import library

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_log_error,accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
```

import dataset

In [2]:

```
df = pd.read_csv("C:\\Users\\Pratima Dhar\\Downloads\\ParticipantData_BTPC\\Participant
Data_BTPC\\Train.csv")
df.head()
```

Out[2]:

	session_id	session_number	client_agent	device_details	date
0	57f879e70d3c5fc2a98102d64c9fd84e	715	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKi...	Desktop - Chrome	2020-01-22
1	a5442b0c7c33d0a811e7661e556b2de8	55	Product/8.0 iPhone/8.1.3	iPhone - iOS	2020-02-27
2	305cb1486ed8610c00b37007926cb2c4	11	Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_2 like ...	iPhone - MobileWeb	2019-08-01
3	f2c1ecc9993f0071df91ba178450498c	2794	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Desktop - IE	2019-12-30
4	e460830ae295e55d2216ebdc761ab9a6	3674	Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_1 like ...	iPhone - Web	2019-09-10

In [3]:

```
df.shape
```

Out[3]:

```
(5429, 9)
```

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5429 entries, 0 to 5428
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   session_id      5429 non-null   object 
 1   session_number  5429 non-null   int64  
 2   client_agent    5269 non-null   object 
 3   device_details  5429 non-null   object 
 4   date            5429 non-null   object 
 5   purchased       5429 non-null   int64  
 6   added_in_cart   5429 non-null   int64  
 7   checked_out     5429 non-null   int64  
 8   time_spent      5429 non-null   float64
dtypes: float64(1), int64(4), object(4)
memory usage: 381.9+ KB
```

count null values

In [5]:

```
df.isna().sum()
```

Out[5]:

```
session_id          0
session_number      0
client_agent      160
device_details      0
date                0
purchased           0
added_in_cart       0
checked_out         0
time_spent          0
dtype: int64
```

find time_spent per device

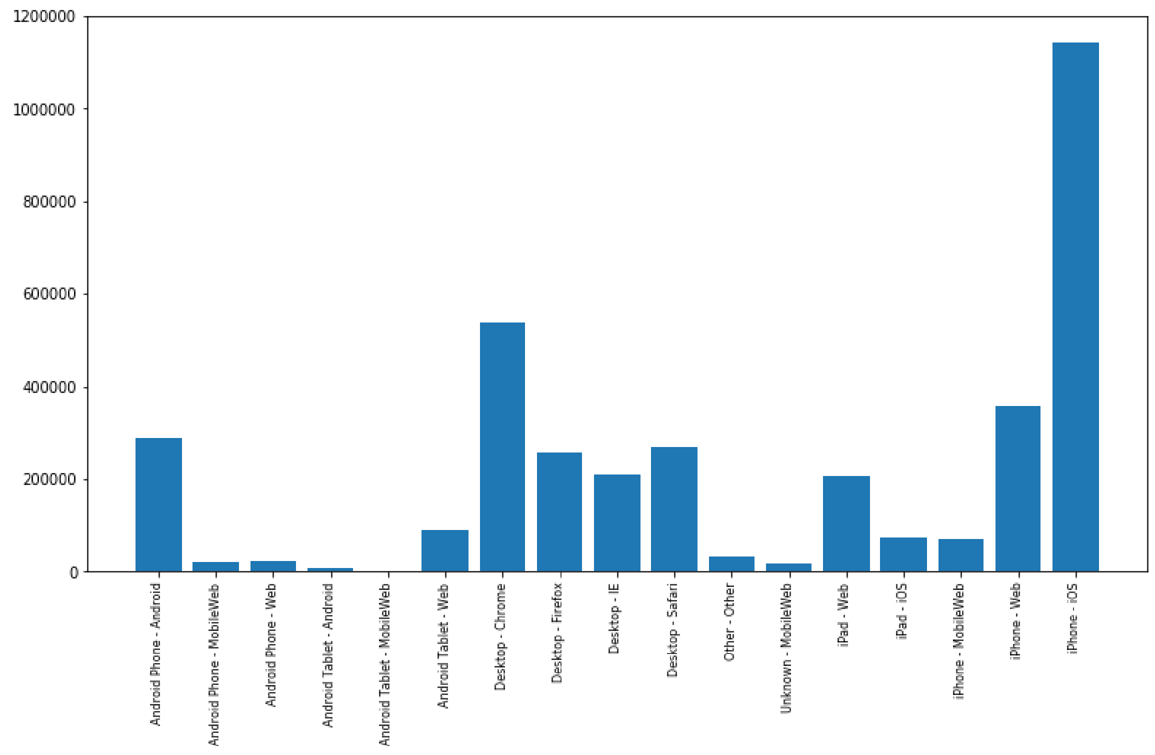
In [6]:

```
device = df.groupby('device_details')
time_per_device = device.sum()['time_spent']
print(time_per_device)
```

```
device_details
Android Phone - Android        287874.268
Android Phone - MobileWeb       19749.579
Android Phone - Web             23179.973
Android Tablet - Android         7389.468
Android Tablet - MobileWeb        665.728
Android Tablet - Web            90217.370
Desktop - Chrome               538223.785
Desktop - Firefox              255377.791
Desktop - IE                   209225.185
Desktop - Safari               267870.699
Other - Other                   32622.676
Unknown - MobileWeb             17508.966
iPad - Web                     205548.311
iPad - iOS                      72787.232
iPhone - MobileWeb              70814.752
iPhone - Web                   358147.685
iPhone - iOS                  1143278.341
Name: time_spent, dtype: float64
```

In [7]:

```
keys = [pair for pair, df in device]
plt.figure(figsize = (13,7))
plt.bar(keys, time_per_device)
plt.xticks(keys, rotation='vertical', size=8)
plt.show()
```



Find time_spent per month

In [8]:

```
df["date"] = df["date"].astype("str")
df["Month"] = df["date"].str[5:7]
df["Month"] = df["Month"].astype("int32")
df.head()
```

Out[8]:

	session_id	session_number	client_agent	device_details	date
0	57f879e70d3c5fc2a98102d64c9fd84e	715	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKi...	Desktop - Chrome	2020-01-22
1	a5442b0c7c33d0a811e7661e556b2de8	55	Product/8.0 iPhone/8.1.3	iPhone - iOS	2020-02-27
2	305cb1486ed8610c00b37007926cb2c4	11	Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_2 like ...	iPhone - MobileWeb	2019-08-01
3	f2c1ecc9993f0071df91ba178450498c	2794	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Desktop - IE	2019-12-30
4	e460830ae295e55d2216ebdc761ab9a6	3674	Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_1 like ...	iPhone - Web	2019-09-10

In [9]:

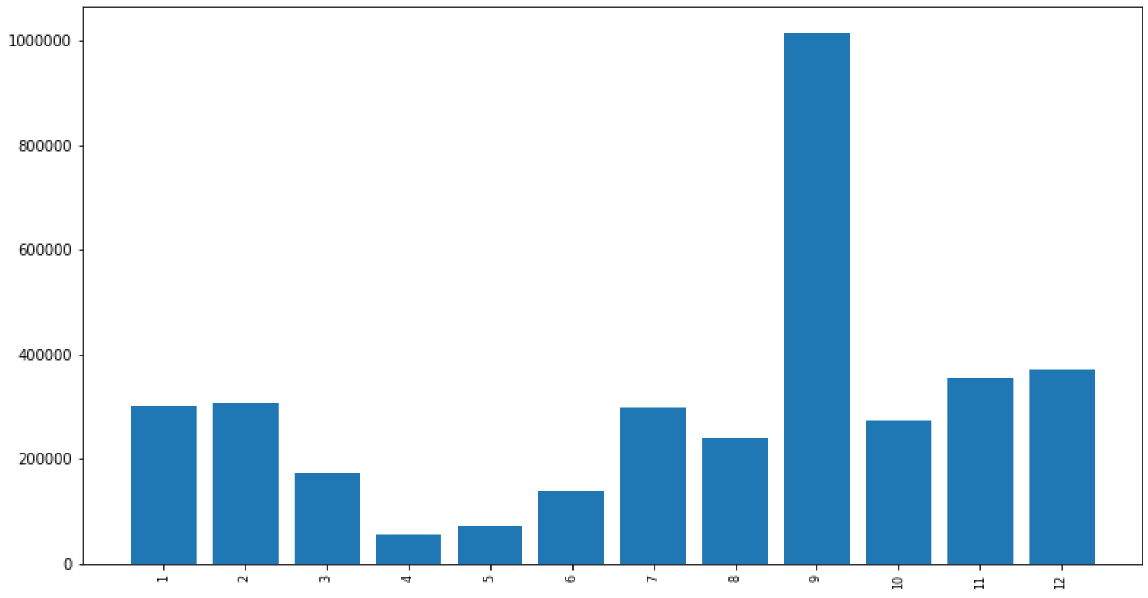
```
months = df.groupby("Month")
monthly_time = months.sum()["time_spent"]
display(monthly_time)
```

```
Month
1        301852.533
2        306616.300
3        173380.530
4         56013.231
5         72951.036
6        138978.605
7        299795.474
8        238769.650
9       1014014.472
10       273100.187
11       353720.864
12       371288.927
Name: time_spent, dtype: float64
```

In [10]:

```
plt.figure(figsize = (13,7))
keys2 = [pair for pair, df in months]
plt.figure(figsize = (13,7))
plt.bar(keys2, monthly_time)
plt.xticks(keys2, rotation='vertical', size=8)
plt.show()
```

```
<Figure size 936x504 with 0 Axes>
```



change data type

In [11]:

```
le = LabelEncoder()
def FunLabelEncoder(df):
    for c in df.columns:
        if df.dtypes[c] == object:
            le.fit(df[c].astype(str))
            df[c] = le.transform(df[c].astype(str))
    return df
```

In [12]:

```
df1 = df.copy()
df1 = FunLabelEncoder(df1)
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5429 entries, 0 to 5428
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   session_id      5429 non-null   int32  
 1   session_number  5429 non-null   int64  
 2   client_agent    5429 non-null   int32  
 3   device_details  5429 non-null   int32  
 4   date            5429 non-null   int32  
 5   purchased       5429 non-null   int64  
 6   added_in_cart   5429 non-null   int64  
 7   checked_out     5429 non-null   int64  
 8   time_spent      5429 non-null   float64
 9   Month           5429 non-null   int32  
dtypes: float64(1), int32(5), int64(4)
memory usage: 318.2 KB
```

Train the model

1. randomforestregression

In [13]:

```
subset_nfl_df = df1["client_agent"]
df1["client_agent"] = subset_nfl_df.fillna(method='bfill', axis=0).fillna(0)
print(subset_nfl_df.isna().sum())
#
features = ["session_number","client_agent","device_details","date","purchased","added_
in_cart","checked_out"]
#
X = df1[features]
y = df1.time_spent

# Splitting the data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,random_state
= 33)

# Training the data and predicting time_spent
#preprocessor = StandardScaler()
#X_train = preprocessor.fit_transform(X_train)
#X_test = preprocessor.transform(X_test)
clf = RandomForestRegressor(n_estimators =300)
clf.fit(X_train,y_train)
y_predict = clf.predict(X_test)

# Calculating error

rmse = np.sqrt(mean_squared_log_error(y_test,y_predict))
print(rmse)
```

```
0
1.6544652029067874
```

Loading and predicting test data

In [14]:

```
test = pd.read_csv("C:\\Users\\Pratima Dhar\\Downloads\\ParticipantData_BTPC\\Test.csv"
)
test = FunLabelEncoder(test)
predictions = clf.predict(test[features])
```

In [15]:

```
predictions
```

Out[15]:

```
array([1250.71178   ,  464.73964667, 1535.74865   , ...,   53.46228667,
        720.76159   , 1830.01784667])
```

In [16]:

```
#Create a  DataFrame
submission = pd.DataFrame({'time_spent':predictions})


#Visualize the first 10 rows
submission.head(10)
s1=submission.to_csv('C:\\Users\\Pratima Dhar\\Downloads\\ParticipantData_BTPC\\submiss
ion submission.csv',index=False)
```

In [17]:

```
s1 =pd.read_csv('submission.csv')
s1.head()
```

Out[17]:

	time_spent
0	1093.695597
1	231.990423
2	1231.928629
3	275.816632
4	254.039661

In [18]:

```
s2 =pd.read_csv("C:\\Users\\Pratima Dhar\\Downloads\\ParticipantData_BTPC\\Sample Submi
ssion.csv")
s2.head()
```

Out[18]:

	time_spent
0	0
1	0
2	0
3	0
4	0

1. xgboostregressor

In [19]:

```
from sklearn.model_selection import GridSearchCV,KFold
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder , OneHotEncoder
from sklearn.metrics import mean_squared_log_error

# np.sqrt(mean_squared_log_error(actual, predicted))

import xgboost as xgb
```

In [20]:

```
subset_nfl_df = df1["client_agent"]
df1["client_agent"] = subset_nfl_df.fillna(method='bfill', axis=0).fillna(0)
print(subset_nfl_df.isna().sum())
#
features = ["session_number","client_agent","device_details","date","purchased","added_
in_cart","checked_out"]
#
X = df1[features]
y = df1.time_spent

# Splitting the data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,random_state
= 33)
xgr = xgb.XGBRegressor(booster='gbtree',n_estimators=100)
xgr.fit(X_train,y_train,eval_set=[((X_train,y_train))],eval_metric="rmse",verbose=2)
ans = abs(xgr.predict(X_test))
score = np.sqrt(mean_squared_log_error(y_test, ans))
score
```

```
0
[20:17:25] WARNING: src/objective/regression_obj.cu:152: reg:linear is now
deprecated in favor of reg:squarederror.
[0]	validation_0-rmse:1801.52
[2]	validation_0-rmse:1720.57
[4]	validation_0-rmse:1664.63
[6]	validation_0-rmse:1623.71
[8]	validation_0-rmse:1580.79
[10]	validation_0-rmse:1546.41
[12]	validation_0-rmse:1521.4
[14]	validation_0-rmse:1497.77
[16]	validation_0-rmse:1478.55
[18]	validation_0-rmse:1463.88
[20]	validation_0-rmse:1455.17
[22]	validation_0-rmse:1443.33
[24]	validation_0-rmse:1429.74
[26]	validation_0-rmse:1421.47
[28]	validation_0-rmse:1402.05
[30]	validation_0-rmse:1386.4
[32]	validation_0-rmse:1380.84
[34]	validation_0-rmse:1366.96
[36]	validation_0-rmse:1354.93
[38]	validation_0-rmse:1346.05
[40]	validation_0-rmse:1331.48
[42]	validation_0-rmse:1320.64
[44]	validation_0-rmse:1317.71
[46]	validation_0-rmse:1312.81
[48]	validation_0-rmse:1305.93
[50]	validation_0-rmse:1297.72
[52]	validation_0-rmse:1288.62
[54]	validation_0-rmse:1286.7
[56]	validation_0-rmse:1277.06
[58]	validation_0-rmse:1274.36
[60]	validation_0-rmse:1266.12
[62]	validation_0-rmse:1260.75
[64]	validation_0-rmse:1250.52
[66]	validation_0-rmse:1249.4
[68]	validation_0-rmse:1246.37
[70]	validation_0-rmse:1241.6
[72]	validation_0-rmse:1239.19
[74]	validation_0-rmse:1235.07
[76]	validation_0-rmse:1233.11
[78]	validation_0-rmse:1229.68
[80]	validation_0-rmse:1227.79
[82]	validation_0-rmse:1223.16
[84]	validation_0-rmse:1220.32
[86]	validation_0-rmse:1216.66
[88]	validation_0-rmse:1212.88
[90]	validation_0-rmse:1210.45
[92]	validation_0-rmse:1209.54
[94]	validation_0-rmse:1208.8
[96]	validation_0-rmse:1208.17
[98]	validation_0-rmse:1205.51
[99]	validation_0-rmse:1205.21
```

Out[20]:

```
1.800953986048339
```

In [21]:

```
test = pd.read_csv("C:\\Users\\Pratima Dhar\\Downloads\\ParticipantData_BTPC\\Test.csv"
)
test = FunLabelEncoder(test)
predictions = xgr.predict(test[features])
```

In [22]:

```
predictions
```

Out[22]:

```
array([1051.6012  ,  272.35062 ,  951.54926 , ...,  -17.525023,
       1018.59436 ,  960.4123  ], dtype=float32)
```

In [23]:

```
#Create a  DataFrame
submission1 = pd.DataFrame({'time_spent':predictions})


#Visualize the first 10 rows
submission1.head(10)
s1=submission.to_csv('C:\\Users\\Pratima Dhar\\Downloads\\ParticipantData_BTPC\\submiss
ion1 submission1.csv',index=False)
```

In [24]:

```
s2 =pd.read_csv("C:\\Users\\Pratima Dhar\\Downloads\\ParticipantData_BTPC\\Sample Submi
ssion.csv")
s2.head()
```

Out[24]:

	time_spent
0	0
1	0
2	0
3	0
4	0

In [ ]:

```

```

In [ ]:

```

```