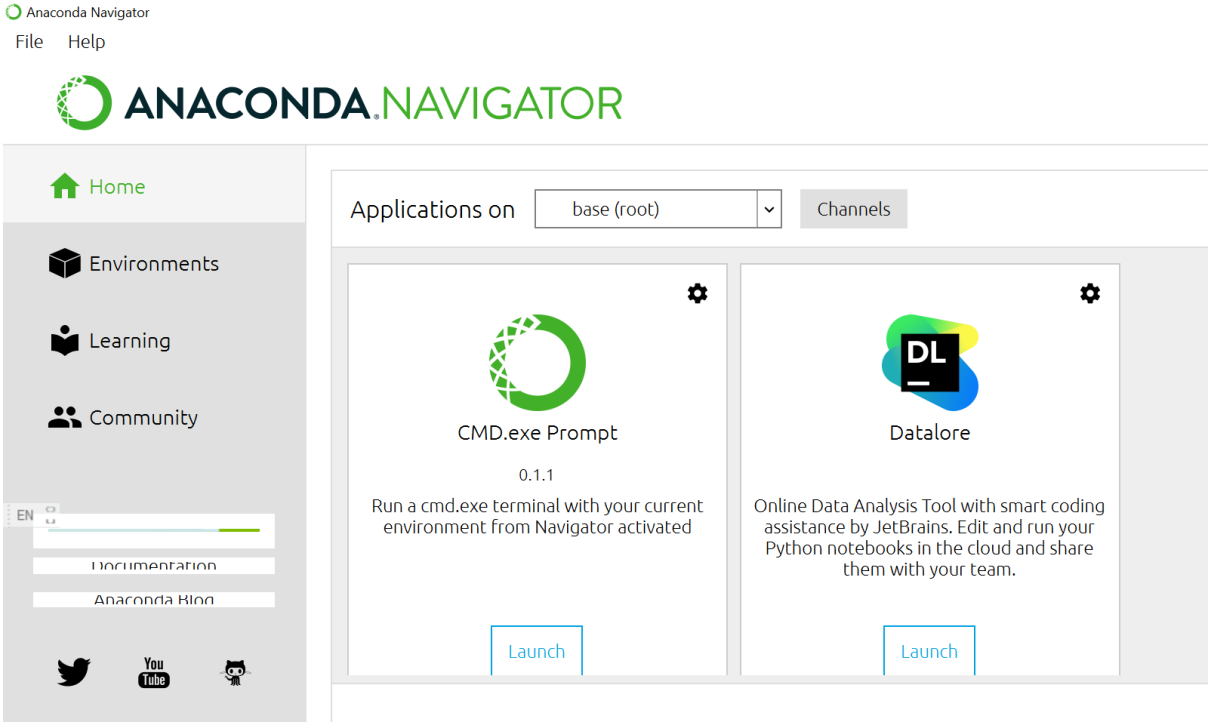
PREREQUISITES:



DOWNLOAD/ CREATE DATASET:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	YEAR	QUARTER	MONTH	DAY_OF_M	DAY_OF_W	UNIQUE_C	TAIL_NUM	FL_NUM	ORIGIN_A	ORIGIN	DEST_AIRI	DEST	CRS_DEP_	DEP_TIME	DEP_DELA	DEP_DEL1	CRS_ARR_	ARR_TIME
2	2016	1	1	1	5	DL	N836DN	1399	10397	ATL	14747	SEA	1905	1907	2	0	2143	2102
3	2016	1	1	1	5	DL	N964DN	1476	11433	DTW	13487	MSP	1345	1344	-1	0	1435	1439
4	2016	1	1	1	5	DL	N813DN	1597	10397	ATL	14747	SEA	940	942	2	0	1215	1142
5	2016	1	1	1	5	DL	N587NW	1768	14747	SEA	13487	MSP	819	820	1	0	1335	1345
6	2016	1	1	1	5	DL	N836DN	1823	14747	SEA	11433	DTW	2300	2256	-4	0	607	615
7	2016	1	1	1	5	DL	N936DL	1975	13487	MSP	10397	ATL	1129	1127	-2	0	1459	1441
8	2016	1	1	2	6	DL	N983DL	2074	10397	ATL	13487	MSP	1745	1745	0	0	1931	1920
9	2016	1	1	2	6	DL	N589NW	2151	13487	MSP	14747	SEA	1740	1751	11	0	1929	1908
10	2016	1	1	2	6	DL	N804DN	2221	13487	MSP	14747	SEA	1115	1115	0	0	1305	1255
11	2016	1	1	2	6	DL	N965DN	2291	13487	MSP	10397	ATL	1430	1443	13	0	1801	1800
12	2016	1	1	2	6	DL	N703TW	2350	10397	ATL	12478	JFK	825	828	3	0	1038	1029
13	2016	1	1	2	6	DL	N538US	2444	10397	ATL	14747	SEA	1345	1355	10	0	1621	1605
14	2016	1	1	2	6	DL	N699DL	2610	10397	ATL	13487	MSP	725	721	-4	0	904	903
15	2016	1	1	2	6	DL	N582NW	2826	11433	DTW	14747	SEA	835	841	6	0	1047	1023
16	2016	1	1	2	6	DL	N920DE	2845	11433	DTW	10397	ATL	1624	1622	-2	0	1830	1805
17	2016	1	1	3	7	DL	N960AT	86	13487	MSP	11433	DTW	1345	1337	-8	0	1620	1616
18	2016	1	1	3	7	DL	N3732J	423	12478	JFK	10397	ATL	1300	1258	-2	0	1538	1519

DATA PREPROCESSING AND MODEL BUILDING:

```
import sys
import numpy as np
import pandas as pd
import seaborn as sns
import pickle
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import sklearn.metrics as metrics
```

Handling Missing Values

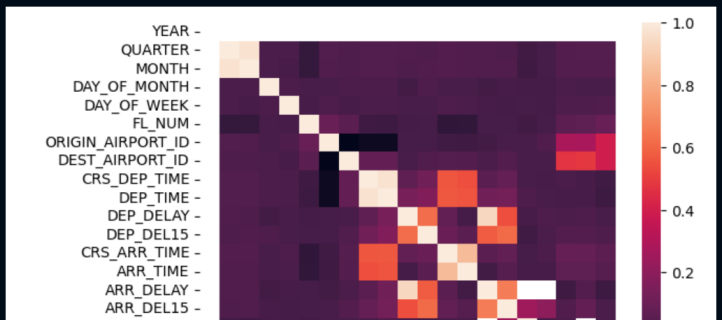
```
data.isnull().sum()
```

```
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
```

Data Visualisation

```
sns.heatmap(data.corr())
```

```
<AxesSubplot:>
```



Encoding

```
le = LabelEncoder()
data['DEST'] = le.fit_transform(data['DEST'])
data['ORIGIN'] = le.fit_transform(data['ORIGIN'])
data.head(5)
```

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

Splitting the dataset into X and Y

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

Python

```
data.head()
```

Python

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15	ORIGIN_0	ORIGIN_1	ORIGIN_2	ORIGIN_3	ORIGIN_4	DEST_0
0	1399	1	1	5	21	0.0	0.0	1	0	0	0	0	0
1	1476	1	1	5	14	0.0	0.0	0	1	0	0	0	0
2	1597	1	1	5	12	0.0	0.0	1	0	0	0	0	0
3	1768	1	1	5	13	0.0	0.0	0	0	0	0	1	0
4	1823	1	1	5	6	0.0	0.0	0	0	0	0	1	0

Splitting into Train and Test

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
```

```
x_train.shape
```

```
(8984, 16)
```

```
y_train.shape
```

```
(8984, 1)
```

+ Code + Markdown

Decision Tree

[30]

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

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

[31]

```
dt = dtc.predict(x_test)
dt
```

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

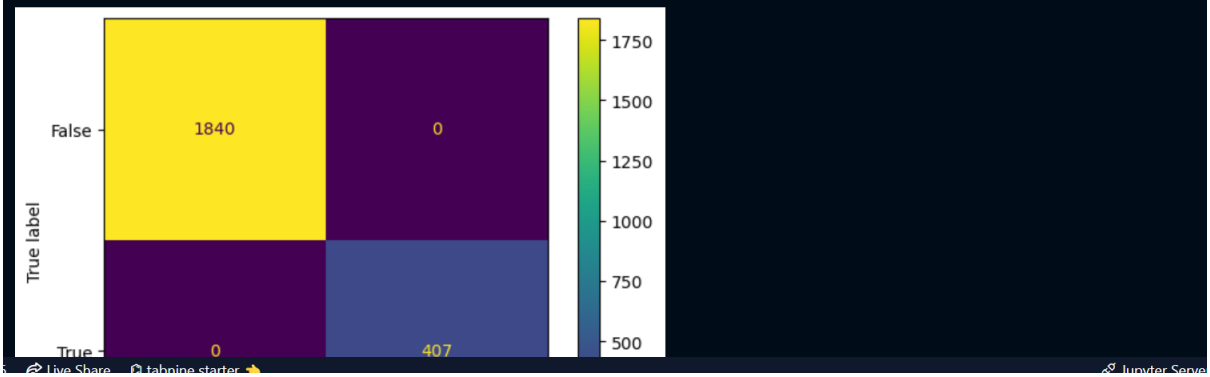
```
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test, dt)
acc
```

```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

confusion_matrix = metrics.confusion_matrix(y_test, dt)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
```



APPLICATION BUILDING:

```
from errno import ENAMETOOLONG
from turtle import st
from flask import Flask, render_template, request, redirect, url_for, session
import numpy as np
import pandas as pd
import pickle
import os
import requests
from markupsafe import escape
from flask_mail import Mail, Message
from random import randint
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.datab

GOOGLE_CLIENT_ID = "340644155083-hm83b3k5d7mbb0ps5u33ck7qkbder4uf.apps.googleusercontent.com"
GOOGLE_CLIENT_SECRET = "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"
REDIRECT_URI = '/gentry/auth'

import json
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "mYBrvKJylOO4wCWoS_TesMMELMxEBSW9rQ1NzP0Wn-se"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
model = pickle.load(open('flight.pkl','rb'))
app = Flask(__name__)

app.secret_key = 'flightdelayflyhigh2022'
```

```
dept = request.form['dept']
arrtime = request.form['arrtime']
actdept = request.form['actdept']

dept15 = int(dept) - int(actdept)
total = [[name, month, day_of_month, day_of_week,dept15, arrtime, origin1,origin2,origin3,origin4,origin5,destination1,dest
print(total)
payload_scoring = {"input_data": [{"field": ['f0', 'f1', 'f2', 'f3', 'f4', 'f5', 'f6', 'f7', 'f8','f9', 'f10', 'f11', 'f12'
response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/739a6f52-043e-49ec-b5ee-d9b2d5a4e6b6/
headers={'Authorization': 'Bearer ' + mltoken})
y_pred = response_scoring.json()
# y_pred = model.predict(total)
# print(y_pred)
pred_result = y_pred['predictions'][0]['values'][0][0]
print(pred_result)
if(pred_result == 0):
# print(y_pred)
```

```
<!DOCTYPE html>
<html>
<head>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-a
<script src="https://kit.fontawesome.com/9a03e12720.js" crossorigin="anonymous"></script>
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
  background-repeat: no-repeat;
  background-size: 100% 100%;
  background-attachment: fixed;
}

.navbar {
  overflow: hidden;
  background-color: none;
  padding:10px;
}

.navbar a {
  float: left;
  font-size: 16px;
  color: white;
  text-align: center;
  padding: 14px 16px;
```

```
</style>
</head>
<body background = "{{url_for('static',filename = 'flight.jpg')}}">
  <a href="#home"><img src="{{url_for('static',filename = 'newlog.png')}}" class="logo" align="right" title="FlyHigh"
<div class="navbar">
  <a href="#home">  <i class="fa-solid fa-house"></i>Home</a>
  <!--<a href="https://www.flightradar24.com/11.03,77.04/8">
    Flight Finder
  </a>-->
  <a href="{{url_for('prediction')}}"><i class="fa-solid fa-plane-circle-exclamation"></i>Prediction</a>
  <a href="{{url_for('signup')}}"><i class="fa-solid fa-right-to-bracket"></i></i>Sign Up</a>
</div>

<div class="wrap">
  <div class="search">
      <div class="dropdown">
          <button class="dropbtn">
              <input type class="searchTerm" placeholder="Select Your Airport">
          </button>
          <div class="dropdown-content">
              <a href="https://www.mspairport.com/flights-and-airlines/flights?text=&flight_type=arrival">MSP(Minneap
              <a href="https://www.detroit-airport.com/dtw-metro-departures">DTW(Detroit Metropolitan Wayne County Ai
              <a href="https://www.jfkairport.com/flight/airlines">JFK(John F. Kennedy International Airport)</a>
              <a href="https://www.portseattle.org/sea-tac">SEA(Seattle-Tacoma International Airport)</a>
              <a href="https://atlfly.com/airlines">ALT(Hartsfield-Jackson Atlanta International Airport)</a>
```