```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV
import imblearn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix,f1_score
dataset= pd.read_csv("flightdata.csv")
```

```
dataset.head()
```

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	...	CRS_ARR_
0	2016	1	1	1	5	DL	N836DN	1399	10397	ATL	...	
1	2016	1	1	1	5	DL	N964DN	1476	11433	DTW	...	
2	2016	1	1	1	5	DL	N813DN	1597	10397	ATL	...	
3	2016	1	1	1	5	DL	N587NW	1768	14747	SEA	...	
4	2016	1	1	1	5	DL	N836DN	1823	14747	SEA	...	

5 rows × 26 columns

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11231 entries, 0 to 11230
Data columns (total 26 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   YEAR                 11231 non-null  int64  
 1   QUARTER              11231 non-null  int64  
 2   MONTH                11231 non-null  int64  
 3   DAY_OF_MONTH         11231 non-null  int64  
 4   DAY_OF_WEEK          11231 non-null  int64  
 5   UNIQUE_CARRIER       11231 non-null  object 
 6   TAIL_NUM             11231 non-null  object 
 7   FL_NUM               11231 non-null  int64  
 8   ORIGIN_AIRPORT_ID    11231 non-null  int64  
 9   ORIGIN               11231 non-null  object 
 10  DEST_AIRPORT_ID      11231 non-null  int64  
 11  DEST                 11231 non-null  object 
 12  CRS_DEP_TIME         11231 non-null  int64  
 13  DEP_TIME             11124 non-null  float64
 14  DEP_DELAY            11124 non-null  float64
 15  DEP_DEL15            11124 non-null  float64
 16  CRS_ARR_TIME         11231 non-null  int64  
 17  ARR_TIME             11116 non-null  float64
 18  ARR_DELAY            11043 non-null  float64
 19  ARR_DEL15            11043 non-null  float64
 20  CANCELLED            11231 non-null  float64
 21  DIVERTED             11231 non-null  float64
 22  CRS_ELAPSED_TIME     11231 non-null  float64
 23  ACTUAL_ELAPSED_TIME  11043 non-null  float64
 24  DISTANCE             11231 non-null  float64
 25  Unnamed: 25          0 non-null      float64
dtypes: float64(12), int64(10), object(4)
memory usage: 2.2+ MB
```

```
dataset = dataset.drop('Unnamed: 25', axis=1)
dataset.isnull().sum()
```

```
YEAR                     0
QUARTER                  0
MONTH                    0
DAY_OF_MONTH             0
DAY_OF_WEEK              0
UNIQUE_CARRIER           0
TAIL_NUM                 0
FL_NUM                   0
ORIGIN_AIRPORT_ID        0
ORIGIN                   0
DEST_AIRPORT_ID          0
DEST                     0
CRS_DEP_TIME             0
DEP_TIME               107
DEP_DELAY              107
DEP_DEL15              107
CRS_ARR_TIME             0
ARR_TIME               115
ARR_DELAY              188
ARR_DEL15              188
CANCELLED                0
DIVERTED                 0
CRS_ELAPSED_TIME         0
ACTUAL_ELAPSED_TIME    188
DISTANCE                 0
dtype: int64
```

```
dataset = dataset[[ "MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK", "ORIGIN", "DEST", "CRS_ARR_TIME", "ARR_DEL15"]]
dataset.isnull().sum()
```

```
MONTH             0
DAY_OF_MONTH      0
DAY_OF_WEEK       0
ORIGIN            0
DEST              0
CRS_ARR_TIME      0
ARR_DEL15       188
dtype: int64
```

```
dataset[dataset.isnull().values.any(axis=1)].head()
```

	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	ARR_DEL15
177	1	9	6	MSP	SEA	852	NaN
179	1	10	7	MSP	DTW	1632	NaN
184	1	10	7	MSP	DTW	912	NaN
210	1	10	7	DTW	MSP	1303	NaN
478	1	22	5	SEA	JFK	723	NaN

```
dataset = dataset.fillna({'ARR_DEL15': 1})
dataset.iloc[177:185]
```

	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	ARR_DEL15
177	1	9	6	MSP	SEA	852	1.0
178	1	9	6	DTW	JFK	1724	0.0
179	1	10	7	MSP	DTW	1632	1.0
180	1	10	7	DTW	MSP	1649	0.0
181	1	10	7	JFK	ATL	1600	0.0
182	1	10	7	JFK	ATL	849	0.0
183	1	10	7	JFK	SEA	1945	0.0
184	1	10	7	MSP	DTW	912	1.0

```
dataset.head()
```

	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	ARR_DEL15
0	1	1	5	ATL	SEA	2143	0.0
1	1	1	5	DTW	MSP	1435	0.0
2	1	1	5	ATL	SEA	1215	0.0
3	1	1	5	SEA	MSP	1335	0.0
4	1	1	5	SEA	DTW	607	0.0

```
import math
```

```
for index, row in dataset.iterrows():
  dataset.loc[index,'CRS_ARR_TIME'] = math.floor(row['CRS_ARR_TIME']/100)
```

```
dataset.head()
```

	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	ARR_DEL15
0	1	1	5	ATL	SEA	21	0.0
1	1	1	5	DTW	MSP	14	0.0
2	1	1	5	ATL	SEA	12	0.0
3	1	1	5	SEA	MSP	13	0.0
4	1	1	5	SEA	DTW	6	0.0

```
dataset = pd.get_dummies(dataset, columns=['ORIGIN', 'DEST'])
dataset.head()
```

	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	CRS_ARR_TIME	ARR_DEL15	ORIGIN_ATL	ORIGIN_DTW	ORIGIN_JFK	ORIGIN_MSP	ORIGIN_SEA	DES
0	1	1	5	21	0.0	1	0	0	0	0	
1	1	1	5	14	0.0	0	1	0	0	0	
2	1	1	5	12	0.0	1	0	0	0	0	
3	1	1	5	13	0.0	0	0	0	0	1	
4	1	1	5	6	0.0	0	0	0	0	1	

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(dataset.drop('ARR_DEL15',axis=1),dataset['ARR_DEL15'],test_size=0.2,random_state=42)
```

```
x_train.shape
```

```
(8984, 14)
```

```
x_test.shape
```

```
(2247, 14)
```

```
from  sklearn.ensemble import RandomForestClassifier
```

```
model = RandomForestClassifier(random_state=13)
model.fit(x_train, y_train)
```

```
▾        RandomForestClassifier
RandomForestClassifier(random_state=13)
```

```
predicted = model.predict(x_test)
model.score(x_test, y_test)
```

```
0.8633733867378727
```

```
from sklearn.metrics import roc_auc_score
```

```
probabilities = model.predict_proba(x_test)
```

```
roc_auc_score(y_test, probabilities[:, 1])
```

```
0.6847853166272488
```

```
from sklearn.metrics import confusion_matrix
```

```
confusion_matrix(y_test, predicted)
```

```
array([[1906,   30],
       [ 277,   34]])
```

```
from sklearn.metrics import precision_score
```

```
train_predictions = model.predict(x_train)
precision_score(y_train, train_predictions)
```

```
1.0
```

```
from sklearn.metrics import recall_score
```

```
recall_score(y_train, train_predictions)
```

```
0.9984025559105432
```

```
import pickle
```

```
pickle.dump(model,open('flight.pkl','wb'))
```

```
model_loaded = pickle.load(open('flight.pkl', 'rb'))
```

```
model_loaded.predict(x_test)
```

```
array([0., 0., 0., ..., 0., 0., 0.])
```

```
import joblib
```

```
joblib.dump(model,'flight2')
```

```
['flight2']
```