```
import seaborn as sns
import numpy
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import joblib
from sklearn.metrics import accuracy_score
```

```
dataset=pd.read_csv(r"/content/flightdata.csv")
```

```
dataset.head()
```

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	TAIL_NUM	FL_NUM
0	2016	1	1	1	5	DL	N836DN	1399
1	2016	1	1	1	5	DL	N964DN	1476
2	2016	1	1	1	5	DL	N813DN	1597
3	2016	1	1	1	5	DL	N587NW	1768
4	2016	1	1	1	5	DL	N836DN	1823

5 rows × 26 columns

```
dataset.isnull().any()
```

```
YEAR                   False
QUARTER                False
MONTH                  False
DAY_OF_MONTH           False
DAY_OF_WEEK            False
UNIQUE_CARRIER         False
TAIL_NUM               False
FL_NUM                 False
ORIGIN_AIRPORT_ID      False
ORIGIN                 False
DEST_AIRPORT_ID        False
DEST                   False
CRS_DEP_TIME           False
```

```
    DEP_TIME               True
    DEP_DELAY              True
    DEP_DEL15              True
    CRS_ARR_TIME          False
    ARR_TIME               True
    ARR_DELAY              True
    ARR_DEL15              True
    CANCELLED             False
    DIVERTED              False
    CRS_ELAPSED_TIME      False
    ACTUAL_ELAPSED_TIME    True
    DISTANCE              False
    Unnamed: 25            True
    dtype: bool
```

```
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
    Unnamed: 25           11231
    dtype: int64
```

```
dataset.columns
```

```
    Index(['YEAR', 'QUARTER', 'MONTH', 'DAY_OF_MONTH', 'DAY_OF_WEEK',
           'UNIQUE_CARRIER', 'TAIL_NUM', 'FL_NUM', 'ORIGIN_AIRPORT_ID', 'ORIGIN',
           'DEST_AIRPORT_ID', 'DEST', 'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY',
           'DEP_DEL15', 'CRS_ARR_TIME', 'ARR_TIME', 'ARR_DELAY', 'ARR_DEL15',
           'CANCELLED', 'DIVERTED', 'CRS_ELAPSED_TIME', 'ACTUAL_ELAPSED_TIME',
           'DISTANCE', 'Unnamed: 25'],
          dtype='object')
```

```
dataset.ARR_DEL15.value_counts()
```

```
    0.0    9668
    1.0    1375
    Name: ARR_DEL15, dtype: int64
```

```
dataset['DEST'].unique()
```

```
    array(['SEA', 'MSP', 'DTW', 'ATL', 'JFK'], dtype=object)
```

```
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
```

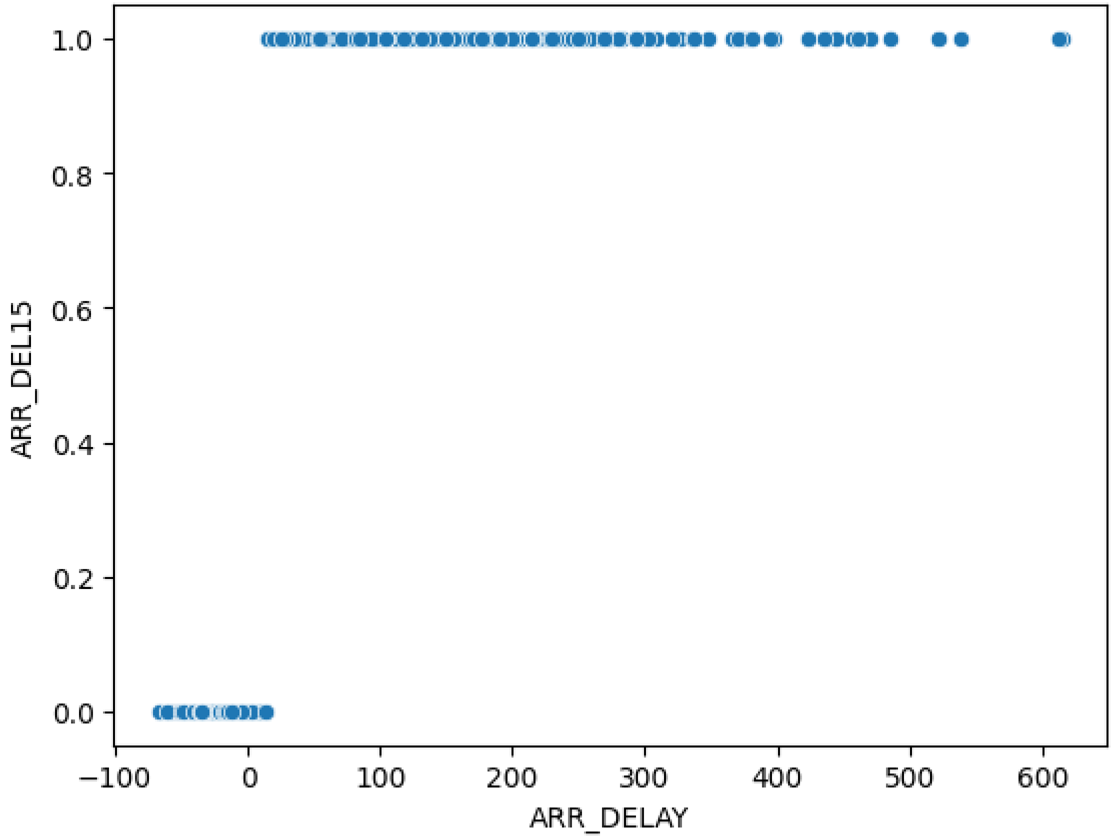
```
import seaborn as sns
%matplotlib inline
```

```
flight_data = pd.read_csv("flightdata.csv")
flight_data.describe()
```

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_NUM
count	11231.0	11231.000000	11231.000000	11231.000000	11231.000000	11231.000000
mean	2016.0	2.544475	6.628973	15.790758	3.960199	1334.325617
std	0.0	1.090701	3.354678	8.782056	1.995257	811.875227
min	2016.0	1.000000	1.000000	1.000000	1.000000	7.000000

```
sns.scatterplot(x='ARR_DELAY',y='ARR_DEL15', data=flight_data)
```

```
<Axes: xlabel='ARR_DELAY', ylabel='ARR_DEL15'>
```

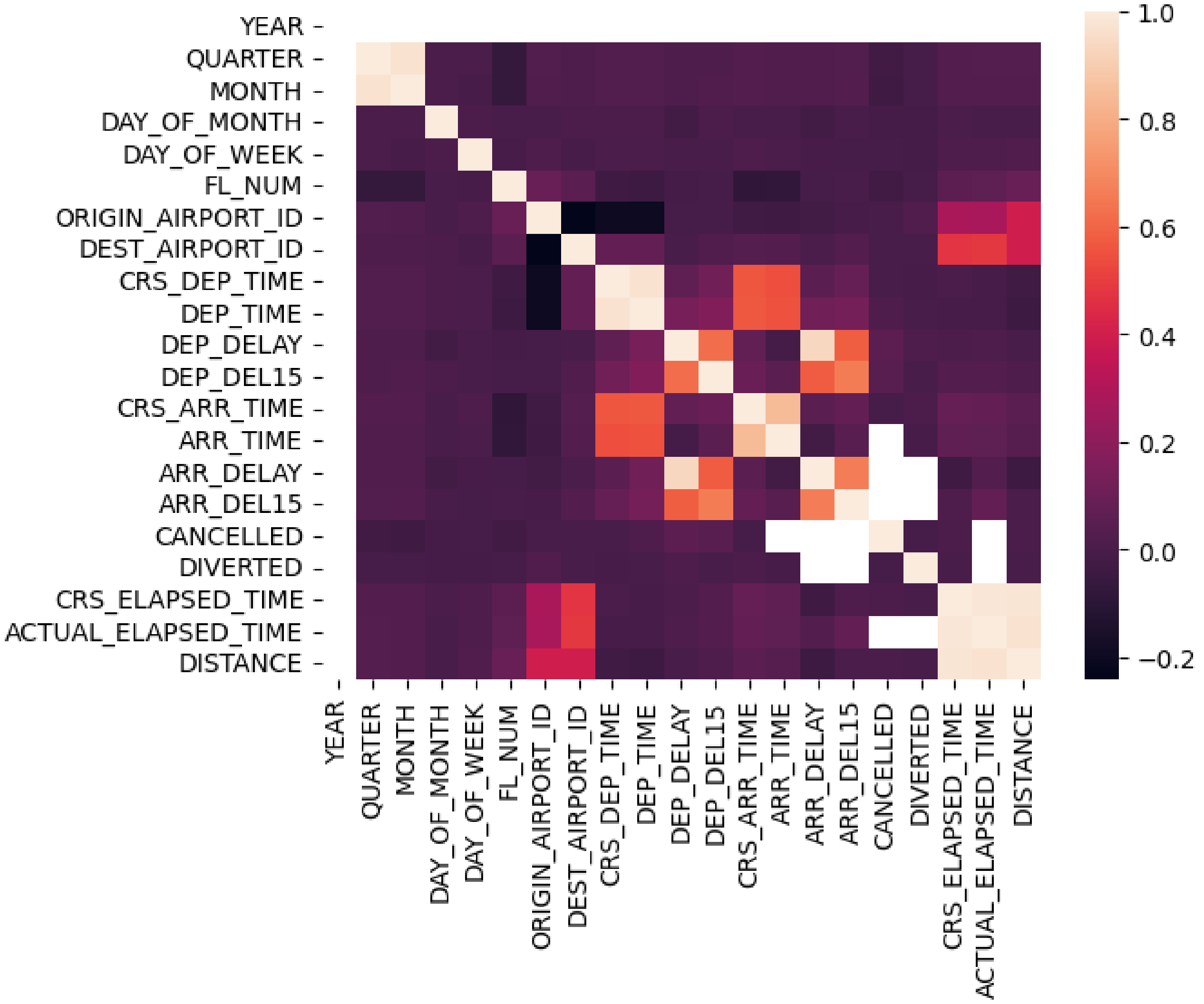


```
sns.catplot(x="ARR_DEL15",y="ARR_DELAY",kind='bar',data=flight_data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fdab72d0040>
```

```
sns.heatmap(dataset.corr())
```

```
<ipython-input-14-aa6664222663>:1: FutureWarning: The default value of numeric_only
  sns.heatmap(dataset.corr())
<Axes: >
```



```
dataset=dataset[["FL_NUM","MONTH","DAY_OF_MONTH","DAY_OF_WEEK","ORIGIN","DEST","CRS_ARR_TI
dataset.isnull().sum()
```

```
FL_NUM          0
MONTH           0
```

```
    DAY_OF_MONTH      0
    DAY_OF_WEEK       0
    ORIGIN            0
    DEST              0
    CRS_ARR_TIME      0
    DEP_DEL15       107
    ARR_DEL15       188
    dtype: int64
```

```
dataset[dataset.isnull().any(axis=1)].head(10)
```

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15
177	2834	1	9	6	MSP	SEA	852	0.0
179	86	1	10	7	MSP	DTW	1632	NaN
184	557	1	10	7	MSP	DTW	912	0.0
210	1096	1	10	7	DTW	MSP	1303	NaN
478	1542	1	22	5	SEA	JFK	723	NaN
481	1795	1	22	5	ATL	JFK	2014	NaN
491	2312	1	22	5	MSP	JFK	2149	NaN
499	423	1	23	6	JFK	ATL	1600	NaN
500	425	1	23	6	JFK	ATL	1827	NaN
501	427	1	23	6	JFK	SEA	1053	NaN

```
dataset['DEP_DEL15'].mode()
```

```
    0    0.0
    Name: DEP_DEL15, dtype: float64
```

```
dataset=dataset.fillna({'ARR_DEL15':1})
dataset=dataset.fillna({'DEP_DEL15':0})
dataset.iloc[177:185]
```

```
import math

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

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15
0	1399	1	1	5	ATL	SEA	21	0.0
1	1476	1	1	5	DTW	MSP	14	0.0
2	1597	1	1	5	ATL	SEA	12	0.0
3	1768	1	1	5	SEA	MSP	13	0.0
4	1823	1	1	5	SEA	DTW	6	0.0

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dataset['DEST']=le.fit_transform(dataset['DEST'])
dataset['ORIGIN']=le.fit_transform(dataset['ORIGIN'])
```

```
dataset['ORIGIN'].unique()
```

```
    array([0, 1, 4, 3, 2])
```

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

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15	ORI
0	1399	1	1	5	21	0.0	0.0	
1	1476	1	1	5	14	0.0	0.0	
2	1597	1	1	5	12	0.0	0.0	
3	1768	1	1	5	13	0.0	0.0	
4	1823	1	1	5	6	0.0	0.0	

```
dataset.iloc[:,8:9]
```

	ORIGIN_1
0	0
1	1
2	0
3	0
4	0
...	...
11226	1
11227	0
11228	1
11230	0

```
x = dataset.iloc[:, 0:8].values
y = dataset.iloc[:, 8:9].values
```

```
y
```

```
array([[0],
       [1],
       [0],
       ...,
       [1],
       [0],
       [0]], dtype=uint8)
```

```
x
```

```
array([[1.399e+03, 1.000e+00, 1.000e+00, ..., 0.000e+00, 0.000e+00,
        1.000e+00],
       [1.476e+03, 1.000e+00, 1.000e+00, ..., 0.000e+00, 0.000e+00,
        0.000e+00],
       [1.597e+03, 1.000e+00, 1.000e+00, ..., 0.000e+00, 0.000e+00,
        1.000e+00],
       ...,
       [1.823e+03, 1.200e+01, 3.000e+01, ..., 0.000e+00, 0.000e+00,
        0.000e+00],
       [1.901e+03, 1.200e+01, 3.000e+01, ..., 0.000e+00, 0.000e+00,
        1.000e+00],
       [2.005e+03, 1.200e+01, 3.000e+01, ..., 0.000e+00, 0.000e+00,
        1.000e+00]])
```

```
y
```

```
array([[0],
       [1],
       [0],
       ...,
       [1],
       [0],
       [0]], dtype=uint8)
```

```
x.shape
```

```
(11231, 8)
```

```
y.shape
```

```
(11231, 1)
```

```
from sklearn.preprocessing import OneHotEncoder
oh = OneHotEncoder()
z=oh.fit_transform(x[:,4:5]).toarray()
t=oh.fit_transform(x[:,5:6]).toarray()
#x=np.delete(x,[4,7],axis=1)
```

```
z
```

```
array([[0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

```
t
```

```
array([[1., 0.],
       [1., 0.],
       [1., 0.],
       ...,
       [1., 0.],
       [1., 0.],
       [1., 0.]])
```

```
x=np.delete(x,[4,5],axis=1)
```

```
x.shape
```

```
(11231, 6)
```

```
x=np.concatenate((t,z,x),axis=1)
```

```
x.shape
```

```
(11231, 29)
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

```
import pandas as pd
```

```
main_folder = r'/content/flightdata.csv' + '\\'
```

```
x_test.shape
```

```
(2247, 29)
```

```
x_train.shape
```

```
(8984, 29)
```

```
y_test.shape
```

```
(2247, 1)
```

```
y_train.shape
```

```
(8984, 1)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(random_state = 0)
classifier.fit(x_train,y_train)
```

```
▾        DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

```
decisiontree = classifier.predict(x_test)
```

```
decisiontree
```

```
array([1, 0, 0, ..., 1, 0, 0], dtype=uint8)
```

```
from sklearn.metrics import accuracy_score
desacc = accuracy_score(y_test,decisiontree)
```

```
desacc
```

```
0.9897641299510458
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,decisiontree)
```

```
cm
```

```
     array([[1790,   12],
            [  11,  434]])
```

```
import sklearn.metrics as metrics
fpr1 ,tpr1 ,threshold1 =metrics.roc_curve(y_test,decisiontree)
roc_auc1 = metrics.auc(fpr1,tpr1)
```

```
fpr1
```

```
     array([0.        , 0.00665927, 1.        ])
```

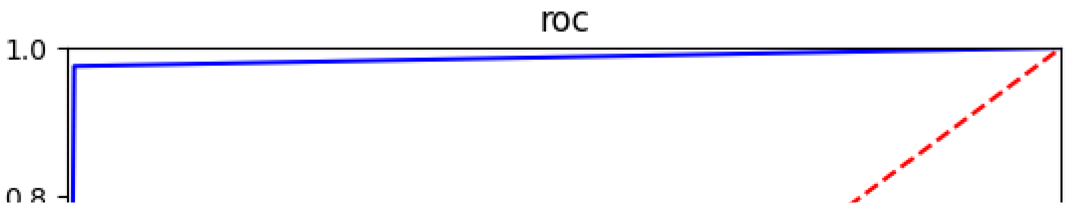
```
tpr1
```

```
     array([0.       , 0.9752809, 1.       ])
```

```
threshold1
```

```
     array([2, 1, 0])
```

```
import matplotlib.pyplot as plt
plt.title("roc")
plt.plot(fpr1,tpr1,'b',label = 'Auc = %0.2f'% roc_auc1)
plt.legend(loc = 'lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.xlabel('tpr')
plt.ylabel('fpr')
plt.show()
```



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

```
!pip install nbconvert
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: nbconvert in /usr/local/lib/python3.9/dist-packages (
Requirement already satisfied: jupyter-core>=4.7 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.9/dist
Requirement already satisfied: traitlets>=5.0 in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages (fro
Requirement already satisfied: defusedxml in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages (from
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.9/dist-
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.9/dist-p
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.9/dist-packag
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.9/dist-pack
Requirement already satisfied: pygments>=2.4.1 in /usr/local/lib/python3.9/dist-pack
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.9/dist-packages (f
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-pack
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: jupyter-client>=6.1.12 in /usr/local/lib/python3.9/di
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.9/dist-pack
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.9/dist-packag
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-packag
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.9/dist
Requirement already satisfied: tornado>=4.1 in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.9/dist-packages (
```

```
!jupyter nbconvert --to html flight.ipynb
```

```
[NbConvertApp] Converting notebook flight.ipynb to html
[NbConvertApp] Writing 845357 bytes to flight.html
```

```
!pip install flask-ngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (f
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: importlib-metadata>=3.6.0 in /usr/local/lib/python3.9
Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.9/dist-pack
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.9/dist-packages 
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dis
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-p
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-pack
Installing collected packages: flask-ngrok
Successfully installed flask-ngrok-0.0.25
```

✓ 3s completed at 09:44