Airbnb new user booking prediction

1. Introduction :

Airbnb is a lodging company that connects host and guest depend on lodging type, date, location, and price through the website and mobile app. The host provides hospitability to the guest in order to gain a higher rate and advertise their rental. For this company to develop, first it is important to provide a great impression with their first experience. Airbnb wants to predict where new user likely to book their first travel experience. Depend on their age, gender, and personal preference, each customer will have a difference in choosing their first destination. Through accurate data analysis, the company will be able to provide a shorter time to find a destination and better-personalized contents for the first users.

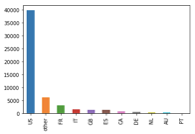
2. Datasets:

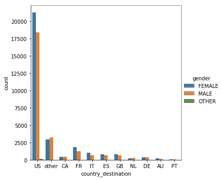
Through Kaggle competition, Airbnb has provided all the data that is necessary to predict the destination country. data is given in 6 different files. There is two background information about destination countries and the ages and gender of the new users. Three files are train\_users, test\_users, and sessions that are provided in a CSV file. The train user is a dataset that contains Airbnb users includes their destination. The next dataset is tested users has the same format as train users.csv without a destination. This is the file that Airbnb wants to have a prediction on. The session file is supplementary data that contains activities such as the weather they click to see the lodge or add to a personal list.

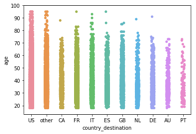
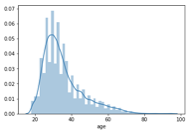
Since there are lot of missing data, first data was cleaned. In the variable age, since it was able to get average from other data, the empty spot for age was filled with mean. However, most of the data were categorical, it was hard to find the mean. Therefore, those NaN data were dropped using dropna function. Also, for the age, since the age below 18 is restricted to use Airbnb and age approximately over 95 would not use the Airbnb, therefore the data with age below 18 and over 95 were also dropped . Then train and test data were merged together in order to see how variables in train is related to the country destination in test data. The empty data in the session is also replaced as NaN value and dropped. This session data is also merged into test and train data by same column ‘user id’ in order to see how variables in session related to country destination.

The variable secs\_elapsed is grouped and summed by same user\_id to check how each id took and choose their destinations.

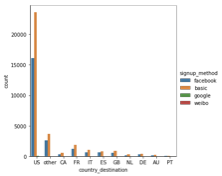
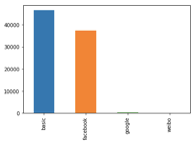
**3. exploratory analysis:**

After wrangling the data, each variable were plotted as a bar graph or scatter plot to reveal the relationship with a destination. But first, target variable was plotted. In the graph, most of the users selected US as there first destination. 

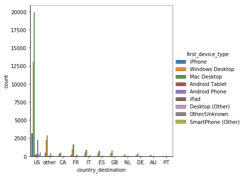
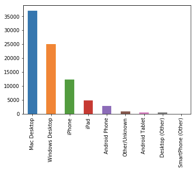
By counting only the gender variable, there was more female user than the male. However, by comparing the gender for each country, there were more female booker or equal amount in gender in other, Canada and Denmark. Since there is difference in gender in country selection, the gender may considered to be significant variable that effect the country destination.

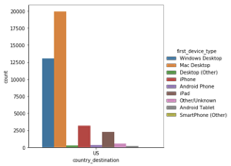
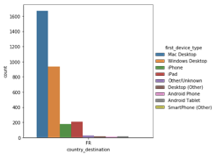


The age was plotted as bar graph. Since the age below 18 and over 95 is dropped, ther eis no data exist. The graph showed that age between 20 and 40 has highest amount of users. For the age in each country, it seems there are large amount of user until 60s. By comparing this plot with country selection plot, the age range tend to drop as country selection is dropped. However, while GB is more selected than ES in country destination, the ES has wider age range than GB. Therefore, the age could be considered as import variable that effect country destination.

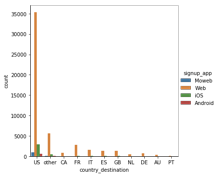
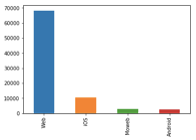


For Signup method, The basic and facebook was selected for most of the users as their signup method. Since the pattern matches in all the countries, the signup method variable was considered to be not related to the country destination.

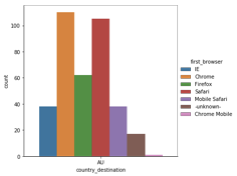
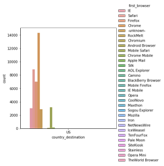
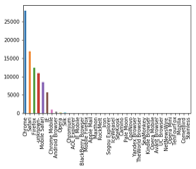




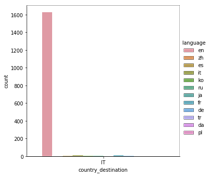
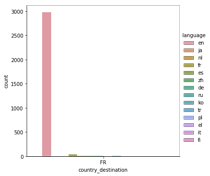
For the variable first device type, the mac and window desktop were used most. When comparing the countries, in all of the countries the mac desktop were used most then the window desktop were used. However, in US, while iphone were 3rd most used device, in France, the ipad were used more than iphoen. Eventhough it is small difference, is could be considered as important variable for the country destination.



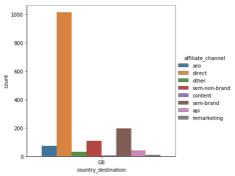
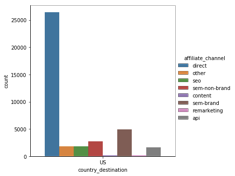
In signup app variable, all the countries revealed same pattern. The web was used most then ios was selected next. Since there was no significant pattern in each country, the signup app was considered to be not significant variable for predicting country destination.



For the first brower, Among the lot of the browser, chrome , safari and firefox were used repectively. It showed same pattern in each country. However, while there were a lot of the browser in US, about 31 kind of browser, there were only 7 browser were used in the Australia. Even though the pattern for top 6 broswers are same, the minor browsers might effect in selecting country destination. Therefore the browser was considered to be important variable

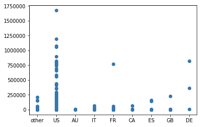


For language, in all the country, English was most used language. However, while it was hard to tell what language is used next to English in US, the French is used more than other languages for user who selected France as their destination. Which is obvious result, therefore, the language was considered to be important variable to predict country destination.



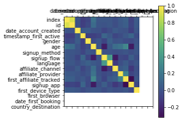
For afilliate channel, the most used channel were same in most countries but there was also minor difference in US and GB. There was more API user than the other channel in GB. This is minor difference, but this could result in country destination, therefore this variable was also considered to be important variable.

The variable affiliate provider and date created were plotted but there was no significant difference in each countries. Direct, google were selected most respectively. Also the date created were start to increase in 2012-01 in most of the country and gradually increased until 2014-07. Therefore, these two variables were considered to be not important as other variable.



while US, France and Italia were selected as country destination , For the variable second\_elapsed the user selected Denmark took more seconds than the user selected France and Italia. Therefore, this second elapsed variable was considered to be important variable for deciding country destination.

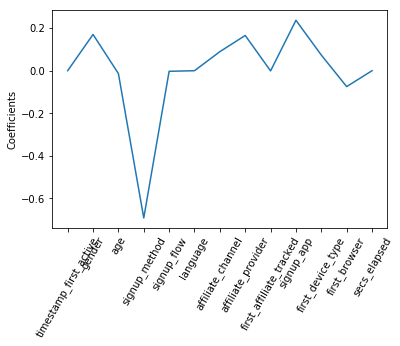
Since it is hard to identify which of the variable has effect on country destination, the correlation color graph was plotted to identify the variable has relationship with country destination. However, it was hard to tell the color of the plot and identify which of the variable has correlation with country destination.



Therefore, the correlation table were plotted using .corr function. However, all of the variable has 0 which means that they are slightly correlated or likely to be uncorrelated. Since the variables are categorical and have low correlation with dependent variables, the variable will be analyzed with multivariate logistic regression. Even though the correlation came out low in univariate analysis, it could be significant in the presence of other variables in the multivariate model.

Results and In-depth analysis using machine learning

In order to figure out important feature among all the variables, the data is splited into y : country destination and X for features. X and y variables were trained using train\_test\_split in order to generate train data and test data to run the data. Using Lasso regression, the number of features used were identified to be 9 features.



Using the Lasso coefficient plot, important 9 features are identified to be gender, age, signup method, affiliate channel, affiliate provider, first affiliated tracked, signup app, first device type, first browser

So I have dropped not used features ,time stamp frist active, signup flow, language and second elapsed.

in the given data , the prior knowledge about customers were given, and each of the given variables are labeled. Therefore the right method for this data is supervised multi classfication. Among the classifcation, since most of the datas were categorical variables and this is classification model, SVM and logistic regression analysis were used.

in order to predict, it is important to split given data into train and test data to learn about variables in train set and check the accuracy on the test set. In the logistic Analysis, the variables were trained in train features and target variable, country destination then the prediction accuracy level on the test set is 71%

The second method is Support Vector Machine (SVM). SVM is also a method of calssification similar to logistic regression model. The goal between SVM and logistic regresson are similar that gets best line that fits in data. The support vector machine algorithm finds the hyperplane that has maximum margin in given number of features.Support vectors are the points close to this hyperplane. using this vectors, we maximuze the margin of classifier and will build SVM.

The accuracy rate for the test set is same is logistic regression analysis, 71%. when the kernel is given to the SVM, the accuracy level dropped to 58%. Therefore, it tells there is no outlier, and the data is linearly separable.

The third model is XGBoost. The XGBoost model support binary classification, multiclass classification, regression and learning to rank. XGBoost is popular model recently that perdict target variable by combinding the estimates of set of simpler , weaker models. The given data is multiclass classification, therefore, the XGBoost were used. However, it only gives 58.54%

The XGBoost model has low accuracy level than the LR and SVM Algorithm which means the LR and SVM are more robust than the XGBoost model. This is due to curse of dimension that has low number of data points than number of features. This could means that there is overfitting with certain features or not having enough data or have too many feature can create certain result.

#code

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

import seaborn as sns

import datetime

test = pd.read\_csv('test\_users.csv')

train = pd.read\_csv('train\_users\_2.csv')

test['gender'] = test['gender'].replace('-unknown-', np.nan)

test['gender'].astype(str)

test = test.dropna(axis=1, how='all')

test.head()

testclean = test.dropna().reset\_index()

train.info()

train['gender'] = train['gender'].astype(str)

train['gender'] = train['gender'].replace('-unknown-', np.nan)

train.head(20)

cleantrain = train.dropna().reset\_index()

cleantrain.head()

df = pd.concat((testclean, cleantrain), axis=0, ignore\_index=True, sort=False)

df['age'].fillna(df['age'].mean(), inplace=True)

df.loc[df.age > 95, 'age'] = np.nan

df.loc[df.age < 18, 'age'] = np.nan

df['gender'].unique()

df.first\_affiliate\_tracked.unique()

df['gender'].value\_counts().plot(kind='bar')

# There is trend that female is more likely to register than male. However, there is significant amount of male also registered

df['country\_destination'].value\_counts().plot(kind='bar')

# The 1st country destination is US. 2nd is other and 3rd is France.

sns.catplot('country\_destination', hue='gender', kind='count', data=df)

# In most of the countries female are more likely to register.

# however, in other countries, male are more likely choose as destination.

# Therefore, there might be relationship between gender and destination selection.

sns.distplot(df.age.dropna())

# According to this bar plot, between age 25 to 40 has highest number of the people use airbnb.

sns.stripplot(x='country\_destination', y='age', data=df, jitter=True)

# In US, most of the age range have used Airbnb. However, other countries and other destination is less likely to chosen by age above 70s and 80s

#since age is not constant in selecting destination, there might be relationship between age and destination.

df.signup\_method.unique()

f['signup\_method'].value\_counts().plot(kind='bar')

# facebook and basic method is most popular method of signup

sns.catplot('country\_destination', hue='signup\_method', kind='count', data=df)

df['first\_device\_type'].value\_counts().plot(kind='bar')

# Most of the users used desktop to register. Among the desktop, I can tell the Mac desktop is more likely to be used than Window desktop.

us = df.loc[df['country\_destination'] == 'US']

other = df.loc[df['country\_destination'] == 'other']

fr = df.loc[df['country\_destination'] == 'FR']

gb = df.loc[df['country\_destination'] == 'GB']

it = df.loc[df['country\_destination'] == 'IT']

au = df.loc[df['country\_destination'] == 'AU']

ca = df.loc[df['country\_destination'] == 'CA']

de = df.loc[df['country\_destination'] == 'DE']

es = df.loc[df['country\_destination'] == 'ES']

nl = df.loc[df['country\_destination'] == 'NL']

pt = df.loc[df['country\_destination'] == 'PT']

sns.catplot('country\_destination', hue='first\_device\_type', kind='count', data=df)

sns.catplot('country\_destination', hue='first\_device\_type', kind='count', data=us)

sns.catplot('country\_destination', hue='first\_device\_type', kind='count', data=fr)

df['signup\_app'].value\_counts().plot(kind='bar')

sns.catplot('country\_destination', hue='signup\_app', kind='count', data=df)

sns.catplot('country\_destination', hue='signup\_app', kind='count', data=us)

sns.catplot('country\_destination', hue='signup\_app', kind='count', data=gb)

df['first\_browser'].value\_counts().plot(kind='bar')

# Among the lot of the browsers, chrome

sns.catplot('country\_destination', hue='first\_browser', kind='count', data=us)

sns.catplot('country\_destination', hue='first\_browser', kind='count', data=other)

sns.catplot('country\_destination', hue='first\_browser', kind='count', data=au)

df['language'].value\_counts().plot(kind='bar')

# Since the data is collected in US, the language used is english and it seems

sns.catplot('country\_destination', hue='language', kind='count', data=it)

sns.catplot('country\_destination', hue='language', kind='count', data=fr)

df['affiliate\_channel'].value\_counts().plot(kind='bar')

sns.catplot('country\_destination', hue='affiliate\_channel', kind='count', data=df)

sns.catplot('country\_destination', hue='affiliate\_channel', kind='count', data=us)

sns.catplot('country\_destination', hue='affiliate\_channel', kind='count', data=gb)

df['affiliate\_provider'].value\_counts().plot(kind='bar')

sns.catplot('country\_destination', hue='affiliate\_provider', kind='count', data=it

)

sns.catplot('country\_destination', hue='affiliate\_provider', kind='count', data=fr)

sns.catplot('country\_destination', hue='affiliate\_provider', kind='count', data=df)

df.info()

df['date\_account\_created'] = pd.to\_datetime(df['date\_account\_created'])

df['date\_account\_created'].value\_counts().plot()

plt.xlabel('Year')

plt.title('Account Created Date')

sns.despine()

# There are increase trend of creating the account as time goes and there was great increase in 2014

ca['date\_account\_created'] = pd.to\_datetime(ca['date\_account\_created'])

ca['date\_account\_created'].value\_counts().plot()

fr['date\_account\_created'] = pd.to\_datetime(fr['date\_account\_created'])

fr['date\_account\_created'].value\_counts().plot()

age = pd.read\_csv('age\_gender\_bkts.csv')

age.groupby(age['country\_destination'])

age = age.set\_index(['country\_destination'])

age.head()

session = pd.read\_csv('sessions.csv')

session.head(10)

session.info()

session['action'] = session['action'].replace('-unknown-', np.nan)

session['action\_type'] = session['action\_type'].replace('-unknown-', np.nan)

session['action\_detail'] = session['action\_detail'].replace('-unknown-', np.nan)

session['device\_type'] = session['device\_type'].replace('-unknown-', np.nan)

session = session.dropna()

session = session.rename(columns={'user\_id': 'id'})

df2 = pd.merge(df,session)

df2 = df2.drop(columns='index')

df2['action'].value\_counts().plot(kind='bar')

df2['device\_type'].value\_counts().plot(kind='bar')

sns.catplot('country\_destination', hue='device\_type', kind='count', data=df2)

secs = df2.groupby(['id']).sum()

df3 = pd.merge(df,secs)

df3 = df3.dropna(subset=['country\_destination'])

df3.head()

us = df3.loc[df3['country\_destination'] == 'US']

fr = df3.loc[df3['country\_destination'] == 'FR']

gb = df3.loc[df3['country\_destination'] == 'GB']

it = df3.loc[df3['country\_destination'] == 'IT']

au = df3.loc[df3['country\_destination'] == 'AU']

ca = df3.loc[df3['country\_destination'] == 'CA']

de = df3.loc[df3['country\_destination'] == 'DE']

es = df3.loc[df3['country\_destination'] == 'ES']

nl = df3.loc[df3['country\_destination'] == 'NL']

pt = df3.loc[df3['country\_destination'] == 'PT']

plt.scatter('country\_destination', 'secs\_elapsed',data = df3)

Among all the variables on the data, there are some variables that

might have relationship with country destination. For gender,

majority of people choose US as their destination and

more female choose US as their destination. But in other countries,

CA, NL DE has different pattern than US.

Some of the countries is selected by more male or have same rate of female and male.

This represents that gender effects in choose country destination.

For age, US and other countries is choose by in most of the age, but in rest countries,

there were less age over 70 espiecally in CA, AU and PT. IN GB, there is no user over 90.

Since there is different pattern of users’ age depend on their country destination,

therefore age effects the country destination.

Rest of the variables signup method, first device type, browser,

language, affiliate \_ cahnnel , provider have different pattern with US and other countries.

Most of the variables had same pattern in first 2 or 3 method but revealed different pattern

in method with small amount. Since this small pattern can decide the country destination,

all the variables are important.

For the session data, it displays huge difference between countries.

With us, it took most seconds however, in rest of the countries FR and DE took 75000 seconds

while rest took less than 25000.

Therefore, it might have significant effect on the country destination.

df.gender = df.gender.astype("category").cat.codes

df.signup\_method = df.signup\_method.astype("category").cat.codes

df.language = df.language.astype("category").cat.codes

df.affiliate\_channel=df.affiliate\_channel.astype("category").cat.codes

df.affiliate\_provider=df.affiliate\_provider.astype("category").cat.codes

df.first\_affiliate\_tracked = df.first\_affiliate\_tracked.astype("category").cat.codes

df.signup\_app = df.signup\_app.astype("category").cat.codes

df.first\_device\_type = df.first\_device\_type.astype("category").cat.codes

df.first\_browser=df.first\_browser.astype("category").cat.codes

df.country\_destination=df.country\_destination.astype("category").cat.codes

Xlr, Xtestlr, ylr, ytestlr = train\_test\_split(df['Height','Weight']].values,

(df.Gender == "Male").values,random\_state=5)

plt.matshow(df.corr())

plt.xticks(range(len(df.columns)), df.columns)

plt.yticks(range(len(df.columns)), df.columns)

plt.colorbar()

plt.show()

correlation = df.corr(method='kendall')

#!/usr/bin/env python

# coding: utf-8

# In[68]:

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

import seaborn as sns

import datetime

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

# In[69]:

train = pd.read\_csv('train\_users\_2.csv')

session = pd.read\_csv('sessions.csv')

# In[70]:

train['gender'] = train['gender'].astype(str)

train = train.replace('-unknown-', np.nan)

train['gender'] = train['gender'].replace('nan', np.nan)

# In[71]:

session['action'] = session['action'].replace('-unknown-', np.nan)

session['action\_type'] = session['action\_type'].replace('-unknown-', np.nan)

session['action\_detail'] = session['action\_detail'].replace('-unknown-', np.nan)

session['device\_type'] = session['device\_type'].replace('-unknown-', np.nan)

session = session.rename(columns={'user\_id': 'id'})

# In[72]:

sec = session.groupby(['id']).sum()

# In[73]:

df2 = pd.merge(train,session)

# In[74]:

secs = df2.groupby(['id']).sum()

secs.head()

# In[75]:

df = pd.merge(train,secs)

df = df.dropna(subset=['country\_destination'])

df.shape

# In[76]:

df.loc[df.age > 95, 'age'] = np.nan

df.loc[df.age < 18, 'age'] = np.nan

# In[77]:

df['age'].fillna(df['age'].mean(), inplace=True)

# In[78]:

df = df.dropna()

df.gender = df.gender.astype("category").cat.codes

df.signup\_method = df.signup\_method.astype("category").cat.codes

df.language = df.language.astype("category").cat.codes

df.affiliate\_channel=df.affiliate\_channel.astype("category").cat.codes

df.affiliate\_provider=df.affiliate\_provider.astype("category").cat.codes

df.first\_affiliate\_tracked = df.first\_affiliate\_tracked.astype("category").cat.codes

df.signup\_app = df.signup\_app.astype("category").cat.codes

df.first\_device\_type = df.first\_device\_type.astype("category").cat.codes

df.first\_browser=df.first\_browser.astype("category").cat.codes

df.country\_destination=df.country\_destination.astype("category").cat.codes

# In[79]:

df = df.drop(['id','date\_account\_created','date\_first\_booking'], axis=1)

# In[80]:

X = df.drop(['country\_destination'], axis=1).values

# In[81]:

X.reshape(-1,1)

# In[82]:

y= df['country\_destination'].values

y.reshape(-1,1)

# In[83]:

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics, datasets

from sklearn.model\_selection import train\_test\_split

# In[84]:

X = df[['gender','signup\_method','language','affiliate\_channel','affiliate\_provider','first\_affiliate\_tracked','signup\_app',

'first\_device\_type','first\_browser','age']]

y = df['country\_destination']

clf = LogisticRegression()

# In[85]:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.4, random\_state=42)

# In[86]:

clf.fit(X\_train, y\_train)

# In[87]:

print(accuracy\_score(clf.predict(X\_test), y\_test))

# In[23]:

from sklearn.linear\_model import Lasso

names = df.drop('country\_destination', axis=1).columns

# In[88]:

lasso = Lasso(alpha=0)

lasso\_coef=lasso.fit(X, y).coef\_

\_ = plt.plot(range(len(names)), lasso\_coef)

\_ = plt.xticks(range(len(names)), names, rotation = 90)

\_ = plt.ylabel('Coefficients')

\_ = plt.axhline(0, color='white')

# In[ ]:

dropping time first active, signupflow, language, and secs\_elapsed

# In[47]:

df

# In[64]:

df = df.drop(['timestamp\_first\_active','signup\_flow','language','secs\_elapsed'], axis=1)

# In[92]:

X1 = df.drop(['country\_destination'],axis=1).values

X1.reshape(-1,1)

# In[93]:

y1 = df.country\_destination.values

y1.reshape(-1,1)

# In[94]:

logreg = LogisticRegression()

X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(X1, y1, test\_size = 0.4, random\_state=42)

# In[96]:

logreg.fit(X\_train1, y\_train1)

y\_pred1 =logreg.predict(X\_test1)

print(confusion\_matrix(y\_test1, y\_pred1))

print(classification\_report(y\_test1, y\_pred1))

print(accuracy\_score(logreg.predict(X\_test1), y\_test1))

# In[82]:

#SVM

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

svc = SVC()

svc.fit(X\_train1, y\_train1)

print("train set accuracy SVM: {:.2f}".format(svc.score(X\_train1, y\_train1)))

print("test set accuracy SVM : {:.2f}".format(svc.score(X\_test1, y\_test1)))

# In[79]:

svm = SVC(kernel='rbf', C=10.0, random\_state=0, gamma=0.10)

# In[83]:

svm.fit(X\_train1, y\_train1)

y\_pred2 = svm.predict(X\_test1)

print('Accuracy: %.2f' % accuracy\_score(y\_test1, y\_pred2))

# In[95]:

target = df.country\_destination

# In[96]:

X = df.drop(['country\_destination'],axis=1)

# In[97]:

X\_train2, X\_test2, y\_train2, y\_test2 = train\_test\_split(X, y, test\_size=0.2, random\_state=123)

# In[1]:

import pip

# In[2]:

get\_ipython().system('pip3 install xgboost')

# In[3]:

import xgboost as xgb

# In[35]:

train = pd.read\_csv('train\_users\_2.csv')

session = pd.read\_csv('sessions.csv')

# In[36]:

session = session.rename(columns={'user\_id': 'id'})

# In[37]:

df2 = pd.merge(train,session)

# In[38]:

secs = df2.groupby(['id']).sum()

# In[39]:

df = pd.merge(train,secs)

# In[40]:

df = df.drop(['id','date\_account\_created','date\_first\_booking'], axis=1)

df.head()

# In[41]:

y = df.country\_destination

X = df.drop('country\_destination',axis=1)

# In[43]:

from sklearn import model\_selection

from sklearn.metrics import accuracy\_score

from sklearn.preprocessing import LabelEncoder

# In[59]:

label\_encoder = LabelEncoder()

label\_encoder = label\_encoder.fit(y)

label\_encoded\_y = label\_encoder.transform(y)

# In[60]:

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, label\_encoded\_y, test\_size = 0.4, random\_state=42)

# In[61]:

model = xgb.XGBClassifier()

model.fit(X\_train, y\_train)

print(model)

# In[50]:

X.info()

# In[56]:

X['gender'] = X['gender'].astype(str)

X.gender = X.gender.astype("category").cat.codes

X.signup\_method = X.signup\_method.astype("category").cat.codes

X.language = X.language.astype("category").cat.codes

X.affiliate\_channel=X.affiliate\_channel.astype("category").cat.codes

X.affiliate\_provider=X.affiliate\_provider.astype("category").cat.codes

X.first\_affiliate\_tracked = X.first\_affiliate\_tracked.astype("category").cat.codes

X.signup\_app = X.signup\_app.astype("category").cat.codes

X.first\_device\_type = X.first\_device\_type.astype("category").cat.codes

X.first\_browser=X.first\_browser.astype("category").cat.codes

# In[58]:

y=y.astype("category").cat.codes

# In[62]:

y\_pred = model.predict(X\_test)

predictions = [round(value) for value in y\_pred]

# In[63]:

accuracy = accuracy\_score(y\_test, predictions)

print("Accuracy: %.2f%%" % (accuracy \* 100.0))

# In[67]:

# set missing values to 0

X[X == '-unknown-'] = 0

# convert to numeric

X = X.astype('float32')

# encode Y class values as integers

label\_encoder = LabelEncoder()

label\_encoder = label\_encoder.fit(y)

label\_encoded\_y = label\_encoder.transform(y)

# split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, label\_encoded\_y, test\_size=0.33, random\_state=7)

# fit model no training data

model = xgb.XGBClassifier()

model.fit(X\_train, y\_train)

print(model)

# make predictions for test data

y\_pred = model.predict(X\_test)

predictions = [round(value) for value in y\_pred]

# evaluate predictions

accuracy = accuracy\_score(y\_test, predictions)

print("Accuracy: %.2f%%" % (accuracy \* 100.0))

# In[ ]: