### **Application Of Machine Learning Algorithms To Predict Flight Delays**



**April-15th MLDL Team-60**

**Submitted By:**

**1)Arjun Bhanot**

**2)Anshul Ranjan**

**3)Mayank Nagpal**

**4)Toshik Langade**

**5)Uday Bhaskar**

**1. INTRODUCTION**

**1.1 Overview**

Over the last few years, air travel has been frequently favored amongsttravelers, chiefly because of its speed and in some cases comfort. Travelling from one country to another and in between the countries, it is the fastest mode of transportation. Most of the people, especially those based on Professional Workers are dependent on flights to reach the destination on time. The flights are known for their punctuality. However, within a few years, the increase in air travel leads to an increase in air traffic. This increase in air traffic results in massive levels of aircraft delays on the ground and the air. Flight delays not only have a financial impression but also harmful environmental impacts. Air-traffic management is becoming increasingly challenging. Flight delays hurt airlines, airports, and passengers. Predicting Flights delays is essential during the decision-making process for all players of commercial aeronautics. Moreover, the accurate prediction models for flight delays become necessarily important due to the complexity of the air transportation system.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1.2 Purpose**

Predicting flights delay is now very important for various purposes. Generating the model which can accurately predict the delay in flights is utmost needed. Using a machine learning model, it can be easy to predict flight arrival delays. The main aim is to display the delay in flight arrival time so that it can be very well coordinated with the effects of the delay in flights as what we see that in the case of US where the economic impact of flight delays for domestic flights is estimated to be more than $19 Billion per year to the airlines and over $41 Billion per year to the national economy. It will be easier for the airline company too to make sure whether the flight will be delayed or not. They can change their strategy that is related to arranging the space for passengers, carry out their luggage etc. The final achievement will be one which helps in time management for the flight company.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2. LITERATURE SURVEY**

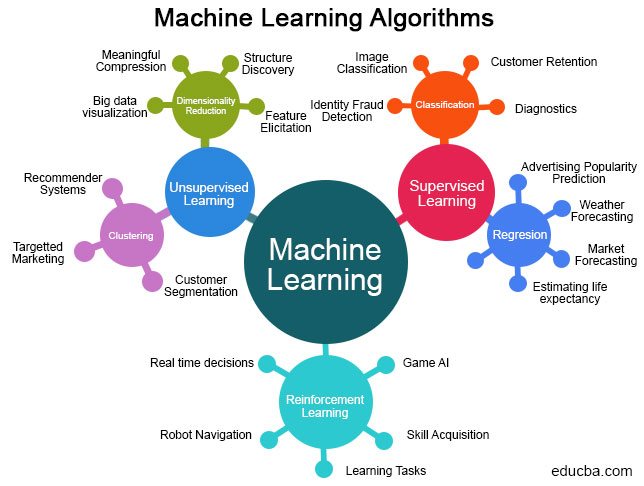
**2.1 Existing problem**

The major issue related to flights is to predict its delay. Air transportation is known for its speed and time. The major expectation of this transportation is that the passenger expects that they will reach the destination on time. Nowadays, due to the increase in heavy traffic, it is important to predict the delay in flights. It is necessary to find out the important criteria which can contribute to flight delays. The major area includes finding and measuring factors affecting aircraft delays on the ground and in the air and develop machine learning algorithms to optimize airline and airport operations based on the factor responsible for the flight delay. A method is required to measure the impact of the delays occurring at one airport on other airports. Another major area of study is to classify the factors accountable for aircraft taxi-delays that happen on the ground. The delay propagation algorithm must be allowed to continuously refreshes flight schedules. Such a method will be unique in the area and more research using such procedures could be very helpful to the flight industry in terms of practical uses. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

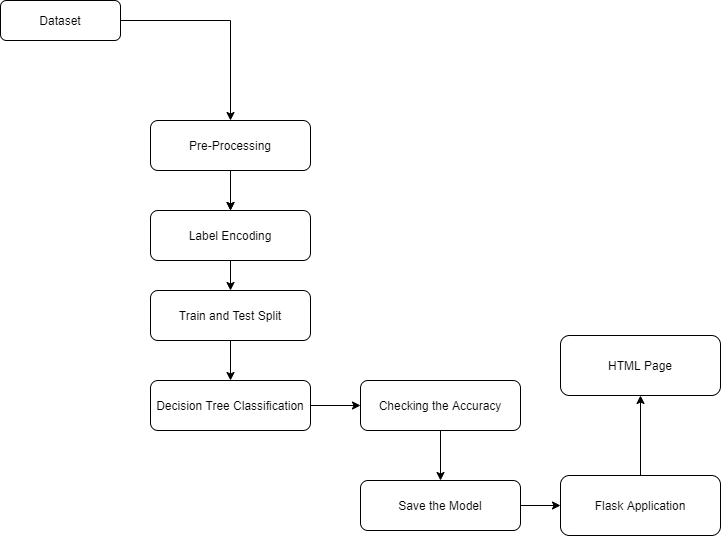
**2.2 Proposed solution**

Using a machine learning model, we can predict the flight arrival delays. We can take various inputs as a row of feature vector like departure date, departure delay, the distance between the two airports, scheduled arrival time, origin airport, destination airport etc to the algorithm. We then use decision tree classifier to predict if the flight arrival will be delayed or not. A flight is considered to be delayed when the difference between scheduled and actual arrival times is greater than 15 minutes. The model will be trained based on the features which will give the higher accuracy. Finally, it will be integrated into a web-based application. The final system allows the user to give the details of the flights as input and the system will display the output whether the flight will be late or not.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3. THEORITICAL ANALYSIS**

**3.1 Block diagram**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3.2 Hardware / Software designing**

The software used to implement this project is mainly Jupyter Notebook and Spyder. The other programming skills required are Python, Flask, HTML and CSS. The project works in 2 parts. The first part is building the model and the other part is building the frontend. The model building is done by using Machine Learning libraries by the help of Python. The frontend development is done by using HTML and the designs are added by using CSS. Both frontend and backend are connected by using Flask. The Flask generate localhost for the application to run.

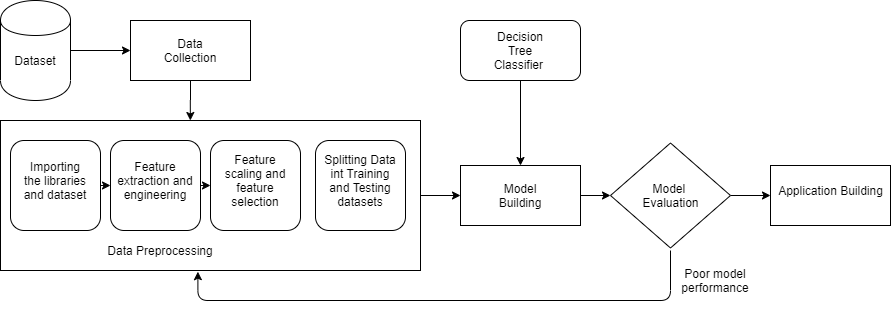
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**4. EXPERIMENTAL INVESTIGATIONS**

The flight's delay is caused by a variety of factors. The dataset which contains the record of every instance of the flight can be very well used to predict the delay of the flight. The data which save the records of every instance of the flight from departure to arrival such as when the flight leave the airport, when the wheels were taken off, when did it take off, when it reached the sky of destination, when did it land, when did passengers were finally allowed to come out etc. will be a better learning module for the Machine Learning Algorithm. This can provide wider options to be checked for the delay in flights rather than that based on the distance between the routes and types of routes. Nowadays, air traffic affects these important steps of flights travel and so including them in the prediction will give better accuracy. More than this, climate also affects the flight delays and so knowing the month and day will also be a major role to decide the delay in flights.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**5. FLOWCHART**

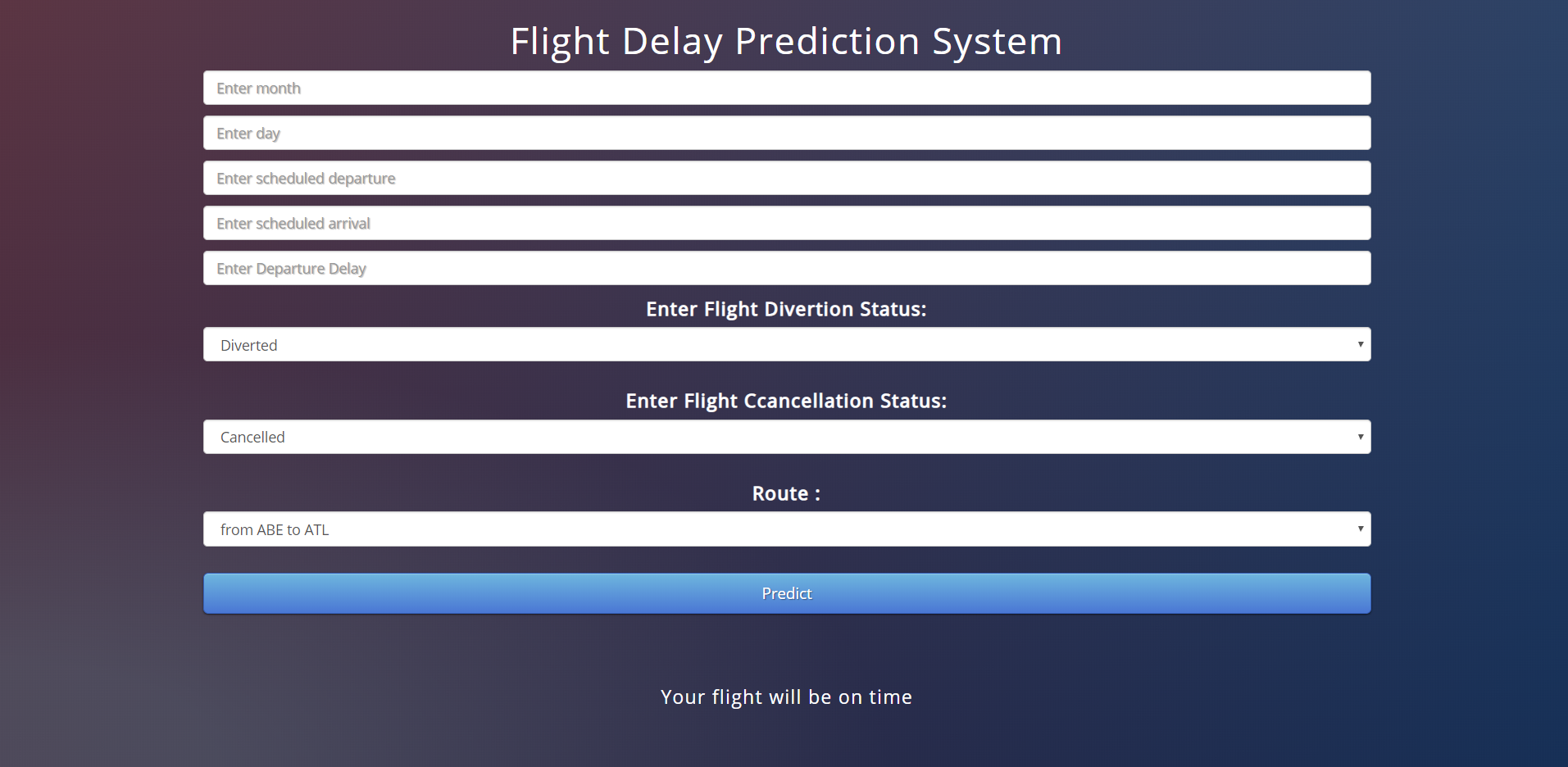


The first step in the building of this application is to get the dataset. We got the required dataset form the Kaggle platform. After data gathering, there is a process called Data preprocessing. In data preprocessing, we perform several methods like data cleaning, data integration, attribute selection, data transformation etc. All that to make our data clear and free from unwanted outliers and noise in data. Also there is a process called exploratory data analysis in which we understan the patterns and trends in our data to get useful insights for future building. We then split our data into training and testing. Generally, 80% of the data is used for training and 20% of the data is used for the testing purposes. Then comes the process of building the model also known as the the training phase. In this model is built using suitable Machine Learning algorithms as per the previous insights and problem statement. After the model is built, the model is tested using the testing data which we had kept aside. If the model is showing good accuracy then the model is accepted. Else if the accuracy is not satisfactory then the process iterates until the required accuracy is achieved or maximum iterations are done. After we achieved the required accuracy in our model, the next and final step is for the prediction of re. In our case we are deploying the application which will take user inputs as the criteria for new data for prediction. This data will be evaluated using our model and the prediction will be made. The predicted output will be displayed to the user.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**6. RESULT**

We have successfuly built the machine learning model and developed a web application which ultimately allows the user to give the details of the flights he wants to go or want to book as inputs and the application will display whether the flight will be delayed or not. Based on the output the user can plan his upcoming strategies for travel.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**7. ADVANTAGES & DISADVANTAGES**

**Advantages**:

The proposed method has proven to be highly capable of handling the challenges of large datasets and capturing the key factors influencing delays. This ultimately enables connected airports to collectively alleviate delay propagation within their network through collaborative efforts (e.g., delay prediction synchronization).

**Disadvantages**:

There can be the cases of accidental delays in flights, so a particular attention should be paid to extreme delays which could be the outliers which we tried to ignore. During the exploration of our dataset from Kaggle, it was seen that sometimes delays of several hours, like even tens of hours, could be recorded. This type of delay is less and the cause of these delays is probably linked to unpredictable events (weather, breakdown, accident. Taking into account a delay of this type will likely introduce a bias in the analysis and hence the output.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8. APPLICATIONS**

The application developed can be used by various type of users like passengers, airport staff, airlines and many more to facilitate the movement of flights and providing better time management strategies for their respective purpose.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**9. CONCLUSION**

This project and the analysis retrieved are useful not only for passengers point of view, but for every decision maker in the aviation industry. Apart from the financial losses incurred by the industry, flight delay also portray a negative reputation of the airlines, and decreases their reliability. It causes various sustainability issues, for example, increase in fuel consumption and gas emissions. The analysis carried here not only predicts delays based on the previous available data, but also give statistical description of airlines, their rankings based on their on-time performance, and delays with respect to time, showing the peak hours of delay.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**10. FUTURE SCOPE**

This application which we've developed can be modeled to give information about other possibilities as well. Like which airline has the least amount of delays and with the help of this airlines will try to compete and will think about measures to reduce delays to be better among the rivals.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**11. BIBILOGRAPHY**

1. <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
2. <https://smartbridge.teachable.com/courses/enrolled/869132>
3. <https://www.kaggle.com/fabiendaniel/predicting-flight-delays-tutorial>
4. <https://www.kaggle.com/usdot/flight-delays>
5. <https://www.edureka.co/blog/introduction-to-machine-learning/#What%20Is%20Machine%20Learning>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**APPENDIX**

1. Machine Learning: Machine learning is a subset of [Artificial Intelligence](https://www.edureka.co/blog/artificial-intelligence-tutorial/) (AI) which provides machines the ability to learn automatically & improve from experience without being explicitly programmed to do so.

2. Algorithm for machine learning: A Machine Learning algorithm is a set of rules and statistical techniques used to learn patterns from data and draw significant information from it. It is the logic behind a Machine Learning model. An example of a Machine Learning algorithm is the Linear Regression algorithm.

3: Training Data: The Machine Learning model is built using the training data. The training data helps the model to identify key trends and patterns essential to predict the output.

4. Testing Data: After the model is trained, it must be tested to evaluate how accurately it can predict an outcome. This is done by the testing data set.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**A. Source code**

**Jupyter Notebook**

# Libraries

**import** **numpy** **as** **np**  
**import** **pandas** **as** **pd**

In [2]:

# Dataset (1,00,000 rows)

df = pd.read\_csv('flightss.csv')

df.shape

Out[5]:

(99999, 31)

df['SCHEDULED\_ARRIVAL']

Out[7]:

0 430  
1 750  
2 806  
3 805  
4 320  
 ...   
99994 1157  
99995 1219  
99996 1842  
99997 1225  
99998 1454  
Name: SCHEDULED\_ARRIVAL, Length: 99999, dtype: int64

In [8]:

df.isnull().sum()

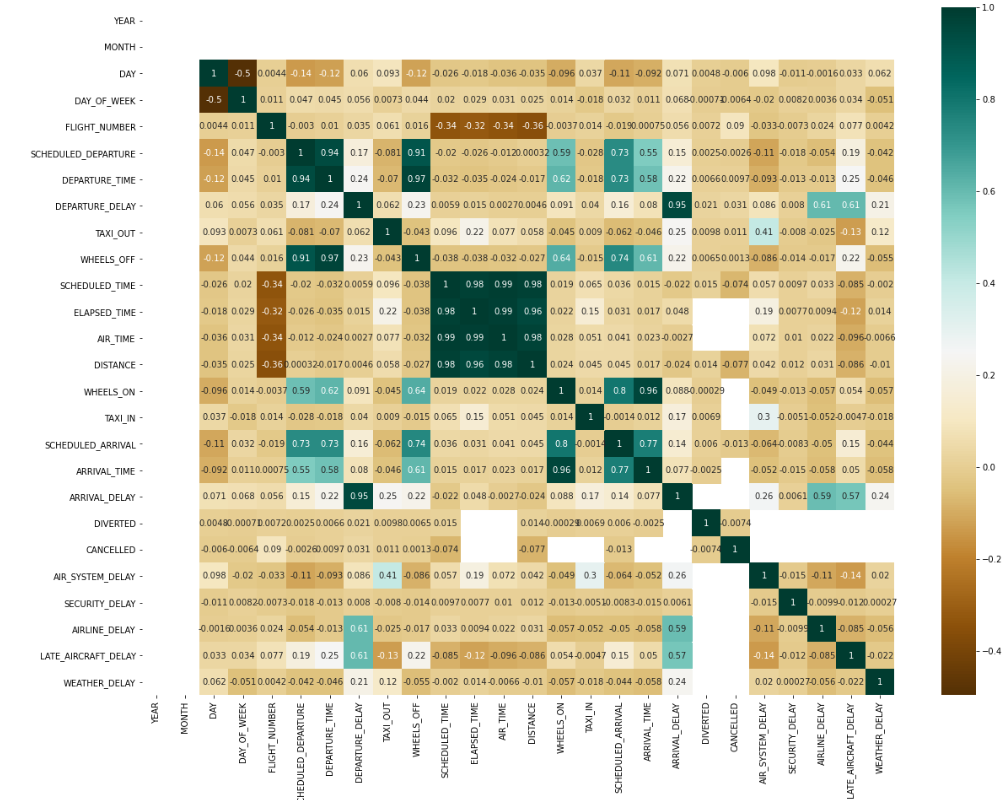
Out[8]:

YEAR 0  
MONTH 0  
DAY 0  
DAY\_OF\_WEEK 0  
AIRLINE 0  
FLIGHT\_NUMBER 0  
TAIL\_NUMBER 167  
ORIGIN\_AIRPORT 0  
DESTINATION\_AIRPORT 0  
SCHEDULED\_DEPARTURE 0  
DEPARTURE\_TIME 2298  
DEPARTURE\_DELAY 2298  
TAXI\_OUT 2371  
WHEELS\_OFF 2371  
SCHEDULED\_TIME 0  
ELAPSED\_TIME 2613  
AIR\_TIME 2613  
DISTANCE 0  
WHEELS\_ON 2440  
TAXI\_IN 2440  
SCHEDULED\_ARRIVAL 0  
ARRIVAL\_TIME 2440  
ARRIVAL\_DELAY 2613  
DIVERTED 0  
CANCELLED 0  
CANCELLATION\_REASON 97610  
AIR\_SYSTEM\_DELAY 65374  
SECURITY\_DELAY 65374  
AIRLINE\_DELAY 65374  
LATE\_AIRCRAFT\_DELAY 65374  
WEATHER\_DELAY 65374  
dtype: int64

In [9]

# Correlation between columns

**import** **matplotlib.pyplot** **as** **plt**  
**import** **seaborn** **as** **sns**  
  
plt.figure(figsize=(20,15))  
c= df.corr()  
sns.heatmap(c,cmap='BrBG',annot=**True**)  
  
*# We observe the correlation coeffients of interdependencies of all the columns here*



# Dropping Irrelevant columns

df =df.drop(['YEAR','FLIGHT\_NUMBER','AIRLINE','TAIL\_NUMBER','TAXI\_OUT','SCHEDULED\_TIME','DEPARTURE\_TIME','WHEELS\_OFF','ELAPSED\_TIME','AIR\_TIME','WHEELS\_ON','DAY\_OF\_WEEK','TAXI\_IN','ARRIVAL\_TIME','CANCELLATION\_REASON'],axis = 1)

# Fill null values

DEPARTURE\_DELAY ARRIVAL\_DELAY AIR\_SYSTEM\_DELAY SECURITY\_DELAY AIRLINE\_DELAY LATE\_AIRCRAFT\_DELAY WEATHER\_DELAY MEAN FOR ALL

In [20]:

df['DEPARTURE\_DELAY']=df['DEPARTURE\_DELAY'].fillna(df['DEPARTURE\_DELAY'].mean())  
df['AIR\_SYSTEM\_DELAY']=df['AIR\_SYSTEM\_DELAY'].fillna(df['AIR\_SYSTEM\_DELAY'].mean())  
df['SECURITY\_DELAY']=df['SECURITY\_DELAY'].fillna(df['SECURITY\_DELAY'].mean())  
df['AIRLINE\_DELAY'] = df['AIRLINE\_DELAY'].fillna(df['AIRLINE\_DELAY'].mean())  
df['LATE\_AIRCRAFT\_DELAY'] = df['LATE\_AIRCRAFT\_DELAY'].fillna(df['LATE\_AIRCRAFT\_DELAY'].mean())  
df['WEATHER\_DELAY'] = df['WEATHER\_DELAY'].fillna(df['WEATHER\_DELAY'].mean())  
df['ARRIVAL\_DELAY'] = df['ARRIVAL\_DELAY'].fillna(int(20)) *#Initialized a random value of 99 which is greater than 15 minutes*

In [21]:

df.isnull().sum()

Out[21]:

MONTH 0  
DAY 0  
ORIGIN\_AIRPORT 0  
DESTINATION\_AIRPORT 0  
SCHEDULED\_DEPARTURE 0  
DEPARTURE\_DELAY 0  
SCHEDULED\_ARRIVAL 0  
ARRIVAL\_DELAY 0  
DIVERTED 0  
CANCELLED 0  
AIR\_SYSTEM\_DELAY 0  
SECURITY\_DELAY 0  
AIRLINE\_DELAY 0  
LATE\_AIRCRAFT\_DELAY 0  
WEATHER\_DELAY 0  
dtype: int64

In [22]:

*#Converting all values of columns to integers*  
df['DEPARTURE\_DELAY']=df['DEPARTURE\_DELAY'].astype(int)  
df['AIR\_SYSTEM\_DELAY']=df['AIR\_SYSTEM\_DELAY'].astype(int)  
df['SECURITY\_DELAY']=df['SECURITY\_DELAY'].astype(int)  
df['AIRLINE\_DELAY']=df['AIRLINE\_DELAY'].astype(int)  
df['LATE\_AIRCRAFT\_DELAY']=df['LATE\_AIRCRAFT\_DELAY'].astype(int)  
df['WEATHER\_DELAY']=df['WEATHER\_DELAY'].astype(int)  
df['SCHEDULED\_ARRIVAL']=df['SCHEDULED\_ARRIVAL'].astype(int)  
df['SCHEDULED\_DEPARTURE']=df['SCHEDULED\_DEPARTURE'].astype(int)

In [23]:

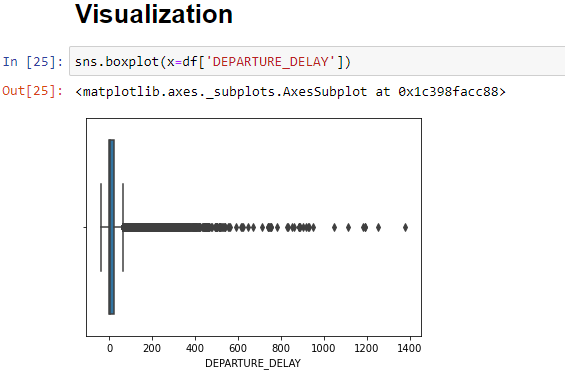
*# We create a new column called route which is a concatenated string of origin and destination airport which can later be label encoded*  
df['route']= df['ORIGIN\_AIRPORT'].astype(str) + df['DESTINATION\_AIRPORT'].astype(str)

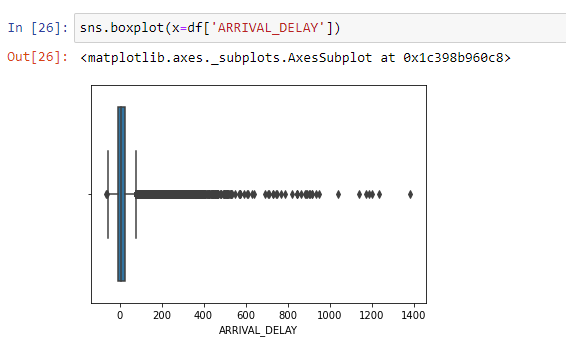
In [24]:

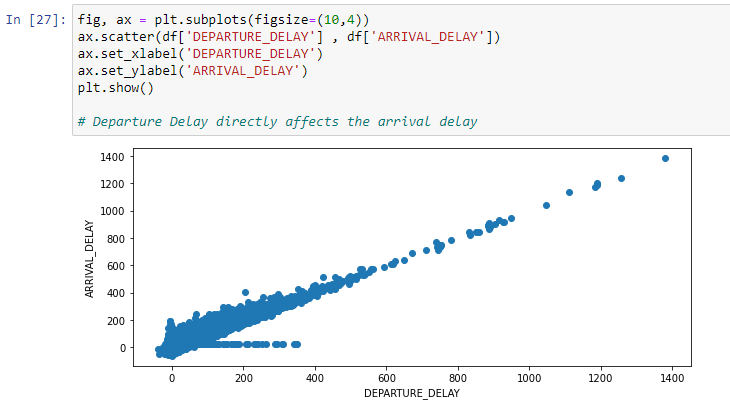
**from** **sklearn** **import** preprocessing  
lab = preprocessing.LabelEncoder()  
df['route']= lab.fit\_transform(df['route'])  
df['route'].unique()

Out[24]:

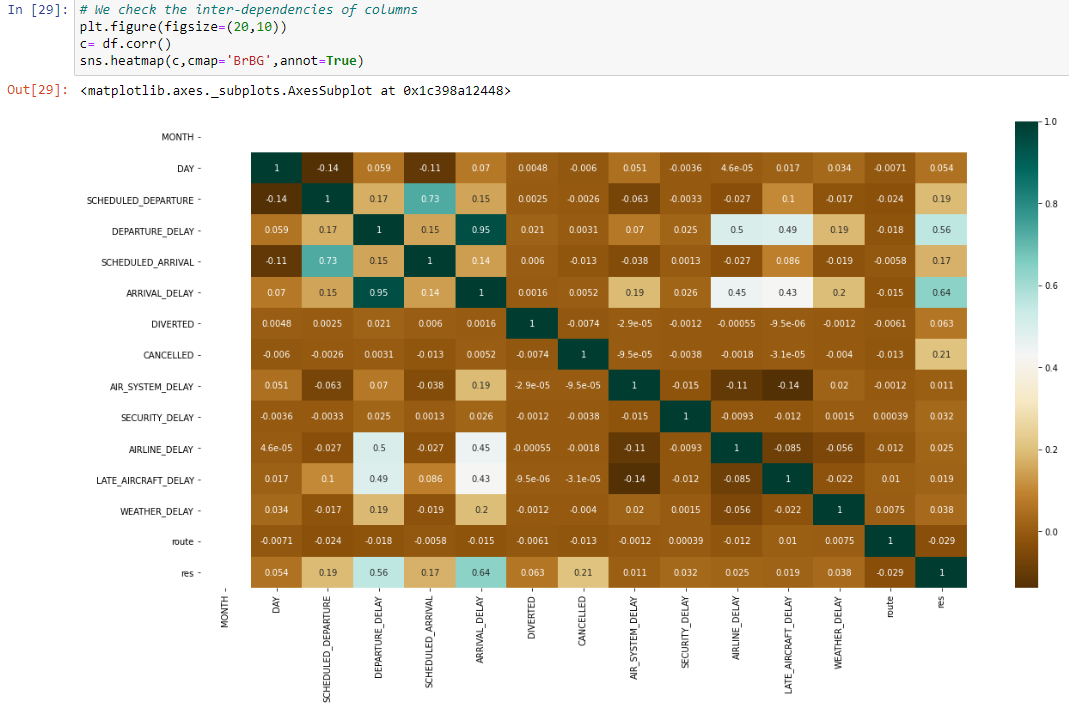
array([ 77, 2198, 3667, ..., 1773, 1520, 1271])



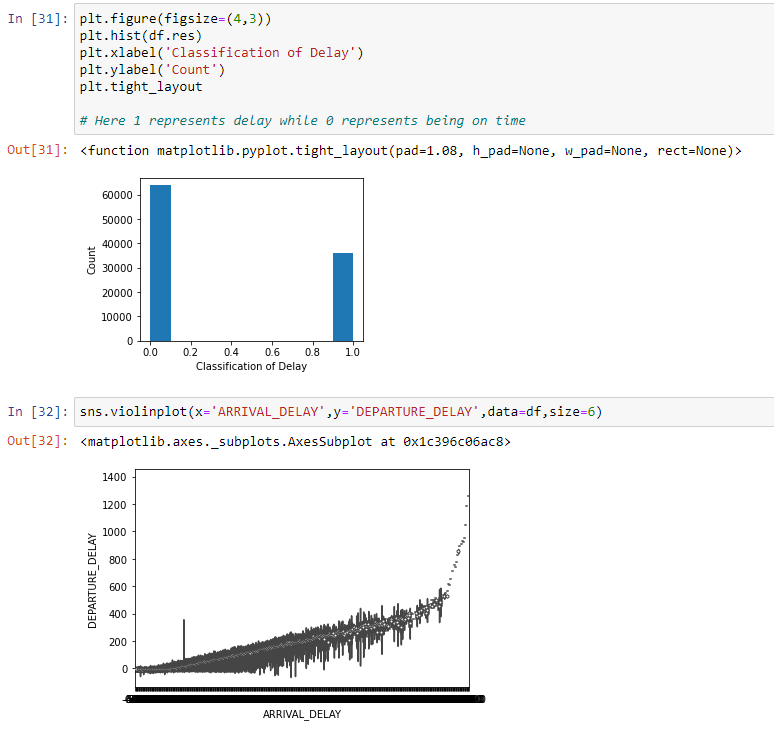




*#Created a final result column where 1 represents that the flight is late and 0 represents that it is on time*  
res=[]  
**for** i **in** range (len(df['ARRIVAL\_DELAY'])):  
 **if** (df['ARRIVAL\_DELAY'][i]>15): *#If a flight is late for more than 15 mins then it is put under the late arrived category*  
 res.append(1)  
 **else**:  
 res.append(0)  
df['res']=res







# Dependent and Independent Variables

In [34]:

x = df.iloc[:,[0,1,4,5,6,8,9,10,11,12,13,14,15]].values  
y = df.iloc[:,16].values

# Train Test Split

In [35]:

**from** **sklearn.model\_selection** **import** train\_test\_split  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size = 0.3, random\_state = 42)

In [36]:

**from** **sklearn.preprocessing** **import** StandardScaler  
sc = StandardScaler()  
x\_train = sc.fit\_transform(x\_train)  
x\_test = sc.transform(x\_test)

# Decision Tree Classifier

In [37]:

**from** **sklearn.tree** **import** DecisionTreeClassifier  
clf = DecisionTreeClassifier()  
clf = clf.fit(x\_train,y\_train)

In [38]:

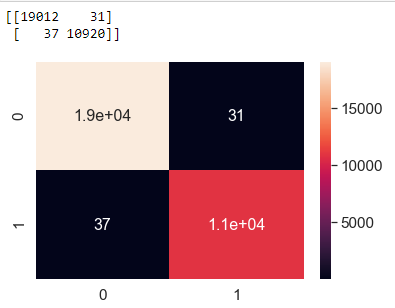
**from** **sklearn.metrics** **import** accuracy\_score  
y\_pred = clf.predict(x\_test)  
print("Accuracy:",accuracy\_score(y\_test, y\_pred)\*100,'%')

Accuracy: 99.77333333333334 %

# Confusion Matrix

In [39]:

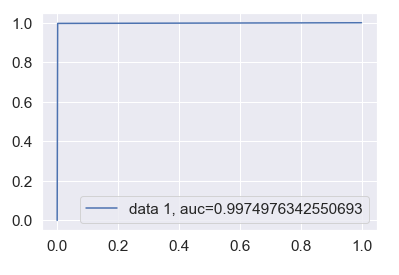
**from** **sklearn** **import** metrics  
cnf=metrics.confusion\_matrix(y\_test,y\_pred)  
print(cnf)  
sns.set(font\_scale=1.4) *# for label size*  
sns.heatmap(cnf, annot=**True**, annot\_kws={"size": 16}) *# font size*  
  
plt.show()



# ROC Curve

In [40]:

**from** **sklearn** **import** metrics   
**import** **matplotlib.pyplot** **as** **plt**  
y\_pred\_proba = clf.predict\_proba(x\_test)[::,1]  
fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_proba)  
auc = metrics.roc\_auc\_score(y\_test, y\_pred\_proba)  
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))  
plt.legend(loc=4)  
plt.show()



**import** **pickle**  
pickle.dump(clf,open('res.pkl','wb'))

# Logistic Regression

In [42]:

**from** **sklearn.linear\_model** **import** LogisticRegression  
logreg = LogisticRegression()  
logreg.fit(x\_train,y\_train)  
y\_pred1=logreg.predict(x\_test)

In [43]:

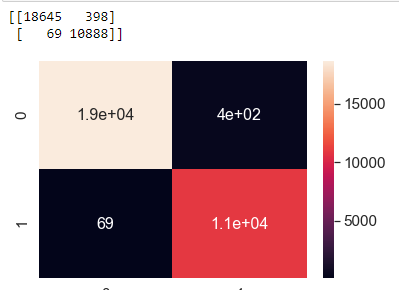
print("Accuracy:",accuracy\_score(y\_test, y\_pred1)\*100,'%')

Accuracy: 98.44333333333334 %

# Confusion Matrix

In [44]:

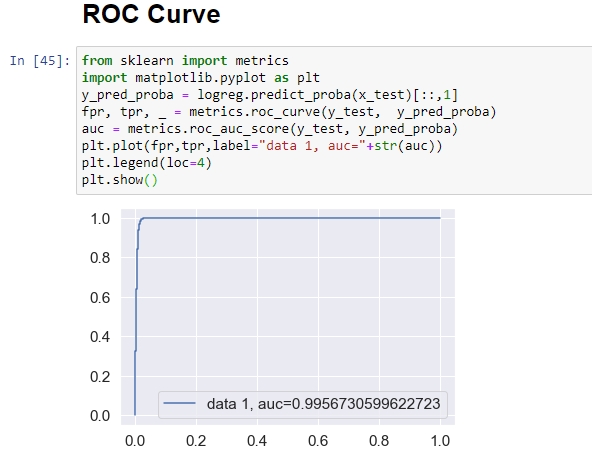
**from** **sklearn** **import** metrics  
cnf=metrics.confusion\_matrix(y\_test,y\_pred1)  
print(cnf)  
sns.set(font\_scale=1.4) *# for label size*  
sns.heatmap(cnf, annot=**True**, annot\_kws={"size": 16}) *# font size*  
plt.show()



# ROC Curve

In [45]:

**from** **sklearn** **import** metrics   
**import** **matplotlib.pyplot** **as** **plt**  
y\_pred\_proba = logreg.predict\_proba(x\_test)[::,1]  
fpr, tpr, \_ = metrics.roc\_curve(y\_test, y\_pred\_proba)  
auc = metrics.roc\_auc\_score(y\_test, y\_pred\_proba)  
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))  
plt.legend(loc=4)  
plt.show()



# Conclusion: The decision tree classifier has a better accuracy than Logistic Regression

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Flask Code:**

**app.py:**

from flask import Flask, render\_template, request

import pickle

import numpy as np

model = pickle.load(open('res.pkl', 'rb'))

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/login', methods=['POST'])

def login():

MONTH = request.form['month']

DAY = request.form['day']

SCHEDULED\_DEPARTURE = request.form['scheduled\_dep']

DEPARTURE\_DELAY = request.form['dep\_delay']

SCHEDULED\_ARRIVAL = request.form['scheduled\_arrival']

DIVERTED = request.form['diverted']

CANCELLED = request.form['cancelled']

AIR\_SYSTEM\_DELAY= 14.009280092800928 #mean

SECURITY\_DELAY= 0.024510245102451023 #mean

AIRLINE\_DELAY = 17.295672956729568 #mean

LATE\_AIRCRAFT\_DELAY = 26.005040050400503 #mean

WEATHER\_DELAY = 2.2603726037260374 #mean

# AIR\_SYSTEM\_DELAY= request.form['air\_sys\_delay']

# SECURITY\_DELAY= request.form['security\_delay']

# AIRLINE\_DELAY = request.form['airline\_delay']

# LATE\_AIRCRAFT\_DELAY = request.form['late\_aircraft\_delay']

# WEATHER\_DELAY = request.form['weather\_delay']

route= request.form['route']

total = [[int(MONTH), int(DAY),int(SCHEDULED\_DEPARTURE), int(DEPARTURE\_DELAY),int(SCHEDULED\_ARRIVAL), int(DIVERTED), int(CANCELLED),int(AIR\_SYSTEM\_DELAY), int(SECURITY\_DELAY), int(AIRLINE\_DELAY), int(LATE\_AIRCRAFT\_DELAY), int(WEATHER\_DELAY), int(route)]]

y\_pred = model.predict(total)

if y\_pred==1:

msg='Your flight will be delayed'

else:

msg='Your flight will be on time'

return render\_template("index.html", showcase = msg)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**HTML Code:**

**index.html:**

|  |  |
| --- | --- |
| <html> |  |
|  | <head> |
|  | <title>Flight Delay Prediction System</title> |
|  | <meta charset="utf-8"> |
|  | <meta name="viewport" content="width=device-width, initial-scale=1"> |
|  | <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"> |
|  | <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> |
|  | <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script> |
|  | <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'> |
|  | <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'> |
|  | <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'> |
|  | <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'> |
|  | <link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}"> |
|  | </head> |
|  | <body> |
|  | <div class="container"> |
|  | <h1>Flight Delay Prediction System</h1> |
|  | <form action = "{{ url\_for('login')}}" method="post"> |
|  | <input type="text" class="form-control" name="month" placeholder="Enter month" required="required"/> |
|  | <input type="text" class="form-control" name="day" placeholder="Enter day" required="required"/> |
|  | <input type="text" class="form-control" name="scheduled\_dep" placeholder="Enter scheduled departure" required="required"/> |
|  | <input type="text" class="form-control" name="scheduled\_arrival" placeholder="Enter scheduled arrival" required="required"/> |
|  |  |
|  |  |
|  | <input type="text" class="form-control"name="dep\_delay" placeholder="Enter Departure Delay" required="required"/> |
|  | <!-- <input type="text" class="form-control" name="distance" placeholder="Enter Distance" required="required"/> --> |
|  | <!-- <input type="text" class="form-control"name="diverted" placeholder="Enter diverted" required="required"/> --> |
|  |  |
|  | <label>Enter Flight Divertion Status:</label> |
|  | <select class="form-control" name="diverted"> |
|  | <option value="1">Diverted</option> |
|  | <option value="0">Not Diverted</option> |
|  | </select> |
|  | <br> |
|  | <!-- <input type="text" class="form-control" name="cancelled" placeholder="Enter cancellation status" required="required"/> --> |
|  | <label>Enter Flight Cancellation Status:</label> |
|  | <select class="form-control" name="cancelled"> |
|  | <option value="1">Cancelled</option> |
|  | <option value="0">Not Cancelled</option> |
|  | </select> |
|  | <br> |
|  | <label>Route :</label> |
|  | <select class="form-control" name="route"> |
|  | <option value="0">from ABE to ATL</option> |
|  | <option value="1">from ABE to DTW</option> |
|  | <option value="2">from ABE to ORD</option> |
|  | <option value="3">from ABI to DFW</option> |
|  | <option value="4">from ABQ to ATL</option> |
|  | <option value="5">from ABQ to BWI</option> |
|  | <option value="6">from ABQ to DAL</option> |
|  | <option value="7">from ABQ to DEN</option> |
|  | <option value="8">from ABQ to DFW</option> |
|  | <option value="9">from ABQ to HOU</option> |
|  | <option value="10">from ABQ to IAH</option> |
|  | <option value="11">from ABQ to JFK</option> |
|  | <option value="12">from ABQ to LAS</option> |
|  | <option value="13">from ABQ to LAX</option> |
|  | <option value="14">from ABQ to MCI</option> |
|  | <option value="15">from ABQ to MDW</option> |
|  | <option value="16">from ABQ to MSP</option> |
|  | <option value="17">from ABQ to OAK</option> |
|  | <option value="18">from ABQ to ORD</option> |
|  | <option value="19">from ABQ to PDX</option> |
|  | <option value="20">from ABQ to PHX</option> |
|  | <option value="21">from ABQ to SAN</option> |
|  | <option value="22">from ABQ to SEA</option> |
|  | <option value="23">from ABQ to SFO</option> |
|  | <option value="24">from ABQ to SLC</option> |
|  | <option value="25">from ABR to MSP</option> |
|  | <option value="26">from ABY to ATL</option> |
|  | <option value="27">from ACT to DFW</option> |
|  | <option value="28">from ACV to SFO</option> |
|  | <option value="29">from ACY to FLL</option> |
|  | <option value="30">from ACY to MCO</option> |
|  | <option value="31">from ACY to MYR</option> |
|  | <option value="32">from ACY to PBI</option> |
|  | <option value="33">from ACY to RSW</option> |
|  | <option value="34">from ACY to TPA</option> |
|  | <option value="35">from ADK to ANC</option> |
|  | <option value="36">from ADQ to ANC</option> |
|  | <option value="37">from AEX to ATL</option> |
|  | <option value="38">from AEX to DFW</option> |
|  | <option value="39">from AEX to IAH</option> |
|  | <option value="40">from AGS to ATL</option> |
|  | <option value="41">from ALB to ATL</option> |
|  | <option value="42">from ALB to BWI</option> |
|  | <option value="43">from ALB to CLT</option> |
|  | <option value="44">from ALB to FLL</option> |
|  | <option value="45">from ALB to IAD</option> |
|  | <option value="46">from ALB to LAS</option> |
|  | <option value="47">from ALB to MCO</option> |
|  | <option value="48">from ALB to MDW</option> |
|  | <option value="49">from ALB to ORD</option> |
|  | <option value="50">from ALB to TPA</option> |
|  | </select> |
|  | <br> |
|  | <!-- <input type="text" class="form-control" name="air\_sys\_delay" placeholder="Enter Air System Delay" required="required"/> |
|  | <input type="text" class="form-control" name="security\_delay" placeholder="Enter security delay" required="required"/> |
|  | <input type="text" class="form-control" name="airline\_delay" placeholder="Enter Airline Delay" required="required"/> |
|  | <input type="text" class="form-control" name="late\_aircraft\_delay" placeholder="Enter Late Aircraft Delay" required="required"/> |
|  | <input type="text" class="form-control" name="weather\_delay" placeholder="Enter Weather Delay" required="required"/> --> |
|  |  |
|  | <button type="submit" class=" btn btn-primary btn-block btn-large">Predict</button> |
|  |  |
|  | </form> |
|  | <br> |
|  | <br> |
|  | {{showcase}} |
|  |  |
|  | </div> |
|  | </body> |
|  | </html> |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**