Rensselaer

2016

Airbnb New User Booking Forecast

Tech Fundamental For Analytics

kaggle

Sayan Majumdar

Rensselaer Polytechnic Institute

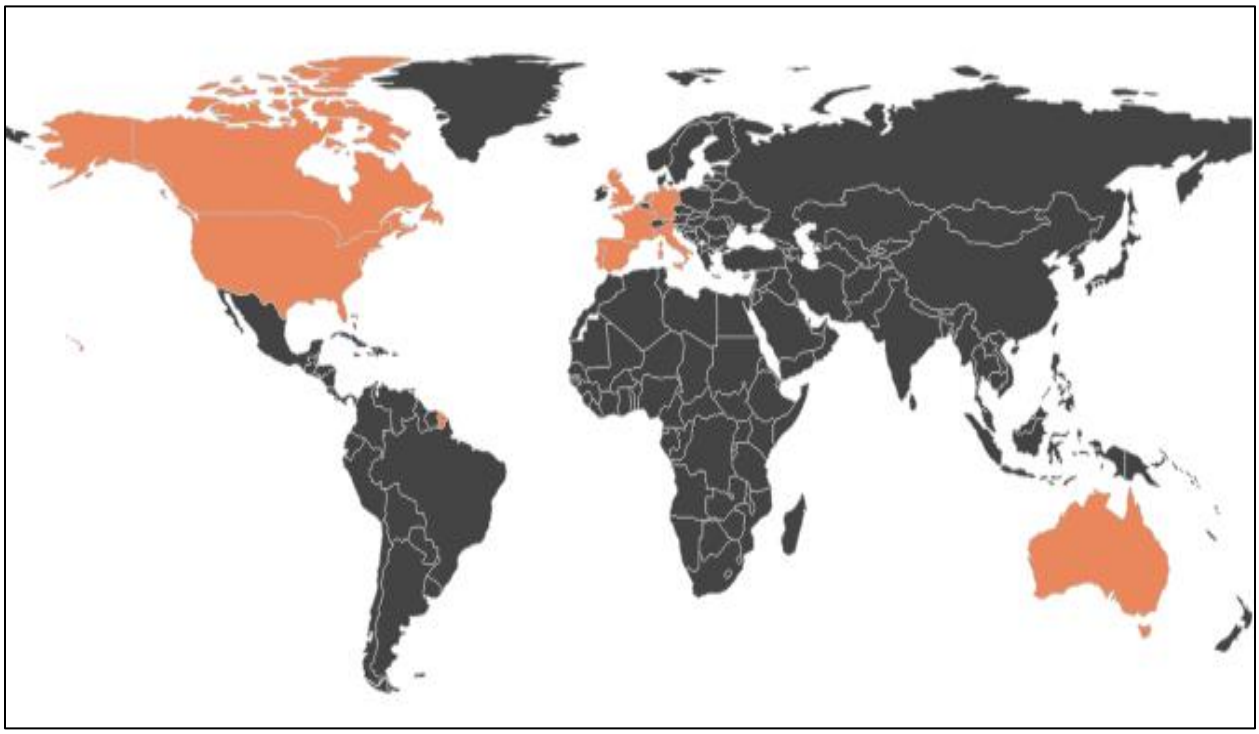
12/9/2016

Introduction:

Airbnb is a peer to peer online marketplace for renting and listing short term accommodation in residential properties. The company has achieved world-wide presence in a short span of time. The business model is based on dual earning from both the guest and host as a percentage fee of the total booking. It has over 2,000,000 listings in 34,000 cities and 191 countries. With such large geographical presence and global customer base, it becomes difficult for Airbnb to optimize marketing activities. The initial wait times for 1st time user to book an accommodation through Airbnb is comparatively high. Airbnb wants to channelize marketing initiative targeted towards the right customer segment.

Objective:

The objective of this business problem is to forecast the destination country for customers at 1st time booking. By able to forecast the destination country of 1st booking, Airbnb aims to implement Targeted marketing activities to the right Customer segment. Through this Airbnb wants to reduce average time to 1st time booking & better forecast demand.



Dataset

The dataset for this project is provided by Airbnb which contains a list of users along with their demographics, web session records, and some summary statistics. The whole dataset contains 5 csv files: train-users, test-users, sessions, countries, age-gender-bkts.

1) Train Users: The train data contains 213452 rows with the following 16 attributes:

•id • date-account-created • date-first-booking • gender • age • signup-method • signup-flow • language	• affiliate-channel • affiliate-provider • first-affiliate-tracked • signup-app • first-device-type • first-browser • country-destination • timestamp-first-active

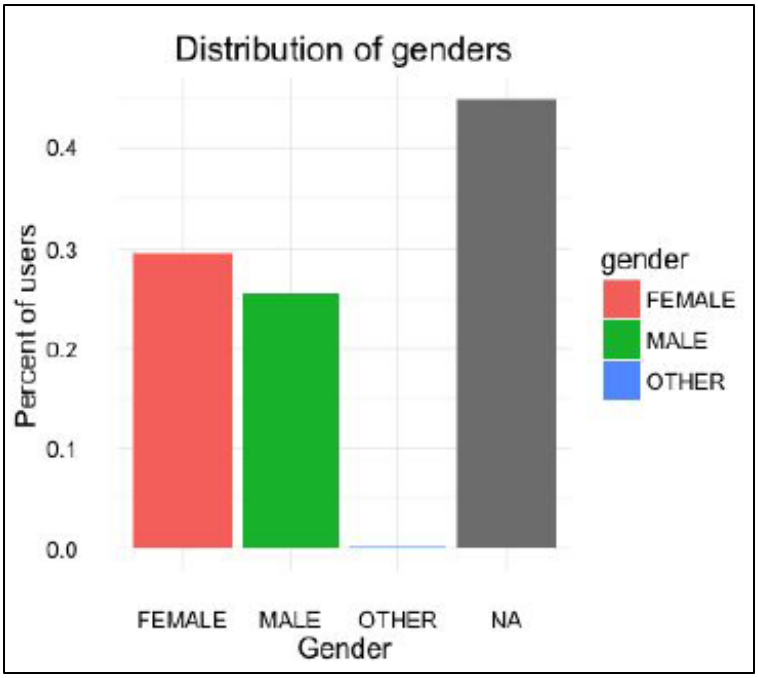
2) Test Users: This file contains 62097 rows with 15 entries. The country-destination attribute is not present in this file for obvious reasons.
3) Session File: This is a web session log record of users.
4) Countries: This file contains the geometric information of the different countries. It contains information on 10 countries on 7 properties.
5) Age Gender Brackets: This file contains statistics of users, age group, gender, country or destination.

It is to be noted that the train and test dataset are most important source of data. The train dataset is used to develop a model and the test data will be used to validate the model and check for the accuracy performance. The other files will give the peripheral information on users, countries and other attributes. A number of data are missing. Therefore, a proper cleaning procedure need to be implemented where the missing data's are suitable treated and the overall dataset are formatted as per the requirement of the models to be applied.

Exploratory Visualization

As we have seen in the previous section, the datasets provided by Airbnb can be used to perform exploratory data analysis. This will enable us to get an insight into the dataset, the business scenario and other information on how we can achieve the business objective. The datasets are fed into tableau in order to get a number of visualization by which we can draw a number of insights.

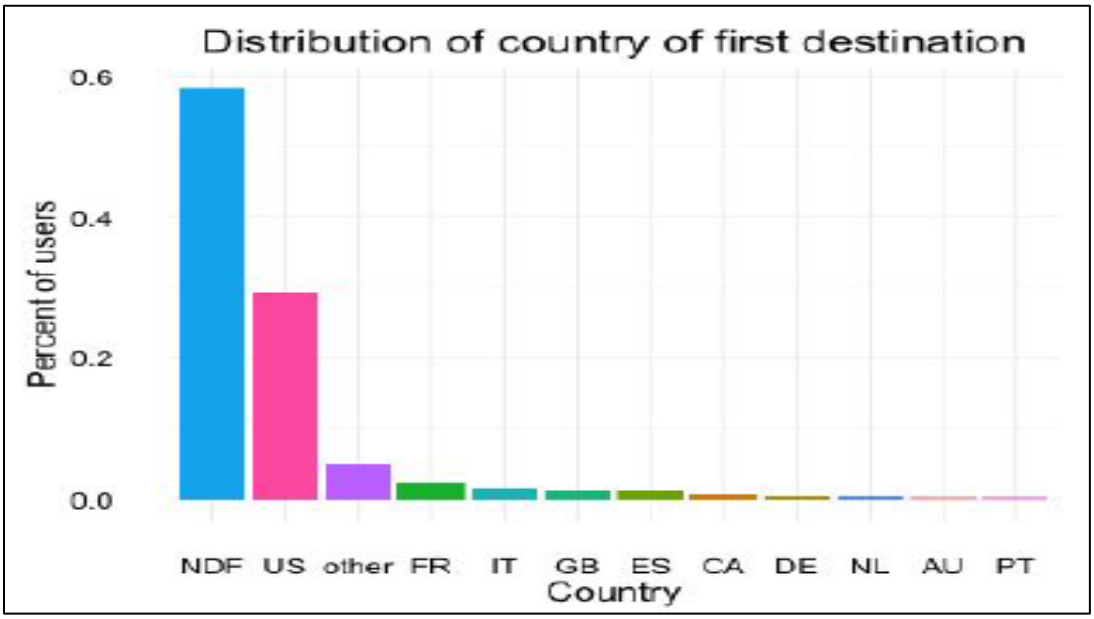
The first such visualization is on the gender distribution as below.

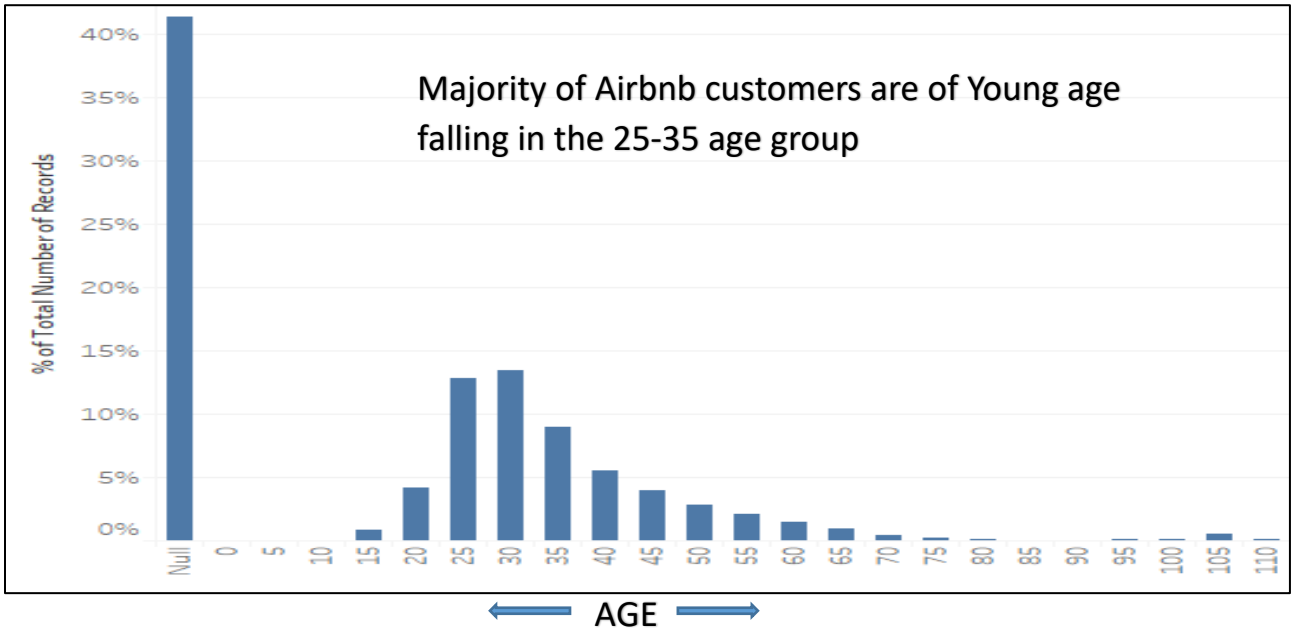


As we can observe a large number of entries has gender data missing. Otherwise the gender is fairly distributed among the data which are available. This implies that proper data cleaning is required to be done for the missing values. We will show in the data cleaning section how we assign the missing values as '-1' so that those data can be fed to the models.

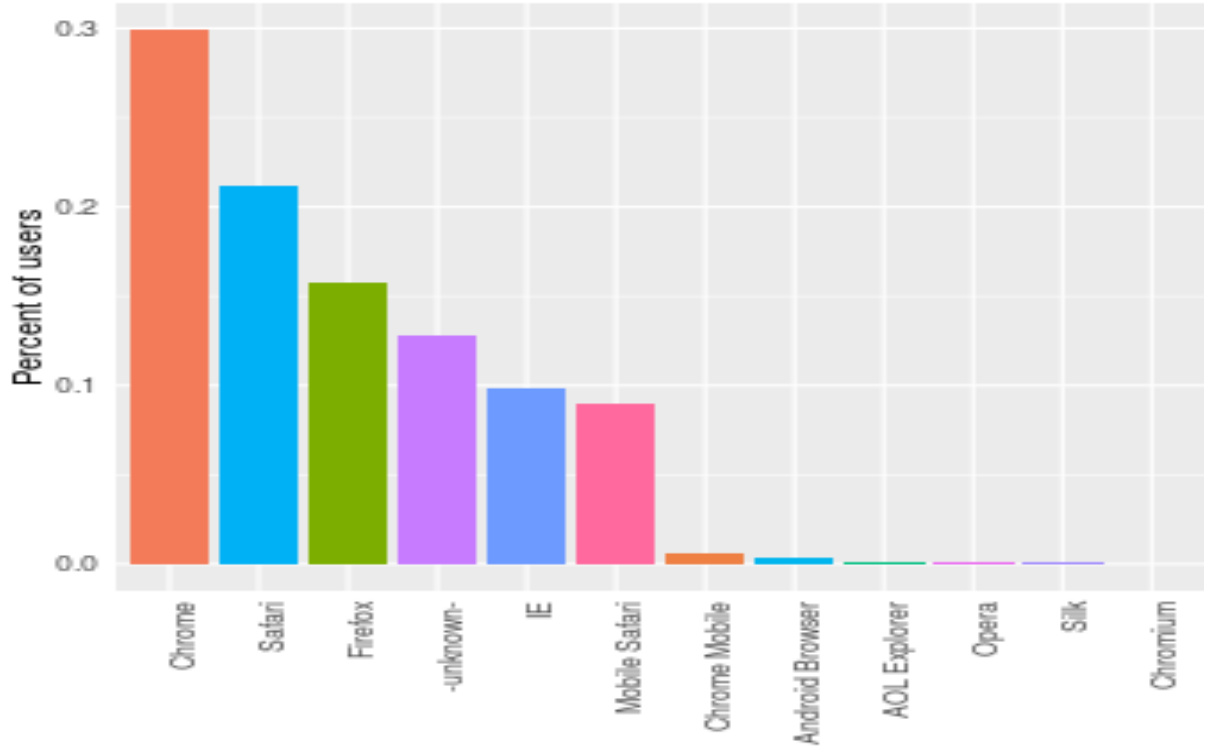
Our next visualization is on gender distribution. As we can see that US is the preferred choice for 1st time booking users. This itself gives us an idea about the possible destination for booking. Another point to consider is the large number (60%) as NDF among 1st time customers. This implies that a large majority do not make any bookings while browsing for 1st time.

Similarly, in the next visualization, the age distribution among the 1st time users have been depicted. We do see that majority of the users are of young age in the age group 25-35. Also, a large number of entries are missing which accounts for more than 40% of total booking. Hence, we need to employ some suitable data cleaning methods I order to prepare them for data modelling.

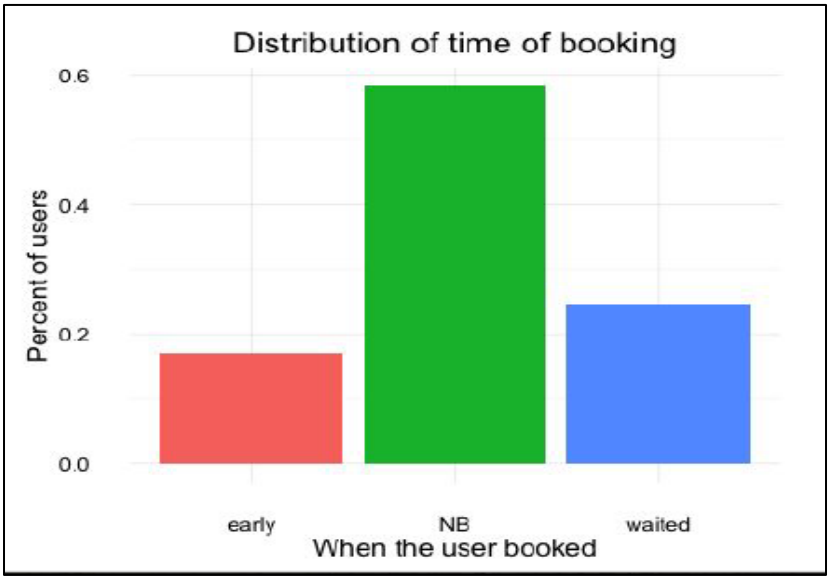




We present the next set of visualization on browsers and percentage of users using them. As we can see the chrome is the preferred tool for accessing Airbnb site.



Similarly, we can also draw an insight into the time spent online during booking. Ironically those who actually book an accommodation spend the least amount of time to do so.



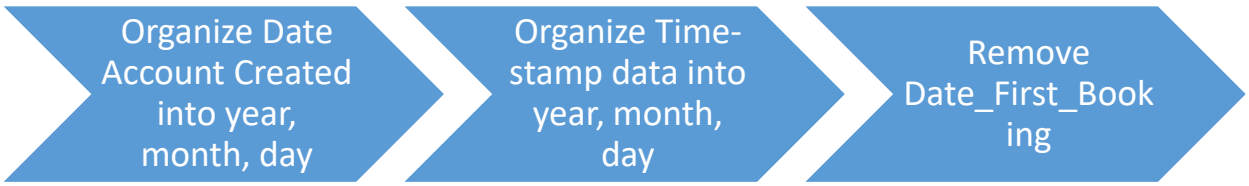
Data Cleaning and processing:

It is essential to clean and process the data so that it can be fed to the model for required processing. There are primarily 4 steps in data cleaning.

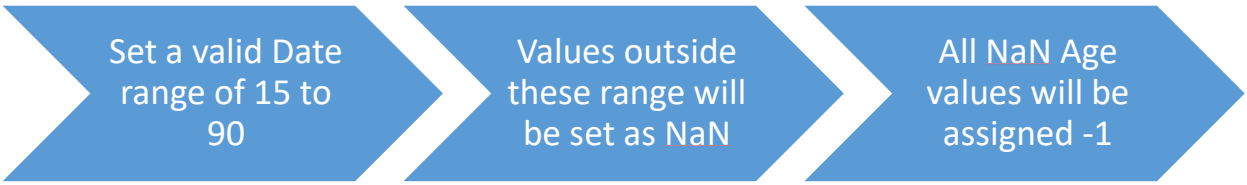
1) Standardizing Formats: When files are imported / exported through different file types, the data might be of different formats making it hard for software to understand. Hence this is an important step.
2) Filling Missing Values: Missing values needs to be suitable treated.
3) Correcting Eccentric Values: These values or outliers might hamper the model to run correctly.
4) Standardizing Categories: Categorical data must have a uniform structure.

Cleaning of Airbnb DataSet:

1) Clean the Dates: **:** It is important to clean the dates data so that the programming language can identify them as dates and able to use them in the prediction algorithm.



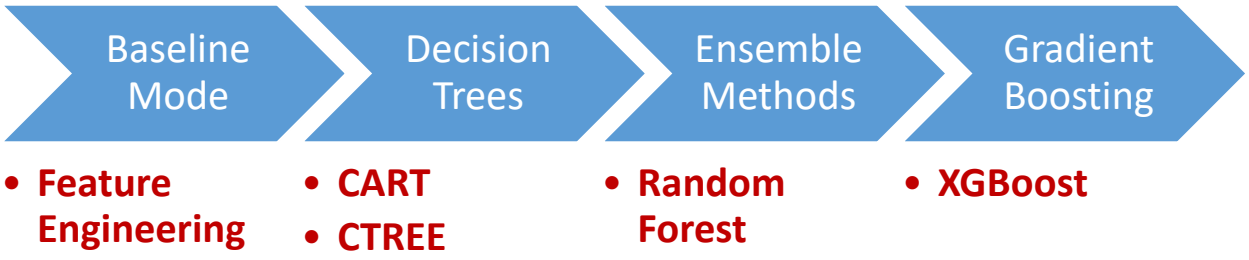
2) Clean the Age Data: Some age data are missing and some age data are unreasonable i.e age more than 100



3) **Clean the Gender Data:** Some gender data are also missing. These entries will be kept as NaN

Data Modelling:

Once the data is cleaned and processed, we can apply the required modelling techniques. We will be applying a number of modelling techniques, starting from baseline model to advance ones. Decision trees are a suitable class of modelling techniques for categorical data as in Airbnb dataset.



1) **Baseline Model:** We implement this model after doing exploratory data analysis and some form of feature engineering. As we had seen, that US as a destination country is the most common among the users who booked, we can consider as a baseline model that by default everyone will book in US as the destination country.

2) **Random Forest:** Random Forest is used because it allows for continuous and categorical feature. It will also rank the features in the form of an importance matrix. Random Forest provides the advantage of reduce chance of over fitting and higher model performance accuracy compared to baseline models

3) **XGBoost**: It is a superior algorithm which gives an even higher prediction accuracy. XGBoost is also a ensemble method like random forest but it tries to improve on each decision tree sequentially.
In order for XGBoost to work, all the categorical data are to be converted into numerical vector which is done by one hot encoding feature in R.

Model Validation:

Now that we have implemented different modelling techniques onto the dataset, it is time we evaluate the performance of those model in terms of prediction accuracy. As the nature of analytics problem, we develop the models in the train dataset and validate those models developed in the test dataset.

MODEL	PERFORMANCE
Baseline Model	29%
CART / CTREE (Decision Trees)	(50-70)%
Random Forest	77%
XGBoost	86%

Conclusion:

- Lot of insights can be inferred from exploratory data analysis and the baseline model can be formed by this.
- Decision Tree method is a good modelling technique for solving problems with categorical dataset like Airbnb.
- Ensemble method like random forest improves prediction rate and gradient boosting methods like XGBoost further enhance forecasting rate.
- Data Cleaning and processing forms a crucial step in this analytical exercise.
- Advance modelling technique like XGBoost not only predicts the outcome but also ranks the attributes in their contribution to the model to help in prediction.

APPENDIX

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from xgboost.sklearn import XGBClassifier
np.random.seed(0)

#Loading data
df_train = pd.read_csv('../input/train_users.csv')
df_test = pd.read_csv('../input/test_users.csv')
labels = df_train['country_destination'].values
df_train = df_train.drop(['country_destination'], axis=1)
id_test = df_test['id']
piv_train = df_train.shape[0]

#Creating a DataFrame with train+test data
df_all = pd.concat((df_train, df_test), axis=0, ignore_index=True)
#Removing id and date_first_booking
df_all = df_all.drop(['id', 'date_first_booking'], axis=1)

#Filling nan
df_all = df_all.fillna(-1)
#####Feature engineering#######
#date_account_created
dac = np.vstack(df_all.date_account_created.astype(str).apply(lambda x: list(map(int, x.split('-
')))).values)
df_all['dac_year'] = dac[:,0]
```

```
df_all['dac_month'] = dac[:,1]

df_all['dac_day'] = dac[:,2]

df_all = df_all.drop(['date_account_created'], axis=1)

#timestamp_first_active

tfa = np.vstack(df_all.timestamp_first_active.astype(str).apply(lambda x: list(map(int,
[x[:4],x[4:6],x[6:8],x[8:10],x[10:12],x[12:14]]))).values)

df_all['tfa_year'] = tfa[:,0]

df_all['tfa_month'] = tfa[:,1]

df_all['tfa_day'] = tfa[:,2]

df_all = df_all.drop(['timestamp_first_active'], axis=1)


#Age

av = df_all.age.values

df_all['age'] = np.where(np.logical_or(av<14, av>100), -1, av)


#One-hot-encoding features

ohe_feats = ['gender', 'signup_method', 'signup_flow', 'language', 'affiliate_channel',
'affiliate_provider', 'first_affiliate_tracked', 'signup_app', 'first_device_type', 'first_browser']

for f in ohe_feats:

    df_all_dummy = pd.get_dummies(df_all[f], prefix=f)

    df_all = df_all.drop([f], axis=1)

    df_all = pd.concat((df_all, df_all_dummy), axis=1)


#Splitting train and test

vals = df_all.values

X = vals[:piv_train]

le = LabelEncoder()

y = le.fit_transform(labels)
```

```
X_test = vals[piv_train:]

#Classifier
xgb = XGBClassifier(max_depth=6, learning_rate=0.3, n_estimators=25,
                    objective='multi:softprob', subsample=0.5, colsample_bytree=0.5, seed=0)
xgb.fit(X, y)
y_pred = xgb.predict_proba(X_test)

#Taking the 5 classes with highest probabilities
ids = []  #list of ids
cts = []  #list of countries
for i in range(len(id_test)):
    idx = id_test[i]
    ids += [idx] * 5
    cts += le.inverse_transform(np.argsort(y_pred[i])[::-1])[:5].tolist()

#Generate submission
sub = pd.DataFrame(np.column_stack((ids, cts)), columns=['id', 'country'])
sub.to_csv('sub.csv',index=False)
```