**CAPSTONE PROJECT REPORT**

**Goal**: The goal of this Capstone Project was to demonstrate my learnings from the Springboard Data Science Intensive course by working on a complete Machine Learning problem.

**Business Problem**: This was a Kaggle competition hosted by Airbnb where the aim was to predict where a new user would book his/her first travel experience.

<https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings>

**Data Description**: Airbnb had provided a list of users along with their demographic data, web history and some summary statistics. The target variable had 12 different values - 'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL','DE', 'AU', 'NDF' (no destination found), and 'other'.

The user file had details such as the user id, date when the account was created, age, gender, timestamp of first activity, international language preference, type of paid marketing, device type, browser and a few other fields. For the web data, there were details about the type of action the user had taken along with the amount of time spent in seconds. Apart from this there were some summary statistics about the user’s age group, gender and destination country.

**Tools**: Python, Jupyter notebook

**Process**: To begin with I started looking at the “train\_users” file provided by Airbnb. This file had 213,451 samples in the data. I created a dataframe using pandas and explored the data by using the head, info and describe methods. I also looked at the distributions using the value\_counts method and created some bar plots using seaborn. This gave me a good understanding of the data in the file.

Next I created some functions to extract year, month, day and hour from a timestamp variable and year, month and day from a date variable. I had also created an indicator for a field called as “date\_first\_booking”. This seemed to be strongly correlated with the target variable.

My next goal was to impute missing values. For that I had to first create dummy variables for all the categorical variables to ensure that all the features going into the model were numeric. Once this was done, I created a couple of functions to impute missing values for categorical features and numeric features respectively.

I then looked at the web sessions data for the various users. This file has more than 10 million records. I grouped this file by user\_id, action\_type, action\_details and device type and created a summary field for the total number of seconds elapsed for a user. I merged this dataframe with the train\_users dataframe. I now had a total of 1,004,182 rows of data.

I created dummy variables for the categorical variables from the sessions data and also checked the data for any duplicate columns. Now I was ready to begin the modeling process.

**Model**: For the modeling process, I decided to use the LabelEncoder to transform the multiclass target into numeric levels. After this, I created my dataframe (y) for the target variable and another dataframe (X) for all the independent features that I wanted to test in the model.

I then split the data into training and test sets using a 70/30 ratio. I then defined a model after importing it from the relevant library. Once I had defined my model, I had used it to fit the model to the training data and score it on the training data. Then I scored the test data using that model and generated the prediction for the target on the test data.

To evaluate the model I looked at metrics such as the confusion matrix and classification report since it was a classification problem. I tried several classifications techniques such as Logistic Regression, KNN, SVM, Decision Tree, SVM, Random Forest, Gradient Boosting, AdaBoost and XGBoost.

I had used scalers and pipeline objects in some of my models. I had also tried hyperparameter tuning techniques such as GridSearch and RandomizedSearch. Some of these models took a long time to run based on the hyperparameters that were specified. I tried to tune the following parameters – learning rate, n\_estimators, max\_depth and max\_features.

For the baseline model since this was a multiclass classification problem, I set the target to the value with the highest frequency (country\_destination = ‘NDF’) for all the samples in the test data and generated the metrics. The Precision, Recall and F1-score were 30%, 54% and 38% respectively.

My best model was a Gradient Boosting model with a training score of 81% and validation score of 79%. The Precision, Recall and F1-score were 81%, 79% and 79% respectively. With the train\_users data my maximum validation score was 67%. But after adding the user sessions data and tuning the hyperparameters, the model improved significantly and the validation score went up to 79%. This model was much better than the baseline model.

I have created a summary report that lists the details of all the models that I had created:



The complete Python code for my final model can be found here:

<https://github.com/kunal786/springboard/blob/master/Capstone%20Project/Capstone%20Project%20-%20Final%20Model%20(Gradient%20Boosting).ipynb>