**FLIGHT DELAY PREDICTION FOR AVIATION INDUSTRY USING MACHINE LEARNING**

PROJECT REPORT TEMPLATE

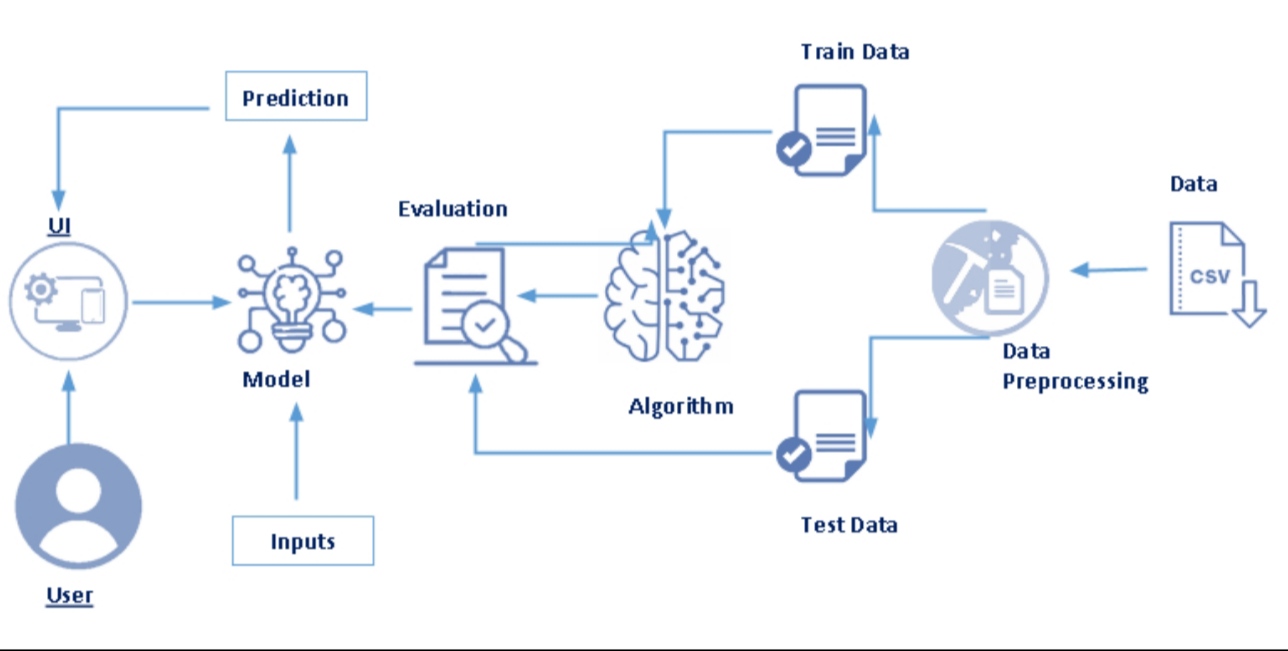
1. **Introduction**

Over the last twenty years, air travel has been increasingly preferred among travelers, mainly because of its speed and in some cases comfort. This has led to phenomenal growth in air traffic and on the ground. An increase in air traffic growth has also resulted in massive levels of aircraft delays on the ground and in the air. These delays are responsible for large economic and environmental losses. According to, taxi-out operations are responsible for 4,000 tons of hydrocarbons, 8,000 tons of nitrogen oxides and 45,000 tons of carbon monoxide emotions in the United States in 2007. Moreover, the economic impact of flight delays for domestic flights in the US is estimated to be more than $19 Billion per year to the airlines and over $41 Billion per year to the national economy in response to growing concerns of fuel emissions and their negative economy in response to growing concerns of fuel emissions and their negative impact on health, there is active research in the aviation industry for finding techniques to predict flight delays accurately in order to optimize flight operations and minimize delays.

Using a machine learning model, we can predict flight arrival delays. The input to our algorithm is rows of feature vector like departure dte, departure delay, distance between the two airports, scheduled arrival time etc. We then use decision tree classifier to predict if the flight arrival will be delayed or not. A flight is delayed when difference between scheduled and actual arrival times is greater than 15 minutes. Furthermore, we compare decision tree classifier with logistic regression and a simple neural network for various figures of merit.

* 1. **Overview:**

**Technical architecture**



**Project flow**

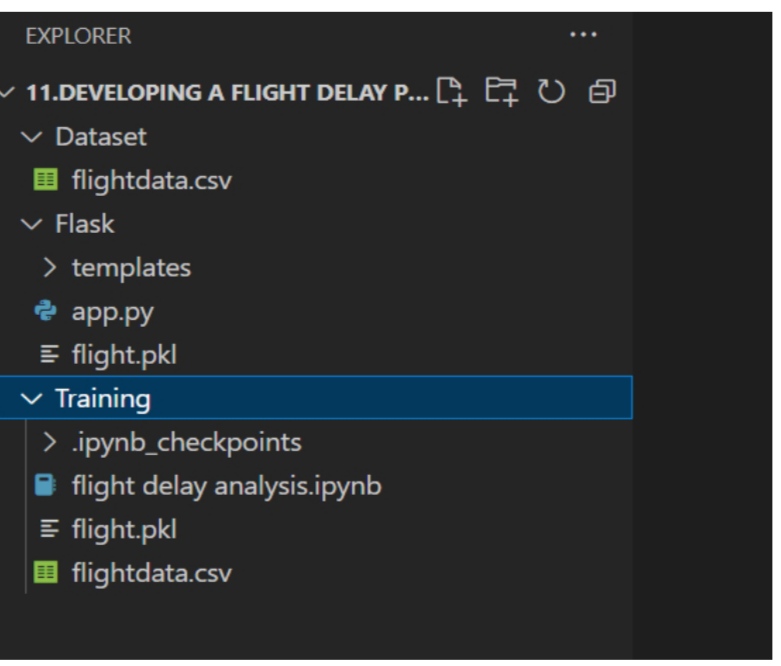
* User interacts with the UI to enter the input.
* Entered input is analysed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI.

To accomplish this, we have to complete all the activities listed below,

* Define problem/problem understanding
* Specify the business problem
* Business requirements
* Literature survey
* Social or business impact.
* Data collection & preparation
* Collect the dataset
* Data preparation
* Exploratory data analysis
* Descriptive statistical
* Visual analysis
* Model building
* Training the model in multiple algorithms
* Testing the model
* Performance testing & hyperparameter tuning
* Testing model with multiple evaluation metrics
* Comparing model accuracy before & after applying hyperparameter tuning
* Model deployment
* Save the best model
* Integrate with web framework
* Project demonstration & documentation
* Record explanation video for project end to end solution
* Project documentation-step by step project development procedure

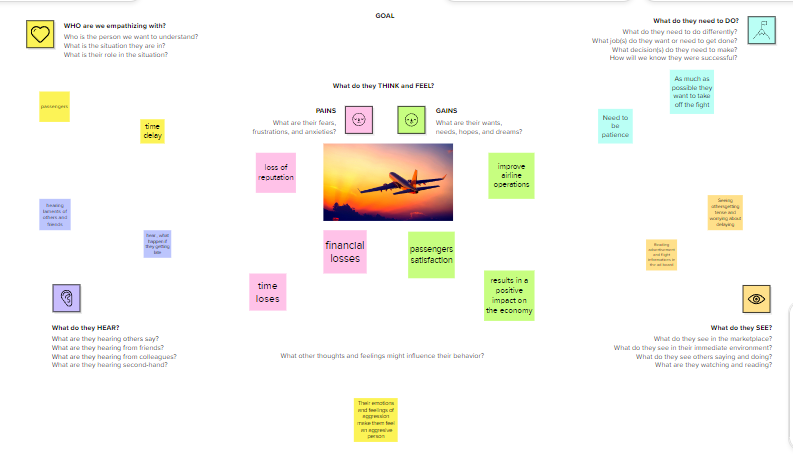
**Project structure**

Create the project folder which contains files as shown below

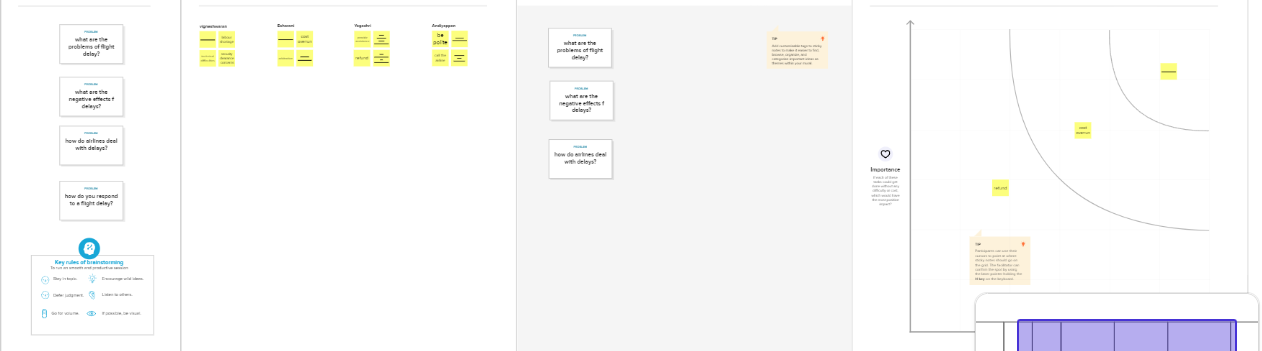


* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* flight.pkl is our saved model. Further we will use this model for flask integration.
* Training folder contains a model training file.
  1. **Purpose**
* Performing a flight delay project in college can serve various purposes, such as providing students with an opportunity to develop skills in data analysis, statistical modeling and programming.
* The project can also help students understand the causes of flight delays, their impact on the airline industry and passengers, and the ways to mitigate them.
* Additionally, the project can contribute to the body of research on air transportation and help improve the efficiency and safety of the air travel system.
* To achieve these goals, the project can involve collecting and analyzing data on flight delays from various sources, such as airline databases, weather reports, and airport operations.
* The data can be used to build predictive models that can estimate the probability and duration of flight delays based on various factors, such as the time of day, the airline, the airport, and the weather conditions.
* The models can be evaluated and refined using statistical techniques, such as regression analysis, hypothesis testing, and model validation.
* The project can also involve developing software tools that can help airlines, airports, and passengers monitor and manage flight delays.
* For example, the tools can provide real-time information on flight status, alternative routes, and travel advisories, and enable passengers to rebook their flights or claim compensation for delays or cancellations.
* Some of the achievements of performing a flight delay project in college can include enhancing students analytical and problem-solving skills, fostering collaboration and teamwork, and preparing students for careers in the aviation industry or related fields.
* The project can also contribute to the academic literature on air transportation and provide insights into the factors that affect flight delays and the ways to mitigate them.
* Ultimately, the project can help improve the efficiency and safety of the air travel system and enhance the overall passenger experience.

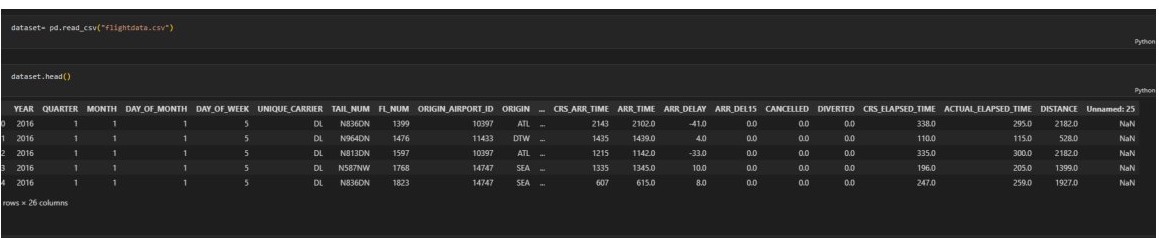
1. **Problem Definition & Design Thinking**
   1. **Empathy map**

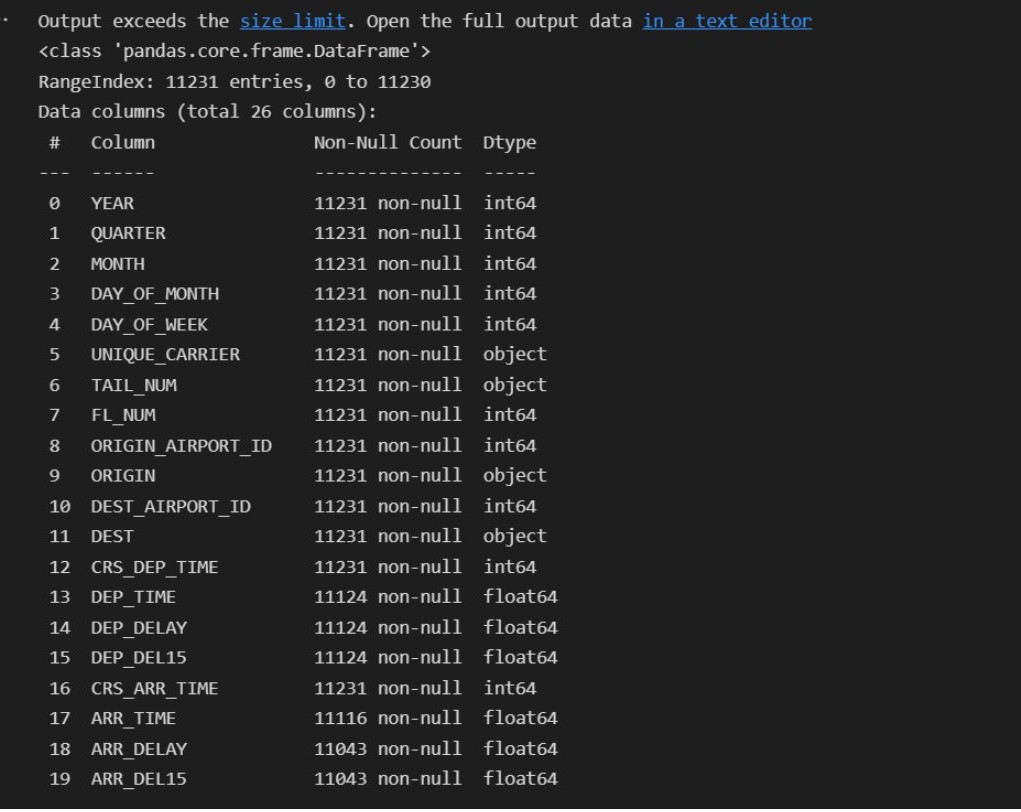


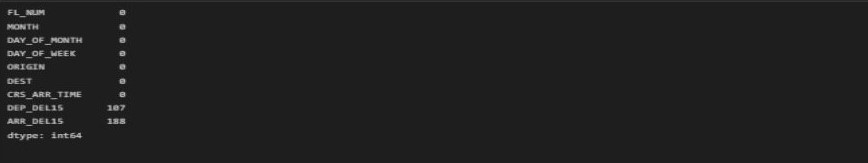
* 1. **Ideation & brainstorming map**

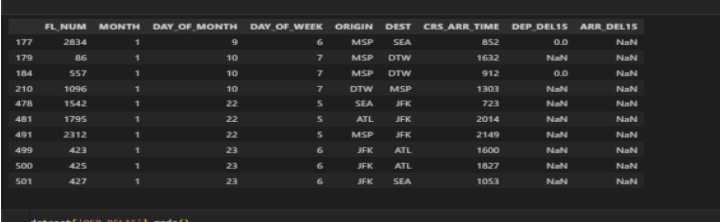
****

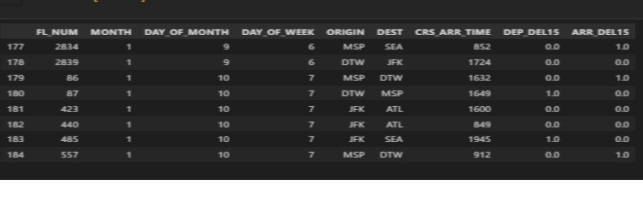
1. **Result**
   1. **colab result**

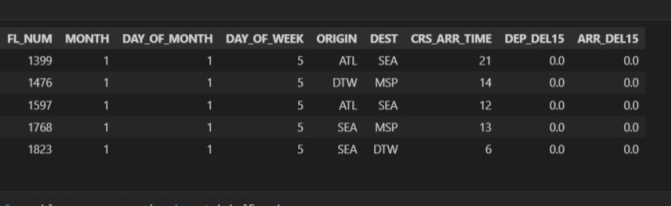
****

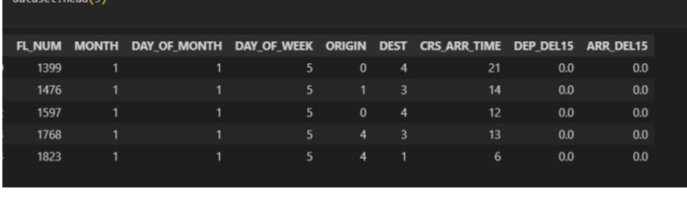
****

****

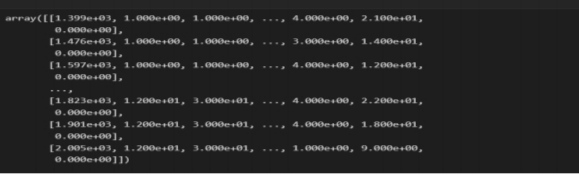


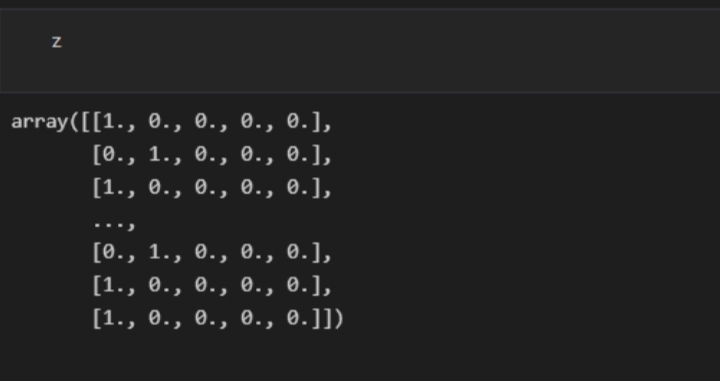


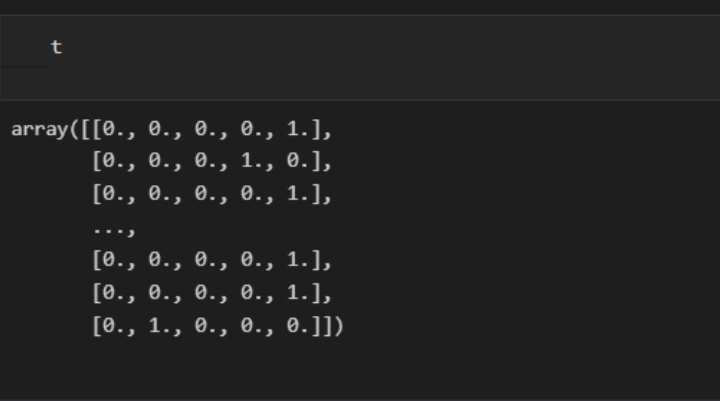


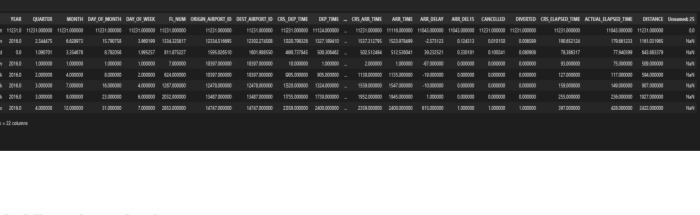


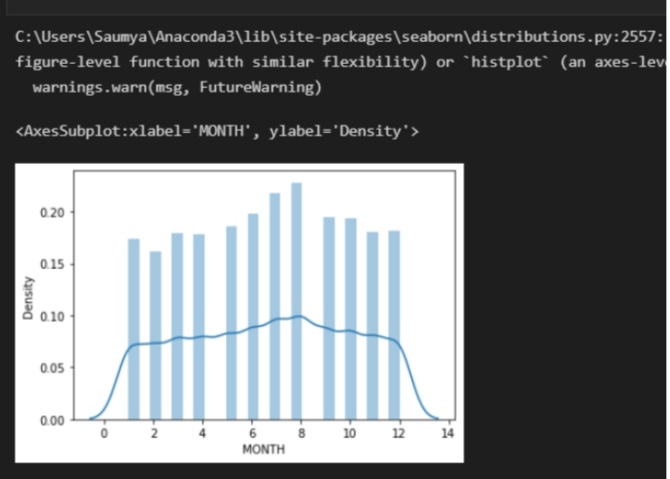


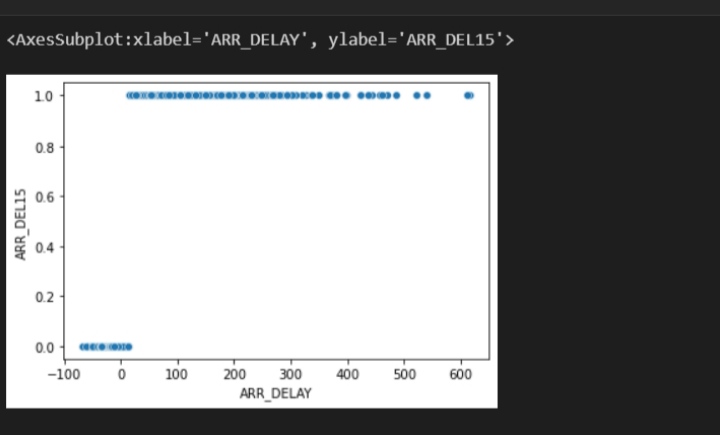


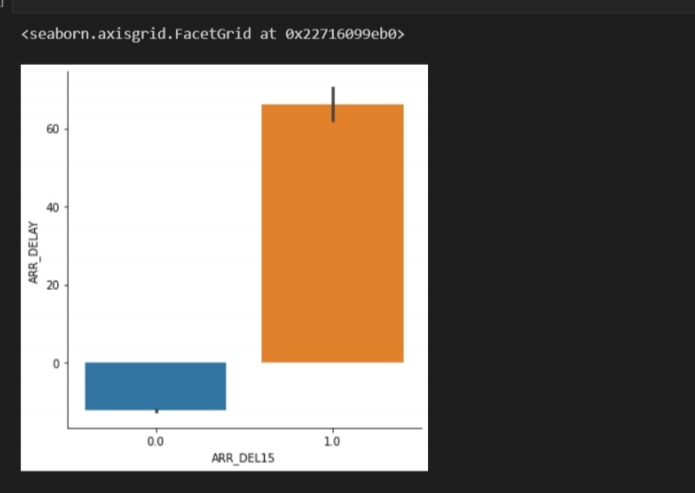


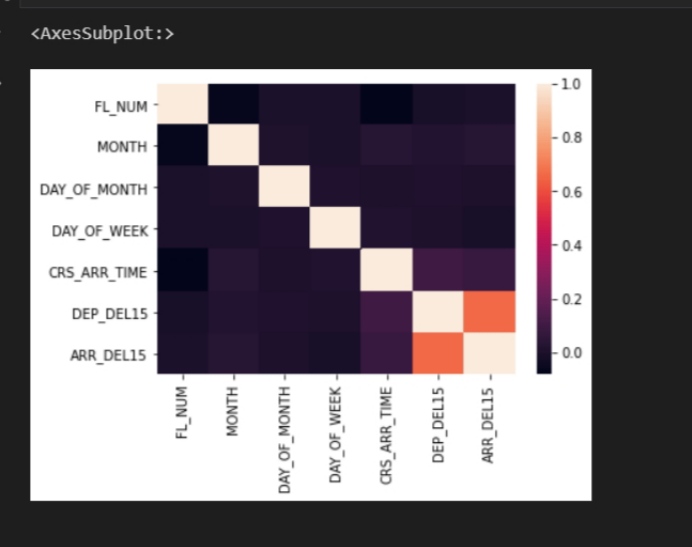


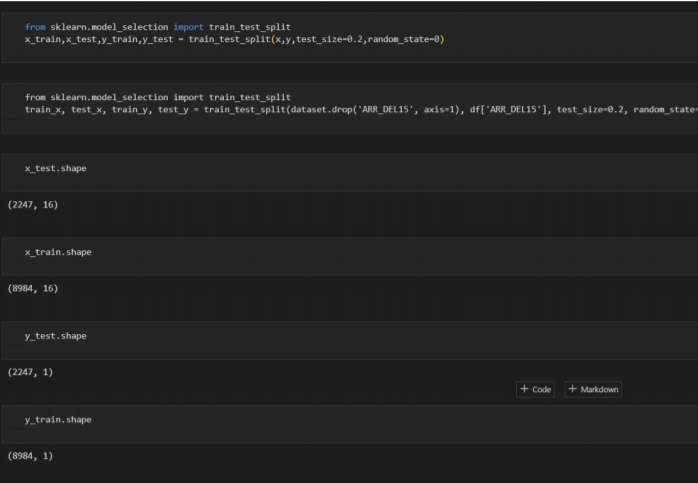


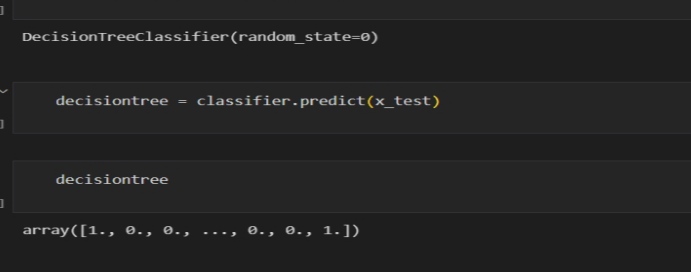


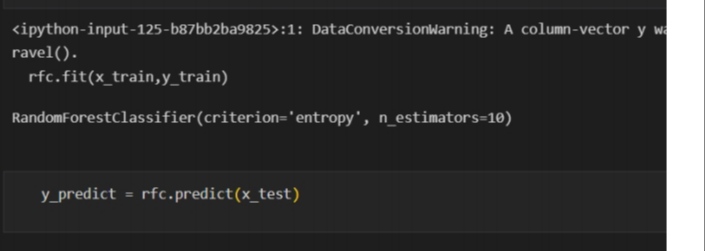


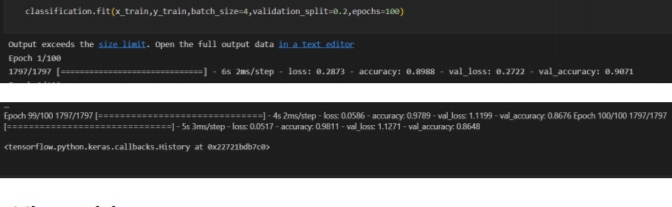


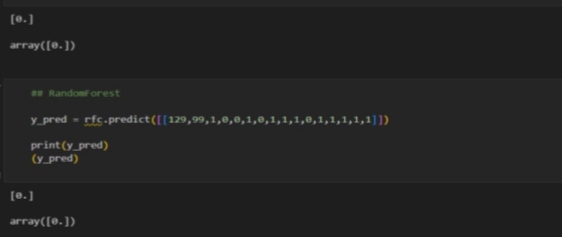


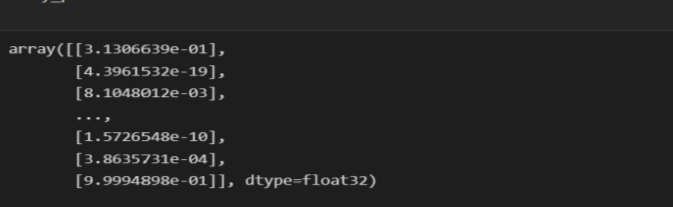


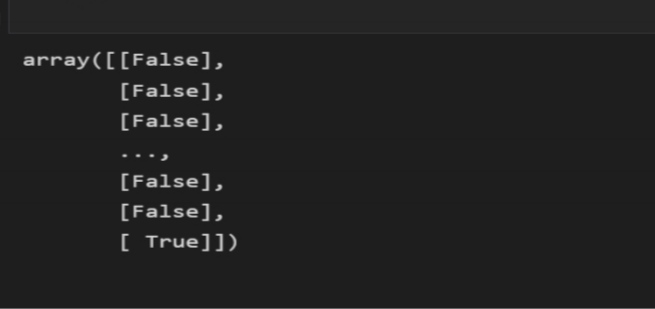


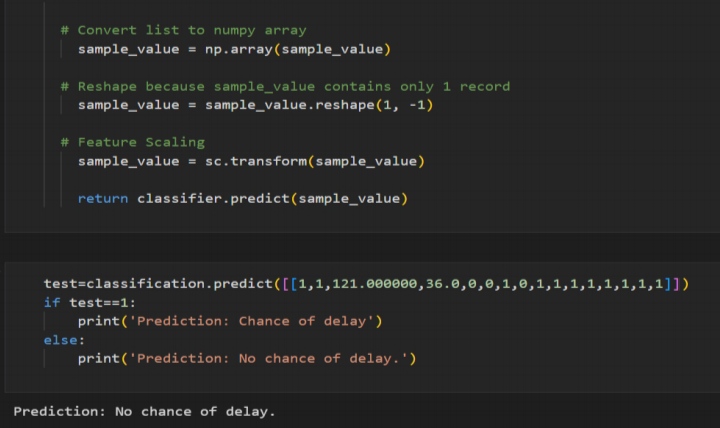


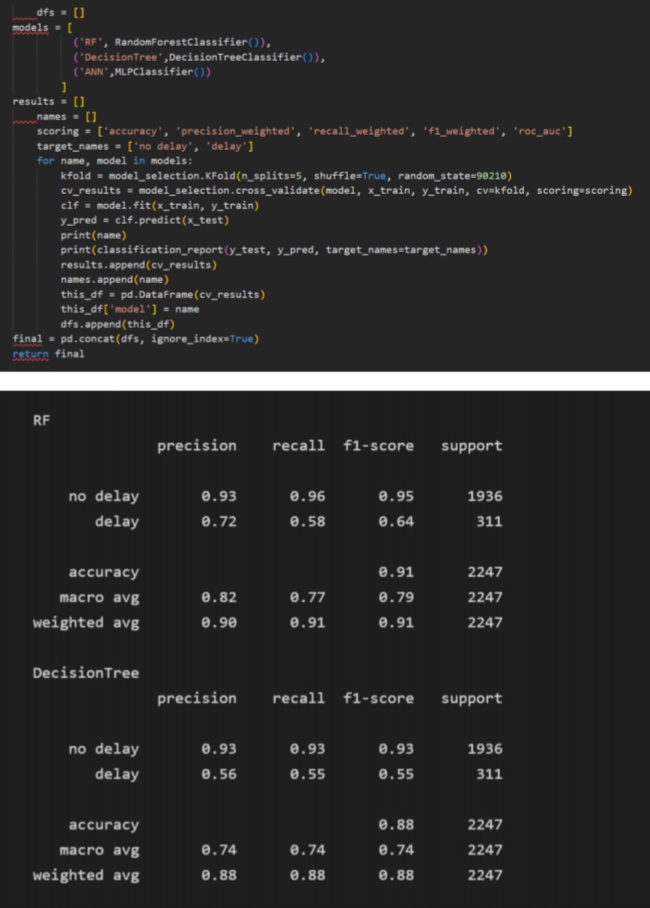


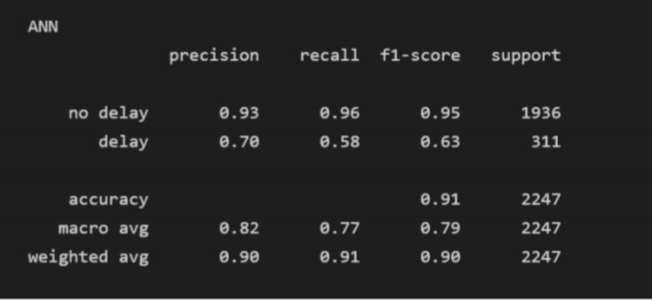


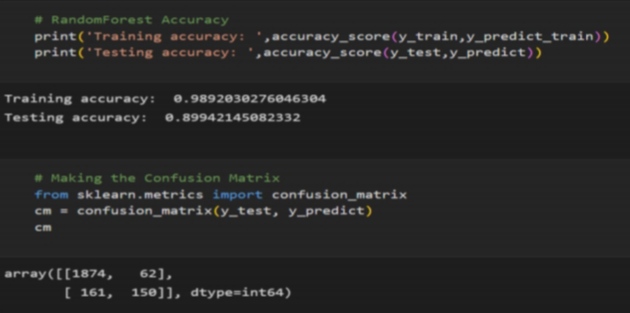


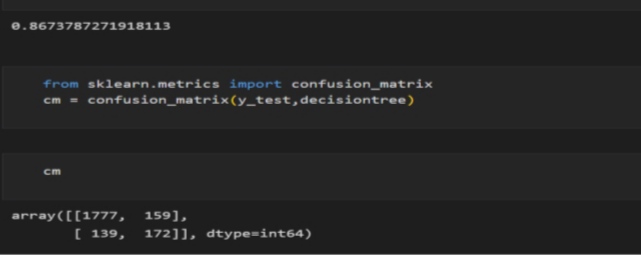


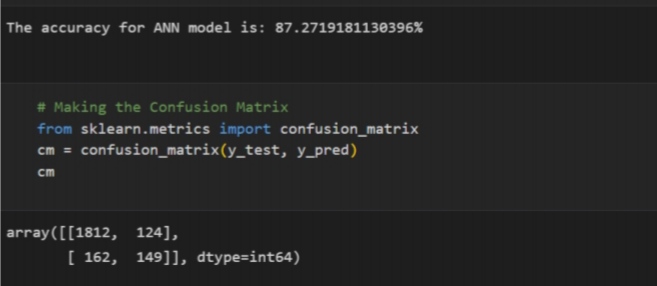


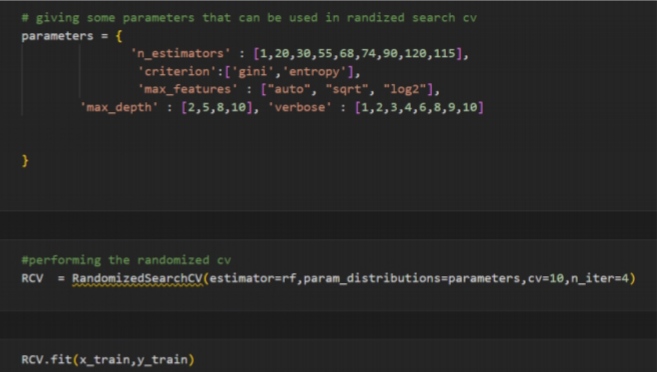


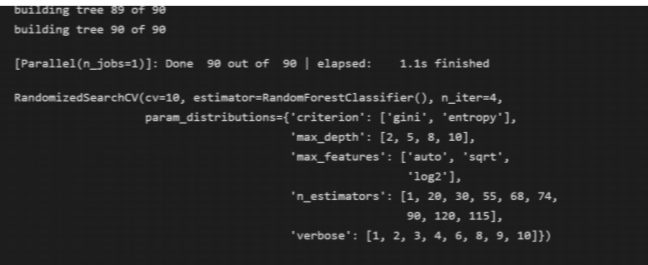


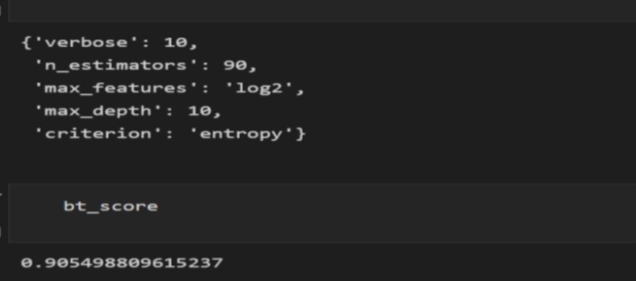


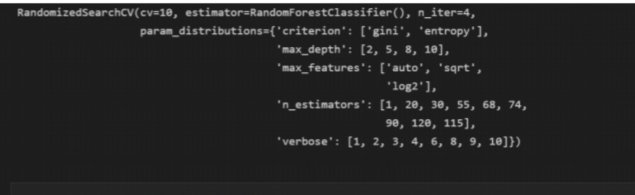


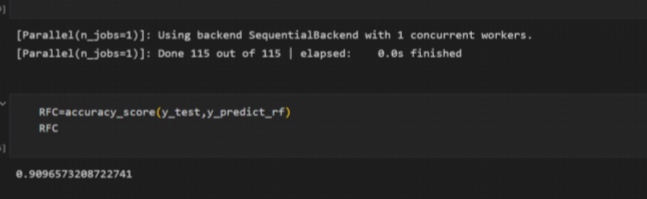




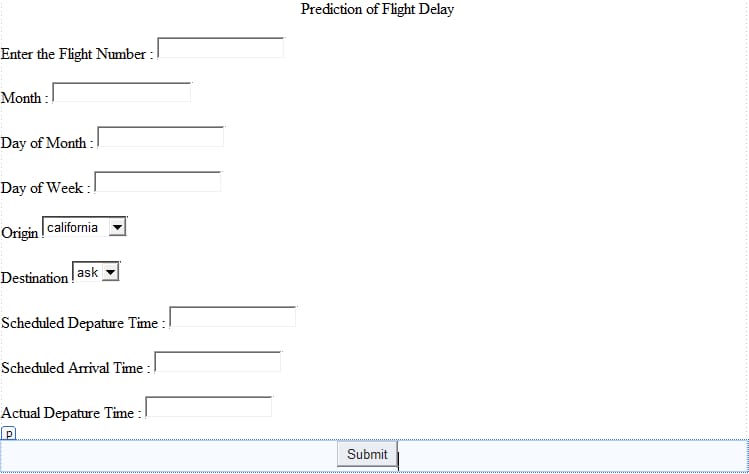








* 1. **website result**

The flight will be on time

1. **Advantages**

* Improved passenger experience: fewer flight delays mean fewer frustrated passengers, leading to a better overall travel experience.
* Cost savings for airlines: flight delays can be expensive for airlines, as they may need to compensate passengers, reschedule flights, and incur other costs. By reducing flight delays, airlines can save money and operate more efficiently.
* Increased safety: delays can put pressure on pilots and crew to make up time, potentially compromising safety.
* By reducing delays, there is less pressure on pilots and crew to rush, reducing the risk of accidents.
* Better use of airport capacity: delays can cause congestion at airports, reducing the overall capacity of the airport.
* By reducing delays, more flights can be accommodated, improving overall airport capacity.

**Disadvantages**

* Increased costs for airlines: some proposed solutions to reduce fight delays, such as upgrading air traffic control systems or investing in more infrastructure, can be expensive for airlines.
* Resistance to change: some proposed solutions may require significant changes to existing systems, which can be difficult to implement and meet with resistance from stakeholders.
* Potential for unintended consequences: any change to the air travel system can have unintended consequences, such as increasing air traffic in certain areas or reducing safety in others.
* Technological limitations: some proposed solutions, such as using new technologies to improve air traffic control, may be limited by technological capabilities or regulatory restrictions.

1. **Applications**

* The Flight Delay Project has a wide range of applications in the aviation industry. Some of the potential applications of this project include:
* Airline operations: Airlines can use the predictive models generated by the Flight Delay Project to optimize their operations, reduce delays and cancellations, and improve on-time performance.
* Air traffic control: Air traffic controllers can use the information provided by the Flight Delay Project to make real-time decisions about routing, landing, and takeoff times.
* Passenger experience: The Flight Delay Project can provide passengers with real-time updates on their flight status, including delays, cancellations, and expected arrival times.
* Maintenance planning: Airlines can use the predictive models generated by the Flight Delay Project to plan maintenance schedules, reduce downtime, and increase the reliability of their fleet.
* Crew scheduling: The Flight Delay Project can help airlines optimize crew schedules, ensuring that the right crew members are available at the right times.
* Airport planning: Airport operators can use the information provided by the Flight Delay Project to plan infrastructure investments, including runway expansions and terminal upgrades.
* Capacity planning: The Flight Delay Project can help airlines and airports optimize their capacity planning, ensuring that they have the right resources in place to handle changes in demand.
* Weather forecasting: The Flight Delay Project can use weather data to predict the impact of weather on flight schedules and help airlines and airports prepare for adverse weather conditions.
* Customer service: The Flight Delay Project can help airlines improve their customer service by providing passengers with timely and accurate information about their flight status.
* Business intelligence: The Flight Delay Project can provide airlines and airports with valuable insights into their operations, helping them make data-driven decisions about resource allocation and investment.

1. **Conclusion**

This paper presented the need to develop a system to predict the delay in flights along with its methodology. The papergives details about the range of different methodology that is used or can be used to find out the delay in flights. As flight delay cost a lot to the airlines as well as passangers in financial and environmental terms, flight delay is a the talk of the hour.Flight delay causes surging of prices by costing a lot on operational purpose They may increase prices to customers and operational prices to airlines. As the outcome is directly associated with the passanger and the airlines which inturn is liked to another set of airline and pasaangers it is very crucial to get real time delay for each player within the air transport system. hence there is a requirement to develop a system to predict the delay in flights to scale back monetory loss and for the higher and smooth operation. Classification or reggrerssion ways are often accustomed determine the delay which includes Feed forward network, Neural Network, Random Forrest, decision tress, Naïve Bayes Classification Tree,Regression Tree, etc. As seen from the articles and papers

these methodologies offer virtually identical accuracy however we want an algorithmic rule that is good with real world prediction and analysis and thus: naïve-

Bayes. except being smart with real time prediction algorithmic rule that considers or assumes independence among predictors that makes the system scalable as other independent attribute may be superimposed up to the algorithmic rule for computation

of the delay. the expected delay can thus facilitate the ground employees for creating correct and smooth operation plans and therefore the data if sent to the passengers will profit the airlines also because the passengers.

1. **Future scope**

The future scope of flight delay project is promising, as it has the potential to revolutionize the aviation industry by significantly reducing flight delays and improving passenger experience. With advancements in technology, machine learning algorithms can be trained on vast amounts of historical flight data to accurately predict flight delays and enable airlines to take proactive measures to prevent them. Furthermore, real-time data analysis can enable airlines to make informed decisions about flight scheduling, maintenance, and crew allocation. This can result in significant cost savings and improved customer satisfaction. The project can also have wider applications in the transportation industry, including railways and shipping.

1. **Appendix**
2. **Source code**

Importing the libraries

import pandas as pd

import numpy as np

import pickle

import matplotlib.pypot as plt

%matplotlib inline

import seaborn as sns

import sklearn

From sklearn.tree import DecisionTreeClassifier

From sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier

From sklearn.neihbors import KNeighborsClassifier

From sklearn.model\_selection import RadomizedSearchCV

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

Read the dataset

dataset= pd.read\_csv(“flightdata.csv”)

dataset.head()

Handling missing values

dataset.info()

dataset=dataset.drop(‘Unnamed: 25’, axis=1)

dataset.isnull().sum()

#filter the dataset to climinate colimns that aren’t relevant to a predictive model.

dataset=dataset[[“FL\_NUM”,”MONTH’,”DAY\_OF\_MONTH”,”DAY\_OF\_WEEK”,”ORIGIN”,”DEST”,”CRS\_ARR\_TIME”,”DEP\_DEL15”,”ARR\_DEL15”]]

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

dataset[‘DEP\_DEL15”].Mode()

#replace the missing values with is.

dataset=dataset.fillna({‘ARR\_DEL15’:1)}

dataset=dataset.fillna({‘DEP\_DEL15’:0)}

dataset.iloc[177:185]

Handling categorical values

import math

for index, row in dataset.iterrows():

dataset.loc[index, ‘CRS\_ARR\_TIME’]=math.floor(row[‘CRS\_ARR\_TIME’]/100)

dataset.head()

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

datset[‘DEST’]=le.fit\_transform(dataset[‘DEST’])

dataset[‘ORIGIN’]=le.fit\_transform(dataset[‘ORIGIN’])

dataset.head(5)

dataset.head()

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

oh=OneHotEncoder()

z=oh.fit\_transform(x[:,4:5]).toarray()

t=oh.fit\_trasform(x[:,5:6]).toarray()

#x=np.delete(x,[4,7],axis=1)

Descriptive statistical

Flight\_data.describe()

Univariate analysis

sns.distplot(flight\_data.MONTH)

Bivariate analysis

sns.scatterplot(x=’ARR\_DELAY’, v=’ARR\_DEL15’, data=flight\_data)

sns.catplot(x=”ARR\_DEL15”,Y=”ARR\_DELAY”, kind=’bar’, data=(flight\_data)

Multivariate analysis

Sns.heatmap(dataset.corr())

Splitting data into dependent and independent variables

dataset=pd.get\_dummies(dataset, columns=[‘ORIGIN’, ‘DEST’])

dataset.head()

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

y=dataset.ilot[:, 8:9].values

splitting data into train and test

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)

from sklearn.model\_selection import train\_test\_split

train\_x, test\_x, train\_y, test\_y=train\_test\_split(datset.drop(‘ARR\_DEL15’, axis=1), df[‘ARR\_DEL15’], test\_size=0.2, random\_state=0)

x\_test.shape

x\_train.shape

y\_test.shape

y\_train.shape

Scaling the data

From sklearn.preprocessing import StandardScaler

Sc=StandardScaler()

x\_train=sc.fit\_transform(x\_train)

x\_test=sc.transform(x\_test)

Decision tree model

From sklearn.tree import DecisionTreeClassifier

classifier=DecisionTreeClassifier(random\_state=0)

classifier.fit(x\_train,y\_train)

decisiontree=classifier.predict(x\_test)

decisiontree

fromsklearn.metrics import accuracy\_score

desacc=accuracy\_score(y\_test, decisiontree)

Random forest model

from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier(n\_estimators=10, criterion=’entropy’)

rfc.fit(x\_train, y\_train)

y\_predict=rfc.predict(x\_test)

ANN model

# Importing the Keras libraries and packages

Import tensorflow

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

# creating ANN skleton view

classification=sequential()

classification.add(Dense(30,activation=’relu’))

classification.add(Dense(128,activation=’relu’))

classification.add(Dense(64,activation=’relu’))

classification.add(Dense(32,activation=’relu’))

classification.add(Dense(1,activation=’sigmoid’))

# compiling the ANN model

classification.compile(optimizer=’adam’,loss=’binary\_crossentropy’, metrics=[‘accuracy’])

# training the model

classification.fit(x\_train,y\_train,batch\_size=4,validation\_split=0.2,epochs=100)

Test the model

## decision tree

y\_pred=classifier.predict([[129,99,1,0,0,1,0,1,1,1,0,1,1,1,1,1]])

print(y\_pred)

(y\_pred)

## RandomForest

Y\_pred=rfc.predict([[129,99,1,0,0,1,0,1,1,1,0,1,1,1,1,1]])

Print(y\_pred)

(y\_pred)

Classification.save(‘flight.h5’)

# testing the model

y\_pred=classification.predict(x\_test)

y\_pred

y\_pred=(y\_pred>0.5)

y\_pred

def predict\_exit(sample\_value):

# Convert list to numpy array

sample\_value=np.array(sample\_value)

# Reshape because sample\_value contains only 1 record

sample\_value=sample\_value.reshape(1,-1)

# Feature scaling

sample\_value=sc.transform(sample\_value)

return classifier.predict(sample\_value)

test=classification.predict([[1,1,121.000000,36.0,0,0,1,0,1,1,1,1,1,1,1,1]])

if test==1:

print(‘prediction: chance of delay’)

else:

pring(‘prediction: no chance of delay.’)

compare the model

from sklearn import model\_selection

from sklearn.neural\_network import MLPClassifier

# RandomForest Accuracy

print(‘Training accuracy: ‘,accuracy\_score(y\_train, y\_redict\_train))

print (‘Testing accuracy: ‘,accuracy\_score(y\_test, y\_predict))

# Making the confusion matrix

From sklearn.metrics import confusion\_matrix

cm=confusion\_matrix(y\_test, y\_predict)

cm

# Accuracy score of desicionTree

From sklearn.metrics import accuracy\_score

desacc=accuracy\_score(y\_test, decisiontree)

desacc

from sklearn.metricsimport confusion\_matrix

cm=confusion\_matrix(y\_test,decisiontree)

cm

# Calculate the accuracy of ANN

from sklearn.metrics import accuracy\_score, classification\_report

score=accuracy\_score(y\_pred, y\_test)

print(‘The accuracy for ANN model is:{}%’.format(score\*100))

# Making the confusion matrix

from sklearn.metrics import confusion\_matrix

cm=confusion\_matrix(y\_test, y\_predd)

cm

Comparing model accuracy before & after applying hyperparameter tuning

# giving some parameters that can be used in randized search cv

Parameters={

‘n\_estimators’:[1,20,30,55,8,74,90,120,115],

‘criterion’:[‘gini,’entropy’],

‘max\_features’:[“auto”,”sqrt”,”log2”],

‘max\_depth’:[2,5,8,10], ‘verbose’: [1,2,3,4,6,8,9,10]

}

# performing the randomized cv

RCV= RandomizedSearchCV(estimator=rf, param\_distributions=parameters, cv=10, n\_iter=4)

RCV.fit(x\_train, y\_train)

# getting the best parameters from the giving list and best score from them

bt\_params=RCV.best\_params\_

bt\_score=RCV.best\_score\_

bt\_params

bt\_score

model=RandomForestClassifier(verbose=10,n\_estimators=120,max\_features=’log2’,max\_depth=10,criterion=’entropy’)

RCV.fit(x\_train, y\_train)

y\_predict\_rf=RCV.predict(x\_test)

RFC=accuracy\_score(y\_test,y\_predict\_rf)

RFC