 rounds.
* Results:   
    
  Coefficient of determination: R^2: 0.8708503344090663  
  R^2 Cross validation score:  
  [0.08227931 0.07888236 0.08916627 0.08166337 0.07215879]   
  Mean 5-Fold R Squared: 0.08083002065095013  
    
    
  Considering the feature importance plot, removing some features may improve the model. 

References:   
Yimin Nie - guidance, advice, troubleshooting, code samples  
<https://stackoverflow.com/> - mixed troubleshootings  
[https://lightgbm.io/](https://lightgbm.readthedocs.io/) - model setting and parameters  
<https://towardsdatascience.com/> - mixed articles, opinions and suggestions