2020

Machine Learning

Audrey Rahimi

1/27/2020

My project is based on building model to predict which country a new user is most likely book first. In this project there is not any partner to peruse this goal and it is self studying with the instructor's supervision Mr. Nie.

In this project 2 data sets such as Session and train csv file are given and these data sets helping us to know more about training and testing and how user profile is impotant.

Train file: 16 features are as bellows:
id, date_account_created, timestamp_first_active, 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 and country_destination which is the target of the project.

Session file: 6 features are as bellows:
User_id, action, action_type, action_detail, device_type, secs_elapsed

I started my analysis through the data cleaning/pipeline for dft data frame (train file) as describe here:

1- By loading 2 files I got information of all the users in the train and test sets.

   I used dfs for session data frame and dft for train data set.

2- I checked null values.

```
(df.isnull().sum())
```

Afterexamining, I got that null values exist in the age column, country_destination, date_first_booking, first_affiliate_tracked and do replacing the unknowns with NaNs then check if there is any missing data. We find that there is unknowns in the columns 'gender' and 'first_browser'.

3- For age I replace with median then set dft['age'].loc[(dft['age']>90) | (dft['age']<18)] to have better and clear zone.

Also based on analysing I got that most users are in the range of 20 to 40.

4- For first browser, I used LabelEncoder() to make it more efficient data.

5- For gender, I prefer to use dummies encoder to have better out put.
I also saw that there are no big differences between the male and female users.

```
dg=pd.get_dummies(dft['gender'])
dg.drop(['-unknown-','OTHER'],axis=1,inplace=True)
```

6- In my analysis, about country destination I got that more people booked Us , also I change this columns with LabelEncoder to create readable column.

```
dc=dft['country_destination'].astype(str)
le2 = LabelEncoder()
dft['country_destination']=le2.fit_transform(dc)
```

7- For affiliated column, I used LabelEncoder() to make the data be more ready.
I got very interesting point that there are about 2/3 of the users came to the website directly without any affiliate(please consider the last page of analyses).

8- first_device_type: our analysis shows that Mac desktop then Windows desktop got better result. Relation of first device and country destination helps us to have the idea that Mac Desktop and after it Windows was the great device for the users in the US, also we used LabelEncoder() to make the data be more efficient. Codes written like the previous example encoder.

9- Language: the most popular language is English. We used LabelEncoder() to make the data be more efficient.

10- Dates: there is an interesting movement in account creation after the year of 2014. And between months, the greatest months were July, August and October. We changed the date to datetime type.

```
dft['date_account_created']=pd.to_datetime(dft.date_account_created)
```

```
dft['date_account_created'].fillna(method='ffill')
```

11- I divide timestamp_first_active and date_account_created column in month and day to make more features to have better result and also convert to datetime.

```
dft['timestamp_first_active']=pd.to_datetime(dft.timestamp_first_active)

dft['timestamp_first_active'].fillna(method='ffill')

dft['Month_timestamp']=dft.timestamp_first_active.dt.month

dft['Day_timestamp']=dft.timestamp_first_active.dt.day
```

Train file
Function "feature_engineering"

preprocessing and cleaning

In session file I did data cleaning such as 1- checking null values and checking unknowns and make correction for 6 columns. Except device_type all columns needed be filled. For secs_elapsed, we filled the empty places with mean.

As I mentioned before, we did all process of predictive modeling such as Loading the data sets and doing data prepocessing also did Feature engineering.

we rename the column user_id as just id to be matched the train and test columns. We did group by user_id 'secs_elapsed' and also with 'country_destination'.

I created one new data frame with the name of merg with 20 columns, it is our new data frame which I checked the did cleaned up again to make sure not having any wrong value.

0 id 213451 non-null object / 1 age 213451 non-null float64/ 2 signup_method 213451 non-null int32

3 signup_flow 213451 non-null int64 / 4 language 213451 non-null int32 / 5 affiliate_channel 213451 non-null int32

6 affiliate_provider 213451 non-null int32 / 7 signup_app 213451 non-null int32 / 8 first_device_type 213451 non-null int32

9 first_browser 213451 non-null int32 / 10 country_destination 213451 non-null int32 / 11 FEMALE 213451 non-null uint8

12 MALE 213451 non-null uint8 / 13 Month_created 213451 non-null int64 / 14 Day_created 213451 non-null int64

15 Month_booking 213451 non-null float64/ 16 Day_boking 213451 non-null float64/ 17 Month_timestamp 213451 non-null int64

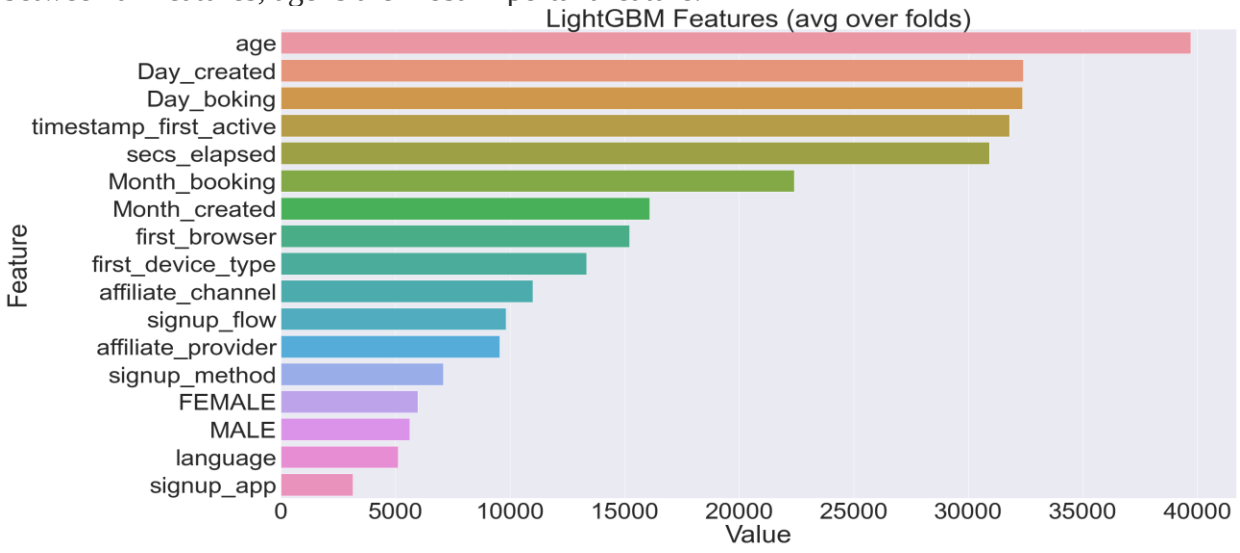
18 Day_timestamp 213451 non-null int64 / 19 secs_elapsed

I did all modeling prediction on this new merg dataframe.

Training the model and making predictions: (Function ‘ Modeling1’)

We used lightgbm and considering kf calculation we got the result with adding some lightgbm properties which work with my data:
Considering: objective='multiclass',and metric='multi_logloss, num_class=11 and eval_metric = 'multi_logloss', verbose=500, early_stopping_rounds = 300

Finally, we got that in best iteration which is in the fifth folder training's multi_logloss: 0.980425.
Features importance: In the last part of project, I did features importance coding and found that between all features, age is the most important feature.



By knowing which destination countries are more favorable along knowing different features, machine learning can help business to predict their market and know where they stand. It is very useful for the business to know some important features to be more successful in big and competitive markets.

Summary of my analysis
Loading the data sets and doing some data preprocessing
Feature engineering: **dft['date_first_booking']=pd.to_datetime(dft.date_first_booking)** **dft['date_first_booking'].fillna(method='ffill') or dft['year_first_active'] = df.timestamp_first_active.dt.yea**
Evxample of in place of NaNs **dft['age'].fillna(-1, inplace=True)**
Example of Creating train dataset **Train-dft= dft.loc[dft['id']]**
Creating test set **Test-dft= dft.loc[dft['id']].drop('country_destination', axis=1)**
Training the model and making prediction

More Models:

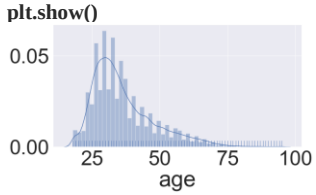
linear model using Perceptron
from sklearn.linear_model import Perceptron
from sklearn.metrics import f1_score,precision_recall_fscore_support,classification_report,confusion_matrix
model_pc = Perceptron(tol=1e-4, random_state=42,penalty='l2')
model_pc.fit(X_train,y_train)
Perceptron(alpha=0.0001, class_weight=None, early_stopping=False, eta0=1.0,
fit_intercept=True, max_iter=1000, n_iter_no_change=5, n_jobs=None,
penalty='l2', random_state=42, shuffle=True, tol=0.0001,
validation_fraction=0.1, verbose=0, warm_start=False)
y_pred_pc = model_pc.predict(X_valid)

Train a random forest as nonlinear model on the same data
from sklearn.ensemble import RandomForestClassifier
model_rf = RandomForestClassifier(n_estimators=5000,random_state=10,n_jobs=-1,max_depth=10,verbose=1)model_rf.fit(X_train,y_train)
y_pred_rf = model_rf.predict(X_valid)
y_pred_rf = model_rf.predict_proba(X_valid)
confusion_matrix(y_valid,y_pred_rf)
f1_score(y_valid,y_pred_rf)

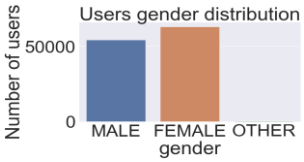
Knowing that we do not use f1 for our data set instead we use logloss in modeling.

Summary of some analysis

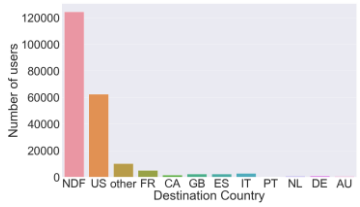
plt.figure(figsize=(12,6))
sns.countplot(x='gender', data=dft)
plt.ylabel('Number of users')
plt.title('Users gender distribution')
plt.show()



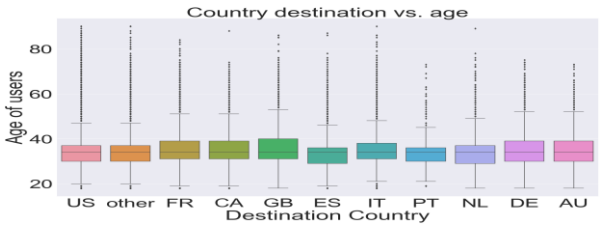
plt.figure(figsize=(12,6))
sns.countplot(x='gender', data=dft)
plt.ylabel('Number of users')
plt.title('Users gender distribution')
plt.show()



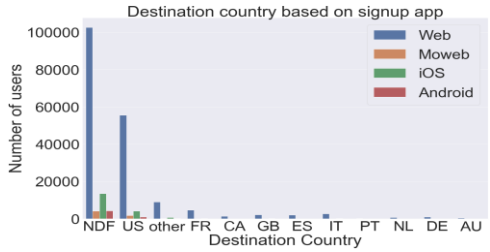
plt.figure(figsize=(25,15))
sns.countplot(x='country_destination', data=dft)
plt.xlabel('Destination Country')
plt.ylabel('Number of users')
plt.show()



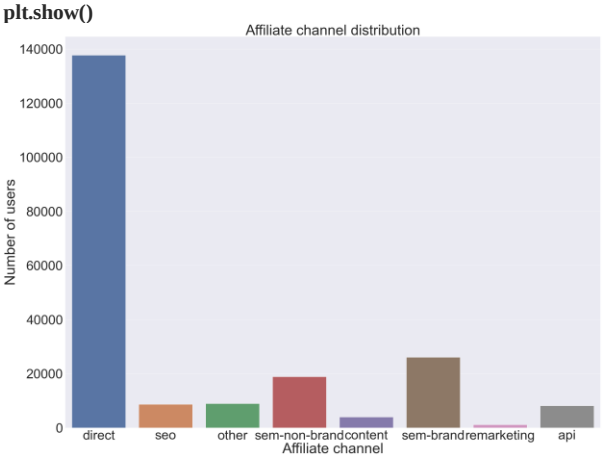
plt.figure(figsize=(25,15))
dft_without_NDF = dft[dft['country_destination']!='NDF']
sns.boxplot(y='age' , x='country_destination',data=dft_without_NDF)
plt.xlabel('Destination Country')
plt.ylabel('Age of users')
plt.title('Country destination vs. age')
plt.show()



plt.figure(figsize=(25,15))
sns.countplot(x='country_destination', data=dft,
hue='signup_app')
plt.xlabel('Destination Country')
plt.ylabel('Number of users')
plt.title('Destination country based on signup app')
plt.legend(loc = 'upper right')
plt.show()



plt.figure(figsize=(40,30))
sns.countplot(x='affiliate_channel', data=dft)
plt.xlabel('Affiliate channel')
plt.ylabel('Number of users')
plt.title('Affiliate channel distribution')
plt.show()



Regarding studying of Tanmayee's article published in 2019