airbnb

Predicting New User Bookings

Anaelia Ovalle, Michael Liston, Brent Rucker

Table of contents

Introduction

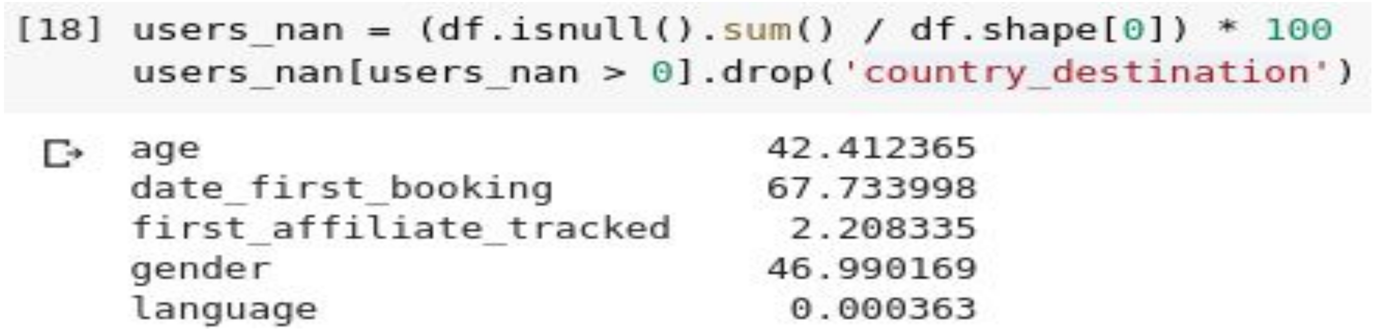
New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. By accurately predicting where a new user will book their first travel experience, Airbnb can share more personalized content with their community, decrease the average time to first booking, and better forecast demand.

The challenge is to predict in which country a new user will make his or her first booking.

Data available:
test_users.csv and train_users.csv : contains basic information and destination of past users.
Sessions.csv : contains additional web log informations generated when user use Airbnb website.

Data exploration

Check for NULL values in features

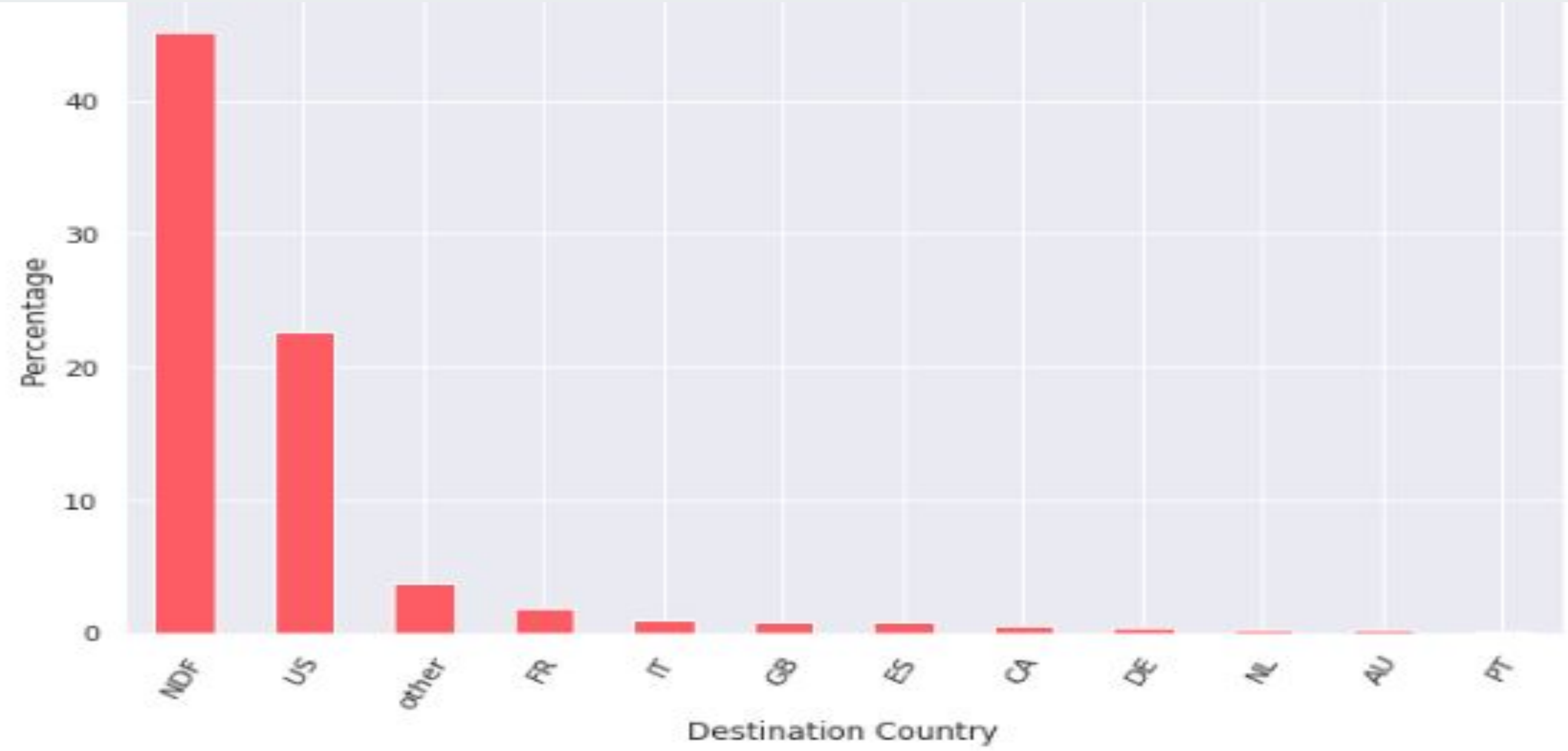


```
[18] users_nan = (df.isnull().sum() / df.shape[0]) * 100
     users_nan[users_nan > 0].drop('country_destination')
```

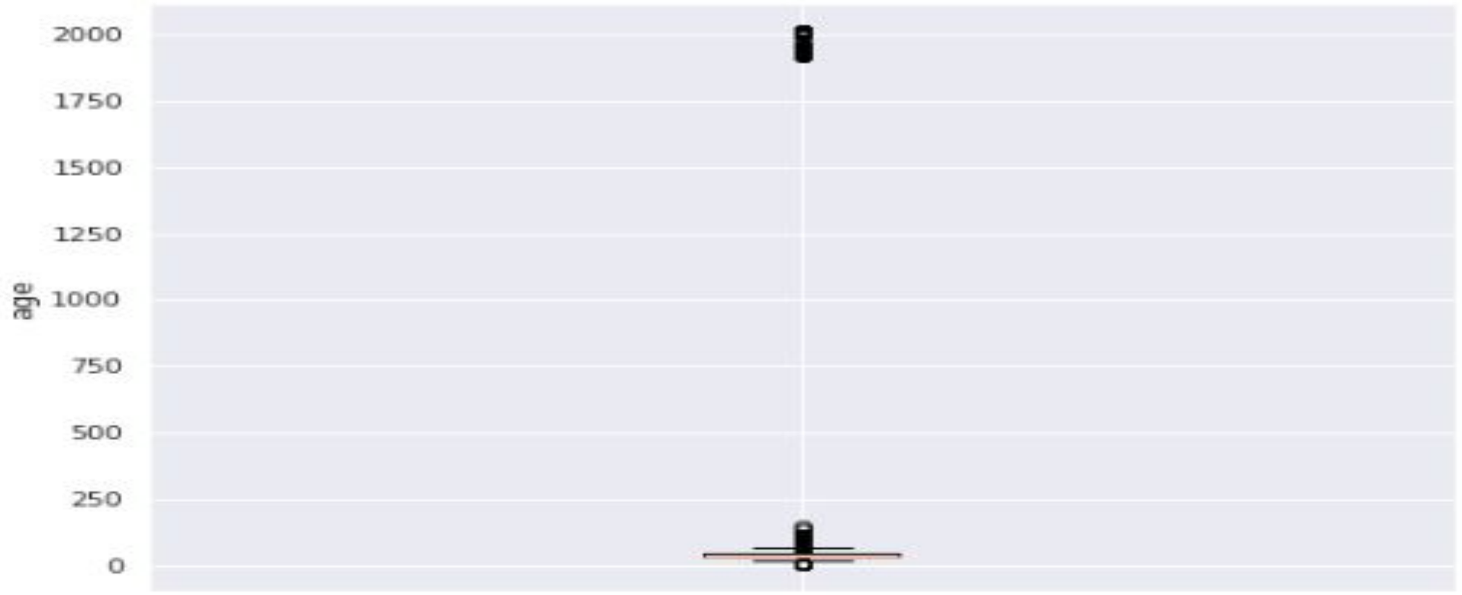
```
age                        42.412365
date_first_booking         67.733998
first_affiliate_tracked     2.208335
gender                     46.990169
language                    0.000363
```

We have quite a lot of NaN in the age and gender wich will yield in lesser performance of the classifiers we will build. The feature date_first_booking has a 67% of NaN values because this feature is not present at the tests users, and therefore, we won't need it at the modeling part.

Distribution of data over destination country(target variable):



Outliers in numerical variables:



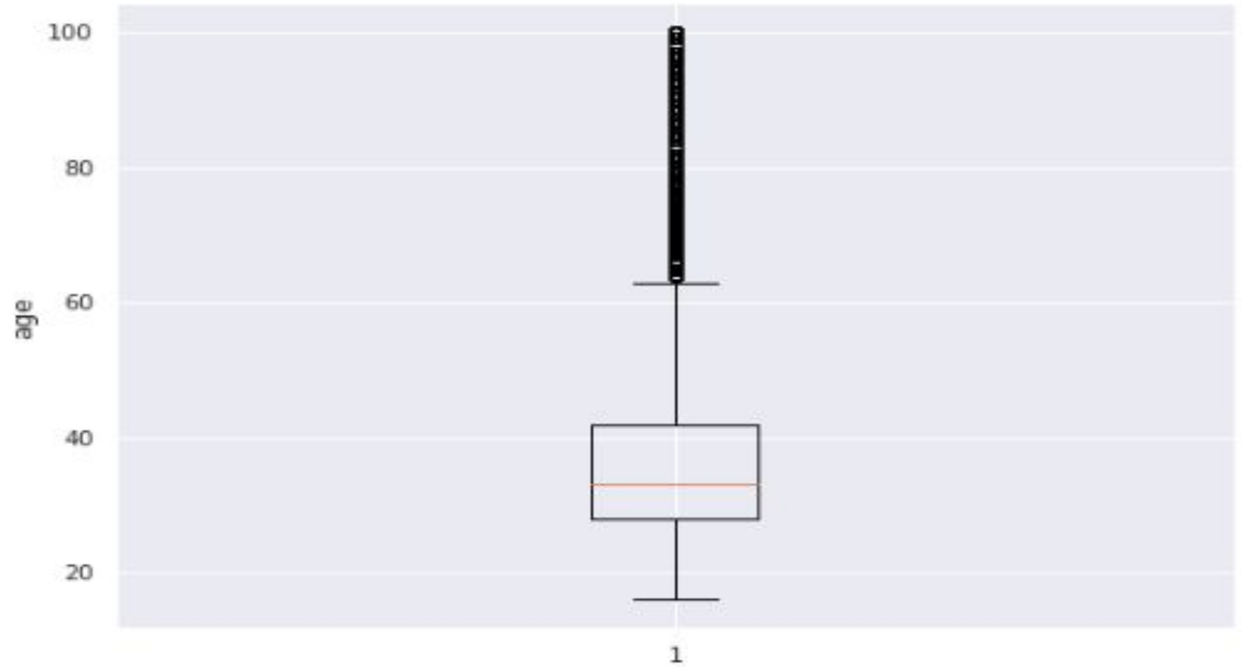
Here is a boxplot of the age variable showing that there are some outliers far away from the logical age interval.

Data pre-processing

- **Processing date columns.**
  For date columns (date_account_created, date_first_active) we separate them to new columns that express meaningful time values like days of the week, month, and day of the month. we do that as these new columns should reflect specific habits or user behavior in specific times at the week or month.

- **Handling outliers.**

We remove age values that are greater than 100 or less than 16



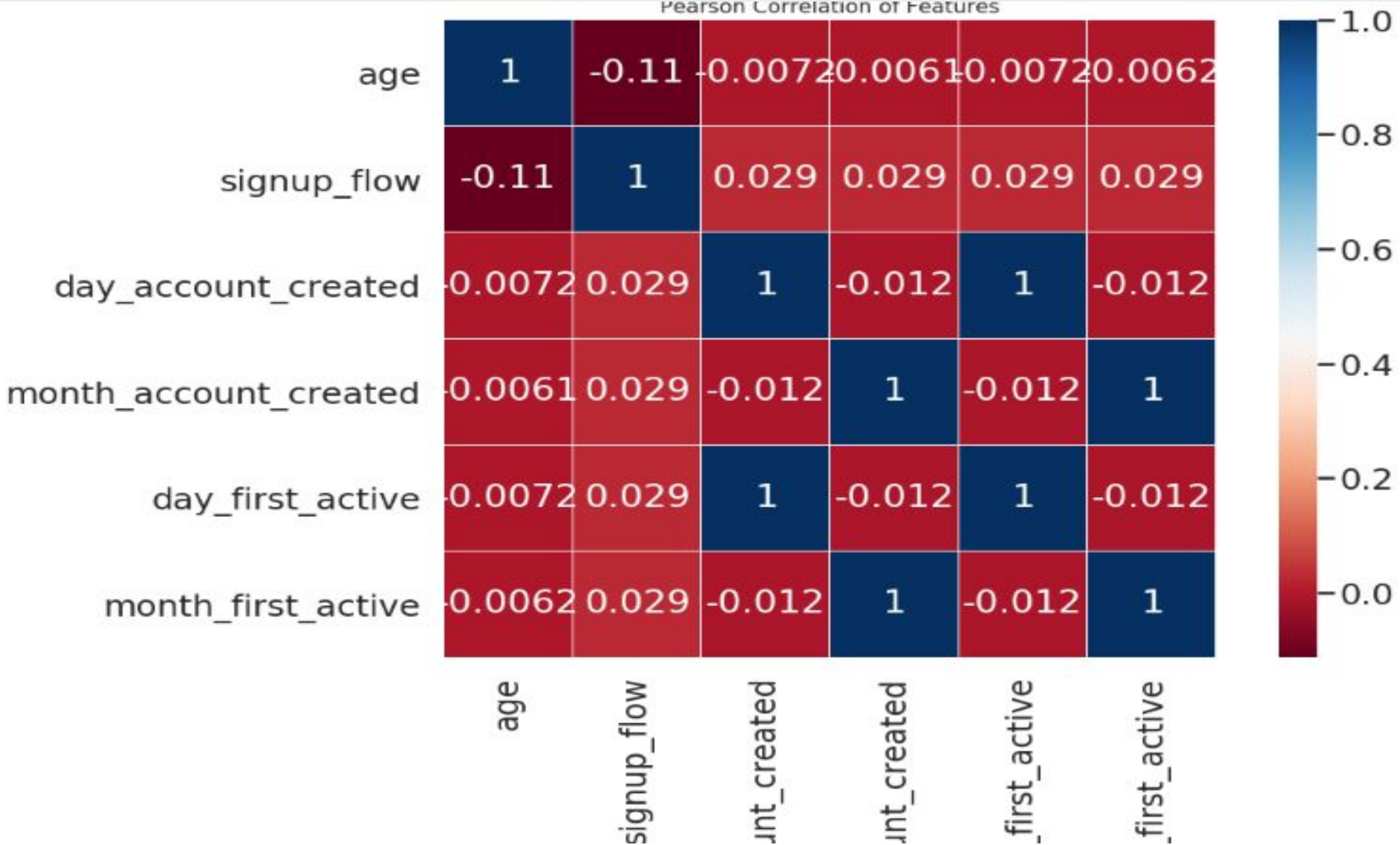
Here is a boxplot showing age distribution after removing outliers

- **Missing values.**

For numerical variables, we fill any missing value with the mean of it's variable. and for categorical variables, we fill it with the most frequent category.

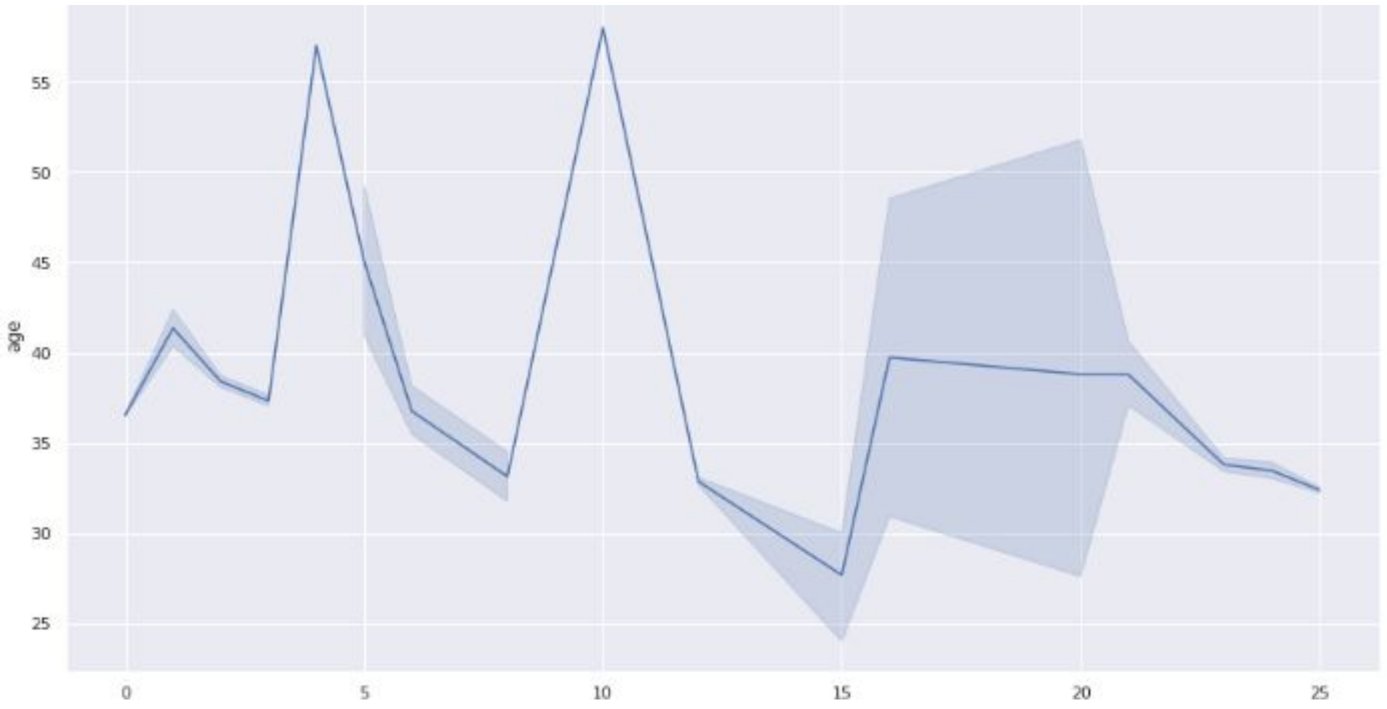
another technique could be used with numerical columns, is to predict those missing values of the variable using other variables that are highly correlated with it.

So we draw a heatmap showing correlation of the variables with each other.


Pearson Correlation of Features
age
signup_flow
day_account_created
month_account_created
day_first_active
month_first_active
1 -0.11 -0.0072 0.0061 -0.0072 0.0062
-0.11 1 0.029 0.029 0.029 0.029
-0.0072 0.029 1 -0.012 1 -0.012
0.0061 0.029 -0.012 1 -0.012 1
-0.0072 0.029 1 -0.012 1 -0.012
0.0062 0.029 -0.012 1 -0.012 1
age
signup_flow
unt_created
unt_created
first_active
first_active
1.0
0.8
0.6
0.4
0.2
0.0

The heatmap showed that ‘signup_flow’ is the most correlated variable to ‘age’.

drawing the relation between ‘signup_flow’ and ’age’.



It’s showing a non-linear relation that can be captured by a decision tree or random forest regressor.

Handling Imbalanced Data.

imbalanced data may lead to unstable training and low score when dealing with scoring metrics such as precision, recall or NDCG.

we can handle this issue with either under-sampling to the most represented classes or oversampling to the least ones.

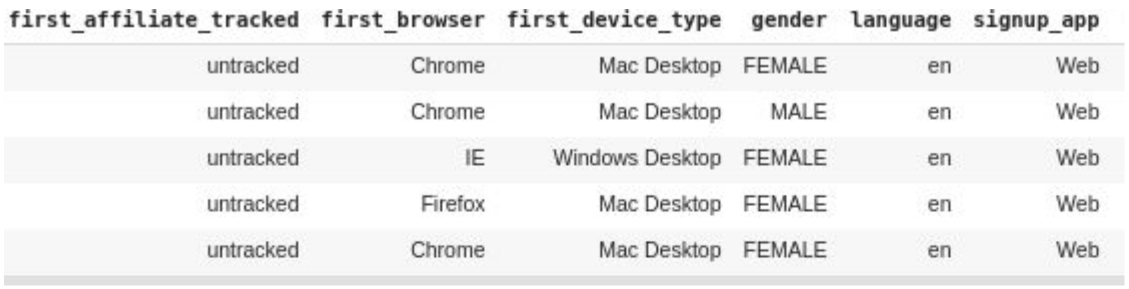
- under-sampling could be done randomly by removing random samples, or using algorithms like Tomek links or Cluster Centroids.

- oversampling could be done by repeating minority class samples or using an algorithm like SMOTE

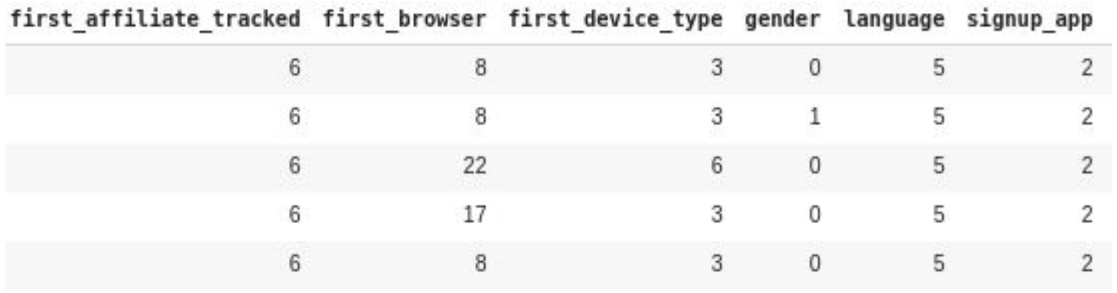
As we saw earlier that our data is imbalanced, So, I have used Tomeklinks to under-sample majority classes: 'NDF', 'US'. and SMOTE to oversample minority classes

Convert categories to numbers

The next step is to convert all categories variables to numerical values so we can input to our models



first_affiliate_tracked	first_browser	first_device_type	gender	language	signup_app
untracked	Chrome	Mac Desktop	FEMALE	en	Web
untracked	Chrome	Mac Desktop	MALE	en	Web
untracked	IE	Windows Desktop	FEMALE	en	Web
untracked	Firefox	Mac Desktop	FEMALE	en	Web
untracked	Chrome	Mac Desktop	FEMALE	en	Web



first_affiliate_tracked	first_browser	first_device_type	gender	language	signup_app
6	8	3	0	5	2
6	8	3	1	5	2
6	22	6	0	5	2
6	17	3	0	5	2
6	8	3	0	5	2

Approaches

I ‘ve implemented three approaches to solve this problem

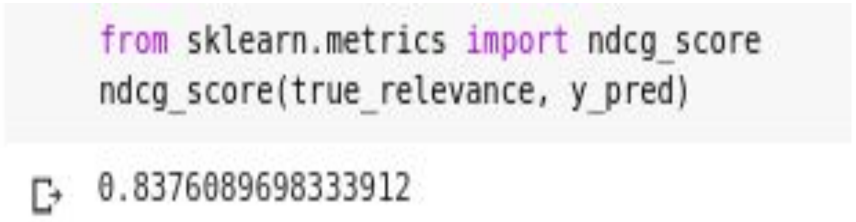
- XGBoost
- Random forest classifier
- Neural network

XGBoost:

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.

- I've implemented an XGBoost model with these hyperparameters.

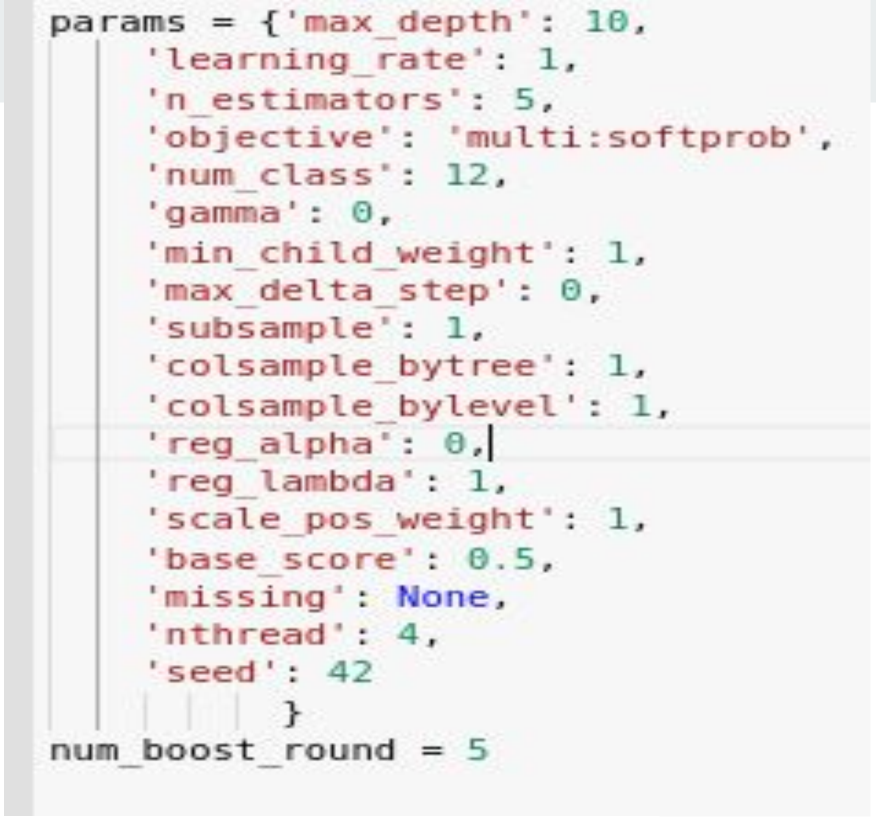
- It got NDCG score = 0.837 on training data



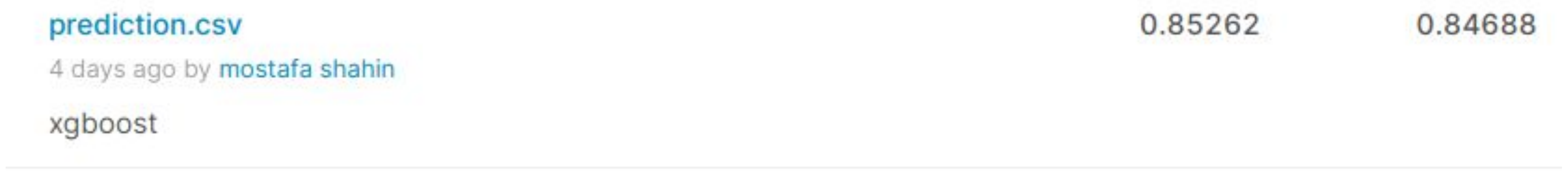
```
from sklearn.metrics import ndcg_score
ndcg_score(true_relevance, y_pred)
```

```
0.8376089698333912
```

- And got NDCG = 0.852 on kaggle submission



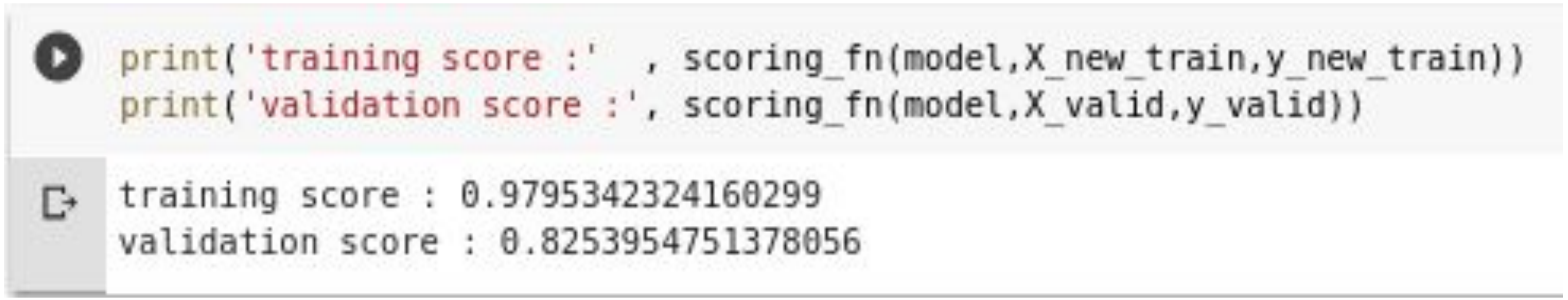
```
params = {'max_depth': 10,
          'learning_rate': 1,
          'n_estimators': 5,
          'objective': 'multi:softprob',
          'num_class': 12,
          'gamma': 0,
          'min_child_weight': 1,
          'max_delta_step': 0,
          'subsample': 1,
          'colsample_bytree': 1,
          'colsample_bylevel': 1,
          'reg_alpha': 0,
          'reg_lambda': 1,
          'scale_pos_weight': 1,
          'base_score': 0.5,
          'missing': None,
          'nthread': 4,
          'seed': 42
          }
num_boost_round = 5
```



prediction.csv 0.85262 0.84688
4 days ago by mostafa shahin
xgboost

Random forest classifier

I've implemented an rf classifier with 100 trees.

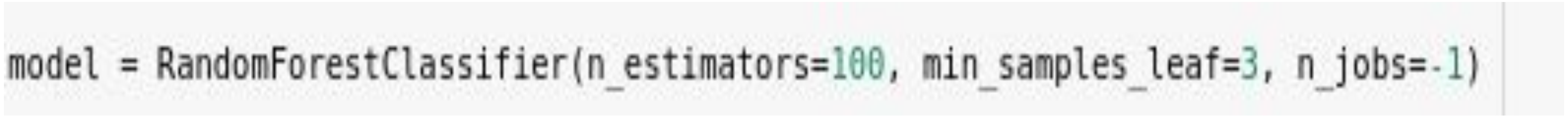


```
print('training score :'   , scoring_fn(model,X_new_train,y_new_train))
print('validation score :', scoring_fn(model,X_valid,y_valid))
```

```
training score : 0.9795342324160299
validation score : 0.8253954751378056
```

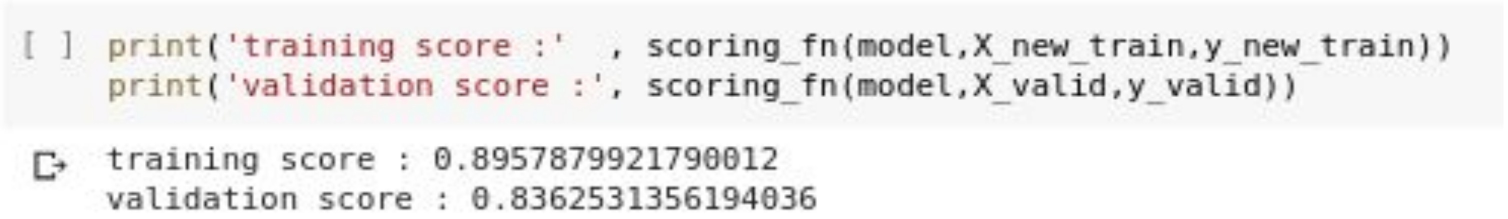
it did well on training-data, but it's not the case for validation-data.

then I set 'minimum samples per leaf' to 3



```
model = RandomForestClassifier(n_estimators=100, min_samples_leaf=3, n_jobs=-1)
```

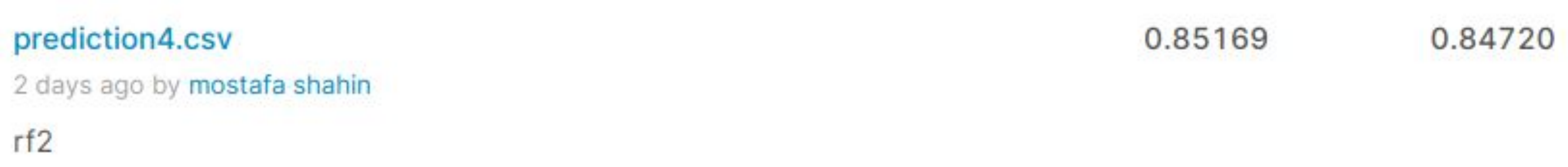
validation score had slightly increased.



```
[ ] print('training score :'  , scoring_fn(model,X_new_train,y_new_train))
    print('validation score :', scoring_fn(model,X_valid,y_valid))

training score : 0.8957879921790012
validation score : 0.8362531356194036
```

Kaggle submision NDCG = 0.851



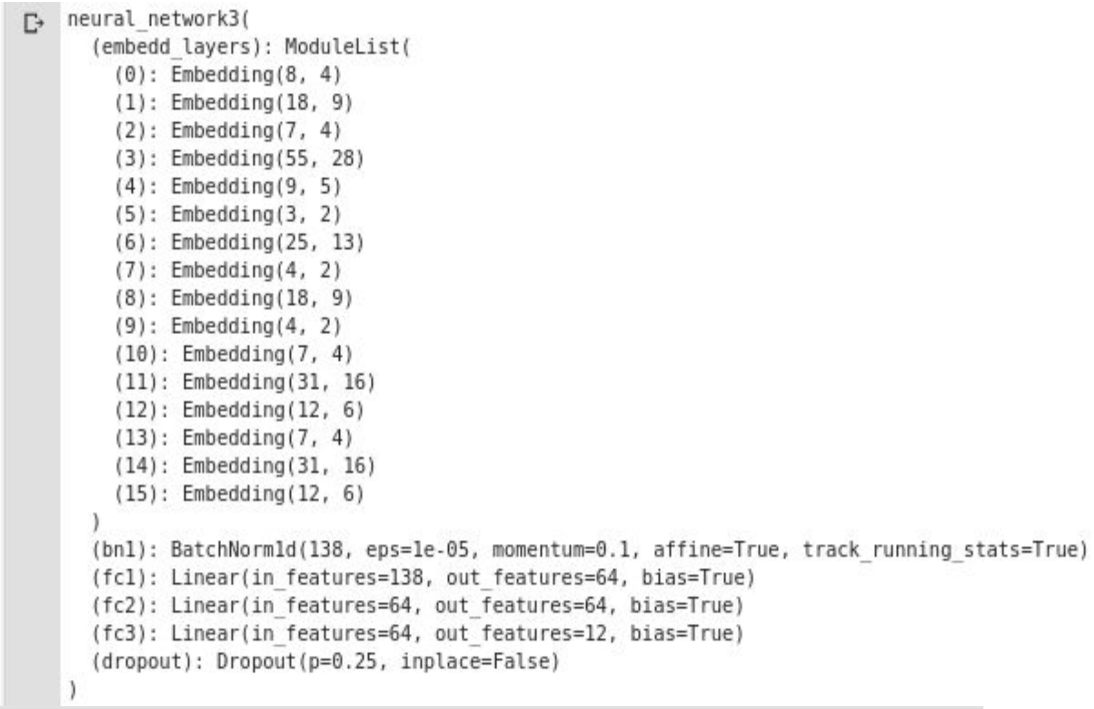
prediction4.csv 2 days ago by mostafa shahin rf2	0.85169	0.84720

Neural network

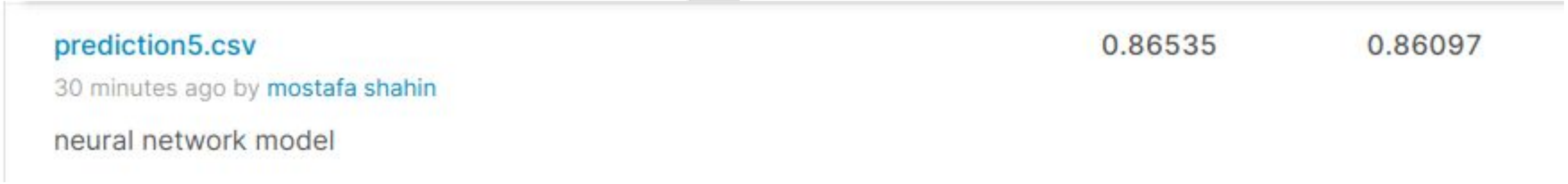
- usually, a neural network approach is mostly used for unstructured data problems but with the help of 'Entity Embedding' concept, we can use it for structured data. So, we can benefit from auto feature extraction that the neural network can introduce.

- I've implemented a neural network with a number of embedding layers(one for each categorical variable) prior to network hidden layers.

a simple network that consists of the embedding layers and three hidden layers got an NDCG score = 0.865 (higher than the previous two approaches)



```
neural_network3(
  (embedd_layers): ModuleList(
    (0): Embedding(8, 4)
    (1): Embedding(18, 9)
    (2): Embedding(7, 4)
    (3): Embedding(55, 28)
    (4): Embedding(9, 5)
    (5): Embedding(3, 2)
    (6): Embedding(25, 13)
    (7): Embedding(4, 2)
    (8): Embedding(18, 9)
    (9): Embedding(4, 2)
    (10): Embedding(7, 4)
    (11): Embedding(31, 16)
    (12): Embedding(12, 6)
    (13): Embedding(7, 4)
    (14): Embedding(31, 16)
    (15): Embedding(12, 6)
  )
  (bn1): BatchNorm1d(138, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (fc1): Linear(in_features=138, out_features=64, bias=True)
  (fc2): Linear(in_features=64, out_features=64, bias=True)
  (fc3): Linear(in_features=64, out_features=12, bias=True)
  (dropout): Dropout(p=0.25, inplace=False)
)
```



prediction5.csv 30 minutes ago by mostafa shahin neural network model	0.86535	0.86097