|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **img2** | **IMG1** | **img5** | **img3** | **img6** |

**GOVERNMENT OF TAMILNADU**

**Naan Muthalvan - Project-Based Experiential Learning**

## FLIGHT DELAY PREDICTION FOR AVIATION INDUSTRY USING MACHINE LEARNING

Submitted by

**R.KEERTHANA – (20626ER047)**

**M.MOHANA – (20626ER049)**

**K.KOWSI– (20626ER048)**

**J.MOHANA PRIYA – (20626ER050)**

**Under the guidance of**

**Mrs.K.P.KAVITHA, MCA.,M.Phil.,(Ph.D)**

## Department of Computer Science



## GOVERNMENT ARTS COLLEGE FOR WOMEN , NILAKOTTAI

B(Affiliated To Mother Teresa Women’s University, Kodaikanal)

Reaccredited with ”C” Grade by NAAC

**NILAKOTTAI - 624208**

**APRIL - 2023**

## GOVERNMENT ARTS COLLEGE FOR WOMEN, NILAKOTTAI

## (Affiliated to Mother Teresa Womens University, Kodaikanal)

## Reaccredited with ‘C’ Grade by NAAC

## Nilakottai - 624208



## DEPARTMENT OF COMPUTER SCIENCE

## BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project entitled **OPTIMIZING FLIGHT DELAY PREDICTION FOR AVIATION INDUSTRY USING MACHINE LEARNING** done **By MS.R.KEERTHANA- (20626ER047), MS. M.MOHANA – (20626ER049),MS. K.KOWSI - (20626ER048), MS. J.MOHANA PRIYA - (20626ER050)** This is submitted in partial fulfillment for the award of the degree of  **Bachelor of Science in Computer Science in GOVERNMENT ARTS COLLEGE FOR WOMEN, NILAKOTTAI** during the period of December 2022 to April 2023.

**Project Mentor(s) Head of the Department**

**Submitted for viva-voce Examination held on** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**Flight Delay Prediction for Aviation**

**Industry Using Machine Learning**

**1. INTRODUCTION**

Air transportation system is one of the crucial modes of modern versatility. With increasing congestion in airtraffic and passenger-traffic, it is important to maintain persistence and resilience. Availability of land and resources contribute to the infrastructure of airports. The norms of improving technology and procedure are to maintain safety, efficiency, capacity, etc., Therefore, the National Airspace System (NAS) focuses on minimizing the environmental effects as a result of improvisation. With the current technology in hand, passengers can visualize their flight path, altitude, heading and other related parameters during their journey. However, air-traffic authorities continuously try to depreciate the delay in departure and arrival of flights. Though their efforts were in phase, the outcome is undesirable as the delays are in terms of hours sometimes causing chaos. Some important parameters that cause delay include weather, maintenance, security, and carrier. Corporate travel and tourism are the two major contributors to flight transportation system which is expected to be doubled by 2030. As a result of this increase, the airtraffic is also expected to increase in the same multiple. To minimize the air-traffic congestion new airports can be constructed. But, the complexity still grows exponentially. Hence, the only possible way of minimizing the delay is to improvise the existing airports. Considering the limited availability of land resources, the latter is more of a logical solution. Delay basically represents the period by which the aircraft is late or cancelled. Commercial aviation is likely to be affected if there is a delay in their mobility. This delay results in the dissatisfaction of trusted customers and sometimes even marketing strategies. With a view of understanding the flight system, scientists and researchers stored the vast amount of data recorded over the entire course of a flight journey.

**1.1Overview**

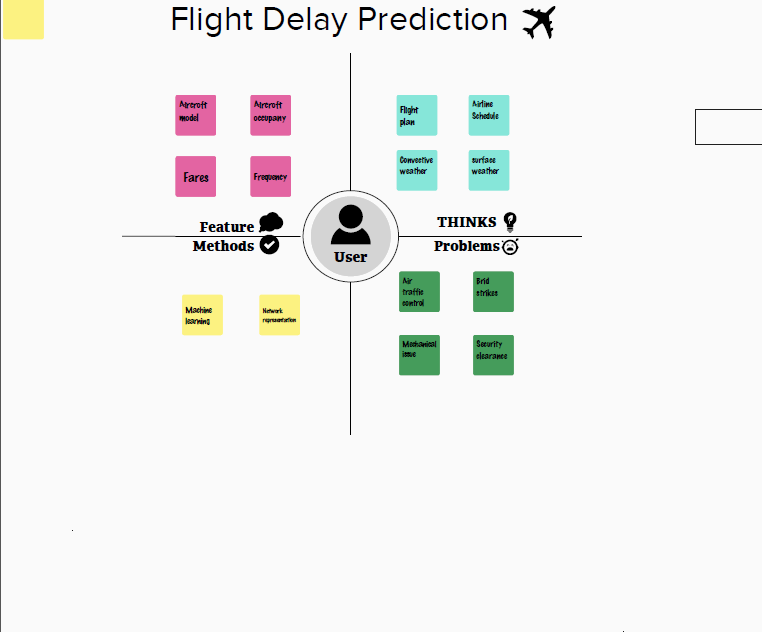
One of the major business problems that airlines face is the significant costs that are associated with flights being delayed due to natural occurrences and operational shortcomings, which is an expensive affair for the airlines, creating problems in scheduling and operations for the end-users thus causing bad reputation and customer dissatisfaction. In our paper, a two-stage predictive model was developed employing supervised machine learning algorithms for the prediction of flight ontime performance. The first stage of the model performs binary classification to predict the occurrence of flight delays and the second stage does regression to predict the value of the delay in minutes. The dataset used for evaluating the model was obtained from historical data which contains flight schedules and weather data for 5 years. It was observed that, in the classification stage, Gradient Boosting Classifier performed the best and in the regression stage, Extra-Trees Regressor performed the best. The performance of the other algorithms is also extensively documented in the paper. Furthermore, a real-time Decision Support Tool was built using the model which utilizes features that are readily available before the departure of an airplane and can inform passengers and airlines about flight delays in advance, helping them reduce possible monetary losses.

**2.1Purpose**

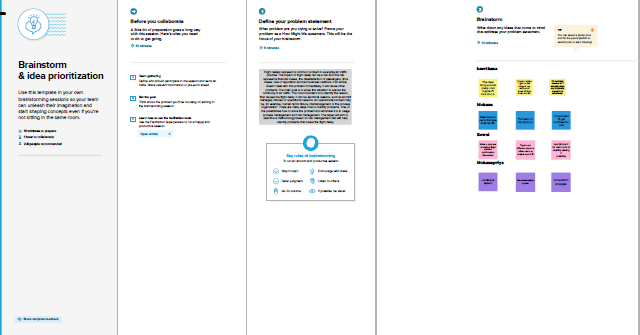
Therefore, predicting flight delays can improve airline operations and passenger satisfaction, which will result in a positive impact on the economy. In this study, the main goal is to compare the performance of machine learning classification algorithms when predicting flight delays.

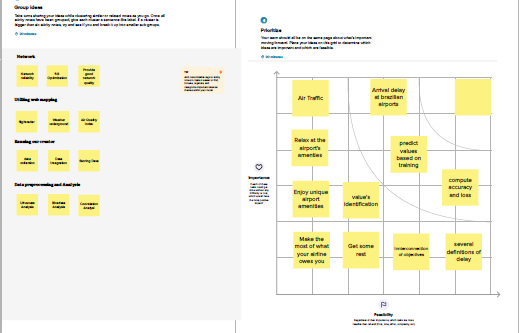
**2.Problem definition & Design Thinking**

**2.1 Empathy map**

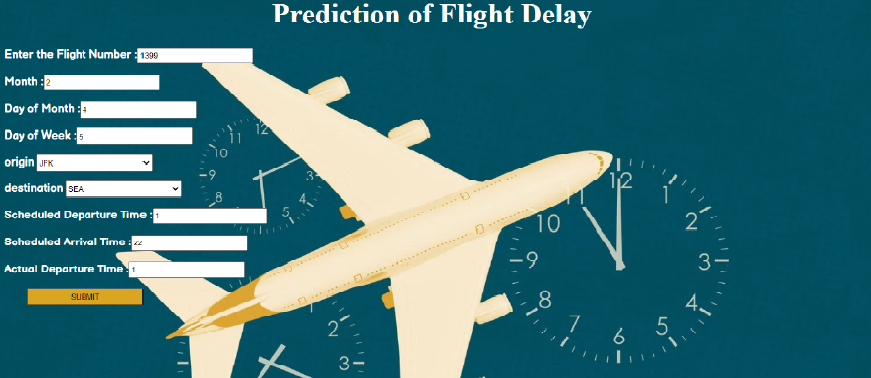


**2.2 Ideation & Brainstorming Map**





**2.RESULT**

**3.ADVANTAGES & DISADVANDAGES**

**ADVANTAGE**

* Due to the stochastic nature of delays, this research investigates the qualitative prediction of airline delays to implement necessary changes and provide better customer experience.

**DISADVANDAGE**

* Finding an accuracy of flight delay is less.
* It does not have required parameters for finding flight delay.

APPLICATION

The model is designed using Python in Tensor flow and is installed on a system of 40 core CPU at a frequency of 2.6 hz, 80 G RAM and 250 G Hard. The flight info data is an open dataset collected by the Bureau of Transportation Statistics of United State Department of Transportation [[163](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00380-z#ref-CR163)] where, the reason for delay is due to canceled or flight delay, and time duration of each flight. Model testing and training employs these data that include 18 million records.

Model, uses 80% of data for training and the remaining 20% for testing [[164](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00380-z#ref-CR164)]. Finally, the model evaluation considers two analysis which are studied in the following section.

**4.CONCLUSION**

Predicting flight delays is on interesting research topic and required many attentions these years. Majority of research have tried to develop and expand their models in order to increase the precision and accuracy of predicting flight delays. Since the issue of flights being on-time is very important, flight delay prediction models must have high precision and accuracy.Based on the analysis of their results, it is evident that the integration of multidimensional heterogeneous data, combined with the application of different techniques for feature selection and regression can provide promising tools for inference in the cancer domain. Regardless of the type of prediction task at hand; regression or classification. It has become the state-of-the-art machine learning algorithm to deal with structured data. Compare to all algorithms MLP algorithm gives high accuracy that is 82%.

**5.FUTURE SCOPE**

Based on the analysis of their results, it is evident that the integration of multidimensional heterogeneous data, combined with the application of different techniques for feature selection and regression can provide promising tools for inference in the cancer domain. The XGBoost is used in the analysis of this paper because XGBoost is one of the most popular machine learning algorithms these days. Regardless of the type of prediction task at hand; regression or classification. It has become the state-of-the-art machine learning algorithm to deal with structured data.

**APPENDIX**

**Source code**

import pandas as pd

import numpy as np

import  pickle

import matplotlib.pyplot as pandas

%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("/content/flightdata (1).csv")

dataset.head()

dataset.info()

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

dataset.isnull().sum()

dataset - dataset[["FL\_NUM", "MONTH", "DAY\_OF\_MONTH", "DAY\_OF\_WEEK", "ORIGIN", "DEST", "CRS\_ARR\_TIME","DEP\_DEL15", "ARR\_DEL15"]]

dataset.isnull().sum()

dataset[dataset.isnull().any(axis-1)].head(18)

dataset ['DEP\_DEL15'].mode()

import math

for intex, row in dataset.iterrows():

  dataset.loc[intex,'CRS\_ARR\_TIME'] = mathh.floor(row['CRS\_ARR\_TIME']/100)

dataset.head()

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

dataset['DEST'] = le.fit\_transform(dataset['DEST'])

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

dataset.head(5)

dataset['ORIGIN'].unique()

dataset - pd.get\_dummies(dataset, columns-['ORIGIN', 'DEST'])

x = dataset.iloc[:, 0:8].values

y = dataset.iloc[:, 8:9].values

x

from sklearn.preprocessing import OneHotEncoder

oh = OneHotEncoder()

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

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

z

t

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

flight\_data.describe()

sns.distplot(flight\_data.MONTH)

sns.catplot(x="ARR\_DEL15",y="ARR\_DELAY",kind='bar',data=flight\_data)

sns.heatmap(dataset.corr())

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(dataset.drop('ARR\_DEL15', axis=1), df['ARR\_DEL15'], test\_size=0.2, random\_state=0)

x\_test.shape

x\_test.shape

y\_test.shape

y\_train.shape

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

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

x\_test = sc.transform(x\_test)

classifier  = DecisionTreeClassifier(random\_state = 0)

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

decisiontree = classifier.predict(x\_test)

decisiontree

from sklearn.metrics import accuracy\_score

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

from sklearn.ensemble import RandomForestClassifier

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

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

y\_predict = rfc.predict(x\_test)

# 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)

#  Decision tree

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

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]])

print(y\_pred)

(y\_pred)

classification.save('flight.h5')

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

y\_pred

y\_pred = (y\_pred > 0.5)

y\_pred

def predict\_exit(sample\_value)

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

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

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

return classifier.pedict(sample\_value)

test=classificaion.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:

   print('prediction: No chance of delay')

from sklearn import model\_selection

from sklearn. neural\_network import MLPClassifier

   dfs = []

models = [

           ('RF',Randmclassifer()),

          ("DecisionTree",DecisionTreeClassifier())

          ('ANN',MLPClassifier())

         ]

results = []

   names = []

  scoring = [‘accuray’,’precision\_weighted’,’recall\_weighted’,’f1\_weighted’,’roc\_auc’]

target names =[‘no delay’,’delay’]

for name,model in models:

kfold = model\_selection.(n\_splits=5, shuffle=True, random\_state=90210)

cv\_results=model\_selection.cross\_validate(model,x\_trainy\_train,cv=kfold,scoring=scori)

clf\_results = model  .fit(x\_train,y\_train)

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

print(name)

print(classification\_report(y\_test,y\_pred,target\_names=target\_names))

results.append(cv\_results)

names.append(name)

this\_df = pd.DatFrame(cv\_results)

this\_df[‘model’] = name

dfs.append(this\_df)

final = pd.concat(dfs, ignore\_index=True

return final

# RandomForest Accuracy

Print(‘Training accuracy:’,accuracy\_score(y\_train,y\_predict\_train

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

#Making the Confusion Matrix

From sklearn.metrics import confusion\_matrix

Cm = confusion\_matrix(y\_predict)

Cm

# Accuracy score of desicionTree

From sklearn.mettrics import accuracy\_score

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

From sklearn.metrics import confusion\_matrix

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

#Calculate the Accuracy of ANN

From sklearn.metrica 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

Form sklearn.metrics import confusion\_matrix

Cm = confusion\_matrix(y\_test,y\_pred)

Cm#giving some parmeters that can be used in randized search cv

Parameters = {

‘n\_estimators’ : [1,20,30,55,68,74,90,120,115],

‘criterion’:[‘gini’,’entropy’],

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

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

}

#performing the randomized cv

RVC = RandomizedSearchCV(estimator=param\_distributions=parameters,cv=10,iter=4)

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

#getting the best parmeters form the giving list and best score from them

Bt\_params = RVC.best\_params

Bt\_score = RVC.best\_score\_

Model = RndomForestClassifier(verbose= 10, n\_estimators= 120, max\_feauters= ‘log2’,max\_depth= 10,criterion=’entropy’)

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

Y\_predict\_rf = RVC.predict(x\_test)

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

RFC

Import pickle

Pickle.dump(RVC,open(‘flight.pkl’,’wb’))

#importing the necessary dependencies

From flask import Flask,request,render\_template

import numpy as np

import pandas as pd

import pickle

import os

model = pickle.load(open(‘flight.pkl’,’rb’))

app = Flask(\_\_name\_\_)#initializing the app

@app.route(‘/’)

@app.route(‘/prediction’, =[‘POST’])

Def predict()

Name = request.from[‘name’]

month = request.from[‘month’]

dayofmonth = request.form[‘dayofmonth’]

dayofweek = request.form[‘dayofweek’]

origin = request.from[‘origin’]

if(origin ==”msp”):  
 origin1,origin2,origin3,origin4,origin5 = 0,0,0,0,1

if(origin ==”dtw”)”

origin1,origin2,origin3,origin4,origin5 = 1,0,0,0,0

if(origin ==”jfk”):

origin1,origin2,origin3,origin4,origin5 =0,0,1,0,0

if(origin ==”sea”):  
 origin1,origin2,origin3,origin4,origin5 = 0,1,0,0,0

if(origin ==”alt”):

origin1,origin2,origin3,origin4,origin5 = 0,0,0,1,0

destination =request.from[‘destination’]

if (destination == “msp”):

destination1,destination2,destination3,destination4,destination5 = 0,0,0,0,1

if(destination ==”dtw”):

destination1,destination2,destination3,destination4,destination5 = 1,0,0,0,0

if(destination ==”jfk”):

destination1,destination2,destination3,destination4,destination5 = 0,0,1,0,0

if(destination ==”sea”):

destination1,destination2,destination3,destination4,destination5 = 0,1,0,0,0

if(destination ==”alt”):

destination1,destination2,destination3,destination4,destination5 = 0,0,0,1,0

dept = request.form[‘dept’]

arrtime = request.form[‘actdept’]

deptl5=int(dept)-int(actept)

total = [[name,month,dayofmonth,dayofweek,origin1,origin2,origin3,origin4,origin5,destination1,destination2,destination3,destination4,destination5,i

#print(total)

Y\_pred = model.predict(total)

Print(y\_pred)

if(y\_pred==[0.]):

ans=”The Flight will be on time”

else:

ans=”The Flight will be delayed”

return render\_template(“index.html”,showcase = ans)

if\_\_name\_\_==’\_\_main\_\_’:

app.run(debug = True)|

4