FLIGHT DELAY PREDICTION FOR AVIATION INDUSTRY USING MACHINE LEARNING

1. INTRODUCTION

1.1 OVERVIEW

Project is about predicting flight delays in the aviation industry using machine learning techniques. We are using input parameters such as distance and weather details to make a decision of whether the specific flight is delayed or not.

The goal of your project is to improve the accuracy of flight delay predictions and help airlines better manage their schedules and resources.

The goal is to develop a model that can accurately predict flight delays based on various factors such as weather conditions, air traffic congestion, and mechanical issues. The model should be able to handle large amounts of data and provide accurate predictions in real-time.

The solution to your problem is to develop a machine learning model that can accurately predict flight delays based on various factors such as weather conditions, air traffic congestion, and mechanical issues. The model should be able to handle large amounts of data and provide accurate predictions in real-time.

To develop the model, you will need to collect and preprocess the data, select appropriate features, and train and evaluate the model using various machine learning techniques. You can then use the model to make predictions on new data.

Flight delay prediction is a common problem in the aviation industry and machine learning can be used to solve this problem.

There are many factors that can contribute to flight delays such as weather conditions, air traffic congestion, and mechanical issues.

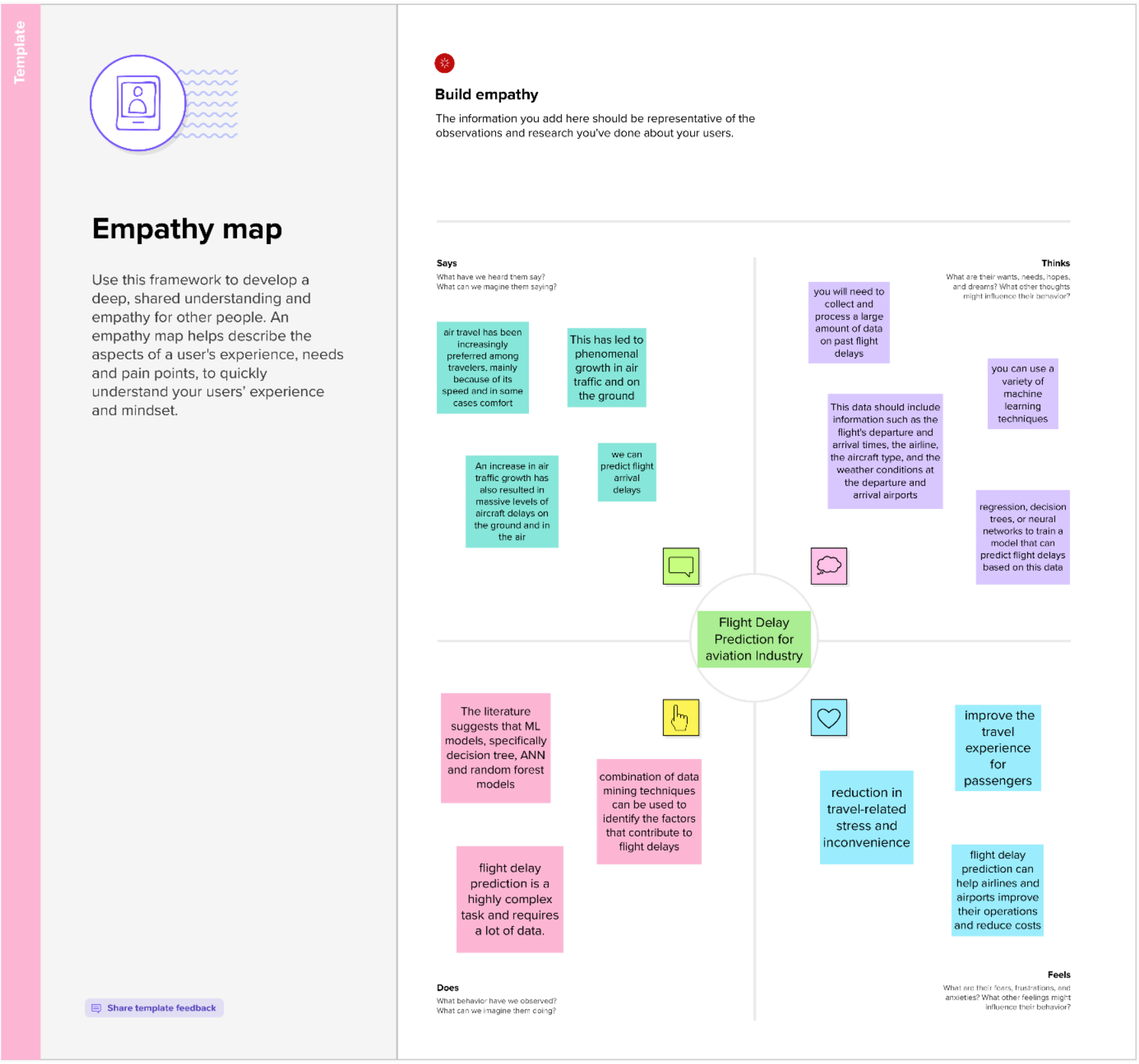
1.2 PURPOSE

The purpose of our project is to improve the accuracy of flight delay predictions and help airlines better manage their schedules and resources. By using machine learning techniques and input parameters such as distance and weather details, you are able to make a decision of whether the specific flight is delayed or not.

2. PROBLEM DEFINITION AND DESIGN THINKING

2.1 EMPATHY MAP

- ✓ Different machine learning algorithms such as decision trees, random forests, and neural networks to see which one works best for your data
- ✓ Feature engineering techniques such as one-hot encoding, scaling, and normalization to improve the performance of your model
- ✓ Hyperparameter tuning techniques
- ✓ Ensemble methods
- ✓ Data augmentation techniques



Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we magine them saying?

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

we can predict flight arrival delays

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

you will need to collect and process a large amount of data on past flight delays

you can use a variety of machine learning techniques

This data should include information such as the flight's departure and arrival times, the airline, the aircraft type, and the weather conditions at the departure and arrival airports

regression, decision trees, or neural networks to train a model that can predict flight delays based on this data

Flight Delay Prediction for aviation Industry

The literature suggests that ML models, specifically decision tree, ANN and random forest models

combination of data mining techniques can be used to identify the factors that contribute to flight delays

flight delay prediction is a highly complex task and requires a lot of data.

Does

What behavior have we observed?
What can we imagine them doing?

improve the travel experience for passengers

reduction in travel-related stress and inconvenience

flight delay prediction can help airlines and airports improve their operations and reduce costs

Feels

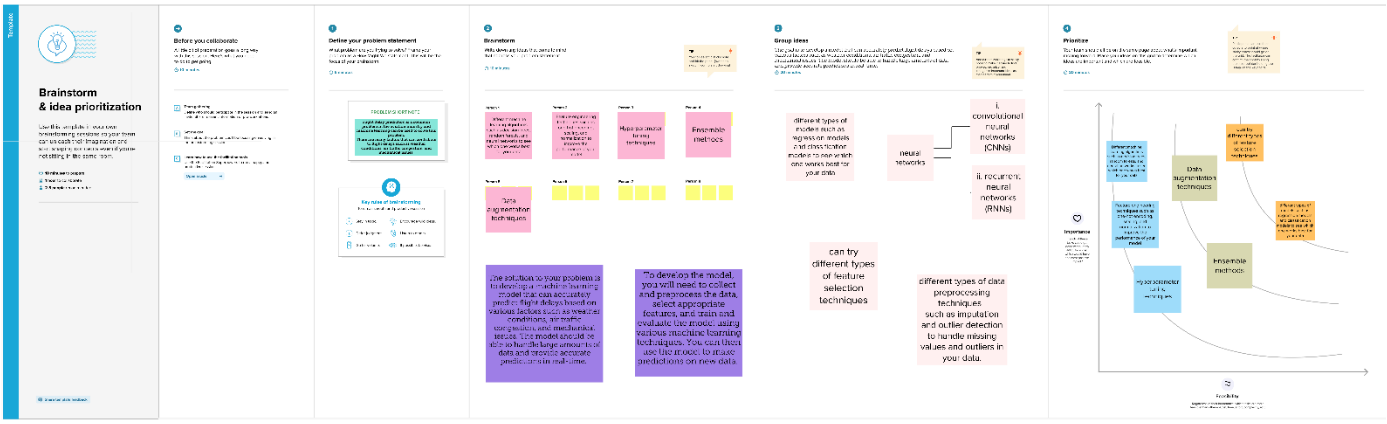
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

2.2 IDEATION AND BRAINSTORMING MAP

- ✓ Different types of models such as regression models and classification models to see which one works best for your data

- ✓ neural networks
  - convolutional neural networks (CNNs)
  - recurrent neural networks (RNNs)

- ✓ can try different types of feature selection techniques
- ✓ different types of data preprocessing techniques
  such as imputation and outlier detection to handle missing values and outliers in your data.



Brainstorm & idea prioritization

Before you collaborate

Define your problem statement

Brainstorm

Hyperparameter tuning techniques

Ensemble methods

Data augmentation techniques

The solution to your problem is to develop a machine learning model that can accurately predict flight delays based on various factors such as weather conditions, air traffic congestion, and mechanical issues. The model should be able to handle large amounts of data and provide accurate predictions in real-time.

To develop the model, you will need to collect and preprocess the data, select appropriate features, and train and evaluate the model using various machine learning techniques. You can then use the model to make predictions on new data.

Group ideas

different types of models such as regression models and classification models to see which one works best for your data

neural networks

i. convolutional neural networks (CNNs)

ii. recurrent neural networks (RNNs)

can try different types of feature selection techniques

different types of data preprocessing techniques such as imputation and outlier detection to handle missing values and outliers in your data.

Prioritize

Data augmentation techniques

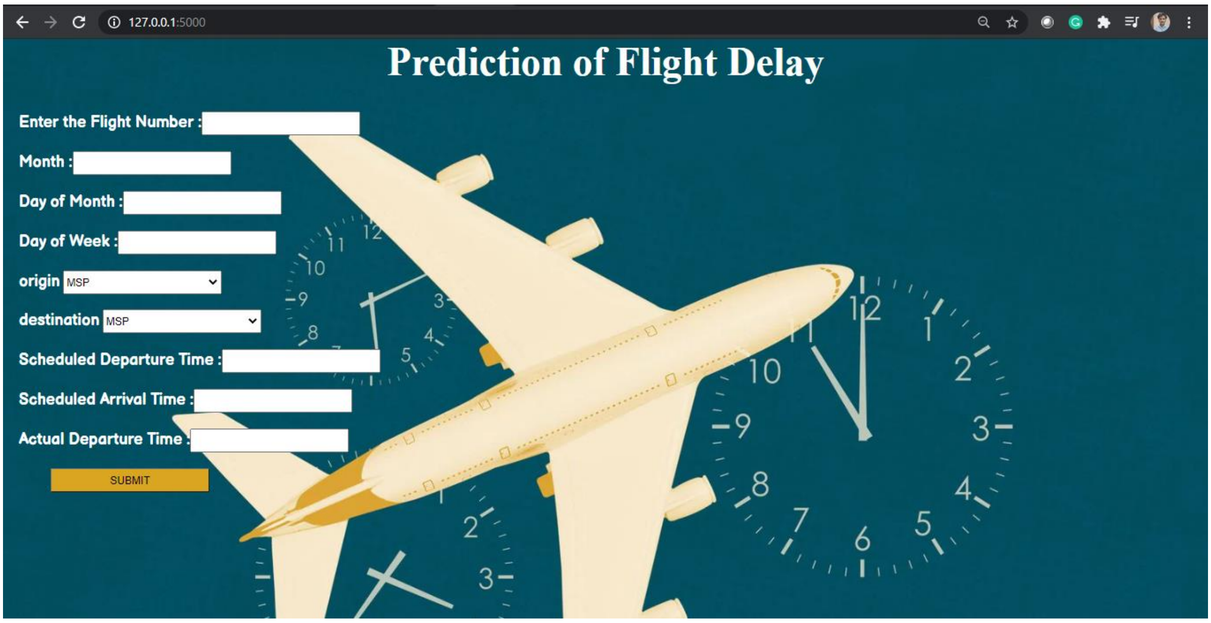
Ensemble methods

Hyperparameter tuning techniques

3. RESULT

Flight delay prediction is an important problem in the aviation industry as it can help optimize flight operations and minimize delays. There are several machine learning algorithms that have been proposed to predict flight delays
Most studies predict flight delays using binary classifiers (delayed/not delayed flight), Multi-class classifiers (multiple delay classes), or Regression (estimating the delay value).

Our project seems to use a decision tree classifier to predict if the flight arrival will be delayed or not.
A flight is delayed when the difference between scheduled and actual arrival times is greater than 15 minutes.Furthermore, you compare decision tree classifier with logistic regression and a simple neural network for various figures of merit. Finally, it will be integrated into a web-based application.



127.0.0.1:5000

Prediction of Flight Delay

Enter the Flight Number :

Month :

Day of Month :

Day of Week :

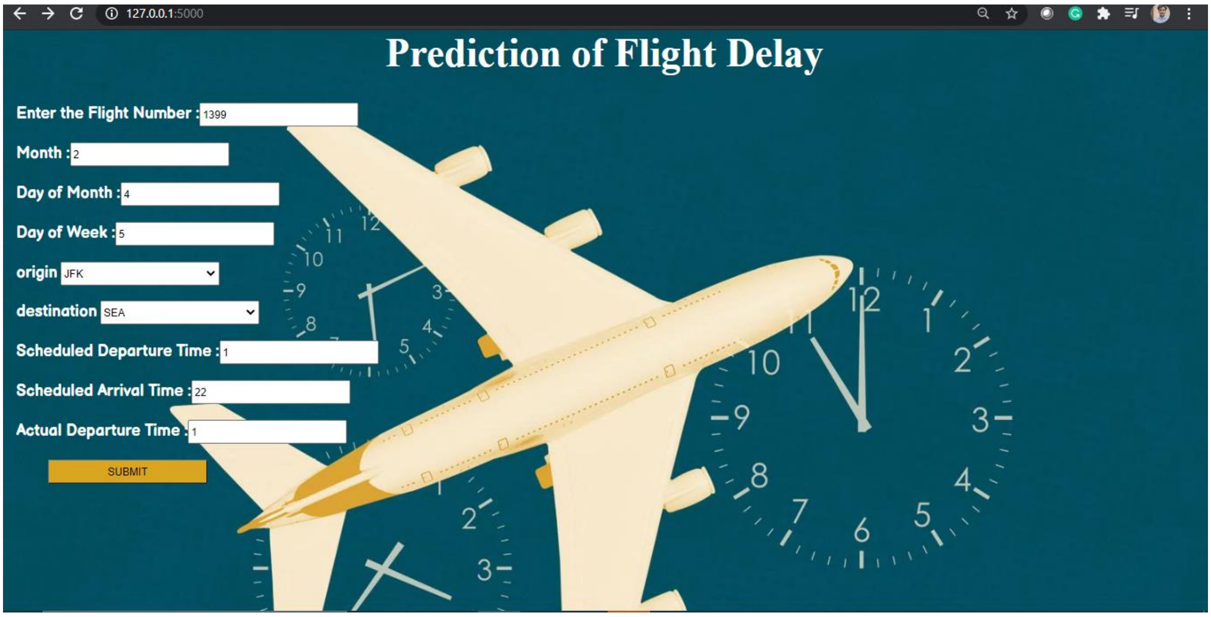
origin MSP

destination MSP

Scheduled Departure Time :

Scheduled Arrival Time :

Actual Departure Time :



127.0.0.1:5000

Prediction of Flight Delay

Enter the Flight Number : 1399

Month : 2

Day of Month : 4

Day of Week : 5

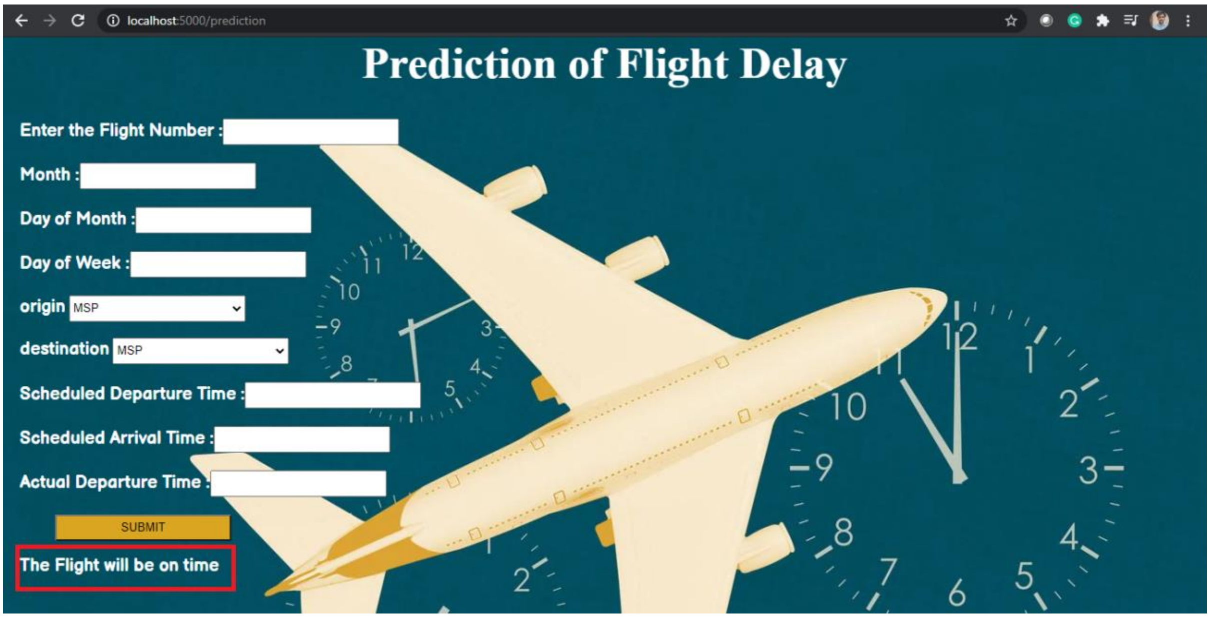
origin JFK

destination SEA

Scheduled Departure Time : 1

Scheduled Arrival Time : 22

Actual Departure Time : 1



Prediction of Flight Delay

Enter the Flight Number :

Month :

Day of Month :

Day of Week :

origin MSP

destination MSP

Scheduled Departure Time :

Scheduled Arrival Time :

Actual Departure Time :

The Flight will be on time

4. ADVANTAGES AND DISADVANTAGES

4.1 ADVANTAGES

- ✓ There are several advantages of using machine learning for flight delay prediction.
- ✓ Accurate flight delay prediction is fundamental to establish more efficient airline business.
- ✓ Recent studies have been focused on applying machine learning methods to predict flight delay.
- ✓ Most of the previous prediction methods are conducted in a single route or airport.
- ✓ However, some studies have explored a broader scope of factors that may potentially influence flight delay and compared several machine learning-based models in designed generalized flight delay prediction tasks.

- ➢ One study proposed a probabilistic flight delay prediction model that reduces the number of conflicted aircraft by up to 74% when compared to a deterministic flight-to-gate assignment model.
- ➢ Another study used supervised machine learning models to predict flight delays and found that these models can help resolve the situation of delayed flights.

4.2 DISADVANTAGES

- ✓ There are some limitations to using machine learning for flight delay prediction.
- ✓ One study found that one of the newest models in the area of machine learning has drawbacks of overfitting.
- ✓ Researchers solved this issue through typical data dropout techniques for each step of the repeated training procedure.
- ✓ Another study found that several reasons are restricting the existing methods from improving the accuracy of flight delay prediction.
- ✓ The reasons are summarized, such as the variety of causes that affect the flight delays, the complexity and relevancy between causes, and also the insufficient availability of flight data.

5. APPLICATIONS

Flight delay prediction using machine learning can be applied in the aviation industry to increase customer satisfaction and incomes of airline agencies. One study found that probabilistic forecasting algorithms can be applied to predict flight delays at a European airport. Another study developed a two-stage predictive machine learning engine that forecasts the on-time performance of flights for 15 different airports in the USA based on data collected in 2016 and 2017.

6. CONCLUSION

To summarize the entire work, you can start by briefly describing the problem statement and the motivation behind the project. Then, you can describe the methodology used to solve the problem and the results obtained. Finally, you can conclude by summarizing the key findings and their implications.

7. FUTURE SCOPE

Enhancements that can be made in the future for the above project could include improving the accuracy of the machine learning models used to predict flight delays by incorporating additional features such as real-time weather data and flight traffic data.

Additionally, the models could be trained on larger datasets to improve their performance.

Another potential enhancement could be to develop a user-friendly interface that allows passengers to access flight delay predictions in real-time and receive notifications of any changes to their travel plans.

APPENDIX

SOURCE CODE

app.py

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


# importing the necessary dependencies
from flask import Flask,request,render_template
import numpy as np
import pandas as pd
import pickle
import os


# In[1]:


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

app = Flask(__name__) # initializing the app


# In[2]:


@app.route('/')
def home():
    return render_template("index.html")

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


# In[5]:


def predict():
    name = request.form['name']
    month = request.form['month']
    dayofmonth = request.form['dayofmonth']
    dayofweek = request.form['dayofweek']
    origin = request.form['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


# In[3]:


destination = request.form['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['arrtime']
arrtime = request.form['actdept']
dapt15=int(dept)-int[actdept]
total =
[[name,month,dayofmonth,dayofweek,origin1,origin2,origin3,origin4,origin5,destination1,de
stination2,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)


# In[4]:


if __name__ == '__main__':
    app.run(debug = True)
```

code.ipynb

```
#!/usr/bin/env python
# coding: utf-8

# In[14]:


import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
get_ipython().run_line_magic('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


# In[15]:


dataset=pd.read_csv(r"C:\Users\Gowthami\Flight Delay Prediction for aviation Industry using Machine Learning\flightdata.csv")


# In[16]:


dataset.head()


# In[7]:


dataset.info()


# In[17]:


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

```
dataset.isnull().sum()


# In[12]:


dataset = dataset[["FL_NUM", "MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK",
"ORIGIN", "DEST", "CRS_ARR_TIME", "DEP_DEL15", "ARR_DEL15"]]
dataset.isnull().sum()


# In[19]:


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


# In[18]:


dataset['DEP_DEL15'].mode()


# In[20]:


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


# In[ ]:


# HANDLING CATEGORICAL VALUES


# In[22]:


import math

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


# In[26]:
```

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


# In[34]:


dataset.head(5)


# In[18]:


dataset['ORIGIN'].unique()


# In[19]:


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


# In[20]:


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


# In[38]:


x


# In[21]:


from sklearn.preprocessing import OneHotEncoder
oh = OneHotEncoder()
z=oh.fit_transform(x[:,4:5]).toarray()
t=oh.fit_transform(x[:,5:6]).toarray()


# In[22]:
```

```
z


# In[25]:


t


# In[26]:


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


# In[ ]:


# EXPLORATORY DATA ANALYSIS
# DESCRIPTIVE STATISTICAL


# In[29]:


dataset.describe()


# In[30]:


sns.distplot(dataset.MONTH)


# In[31]:


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

# In[32]:

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

# In[4]:

import seaborn as sns
import pandas as pd

dataset=pd.read_csv(r"C:\Users\Gowthami\Flight Delay Prediction for aviation Industry
using Machine Learning\Dataset\flightdata.csv")
```

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

# In[5]:


# i am changing some function words
#import pandas as pd
dataset = pd.get_dummies(dataset, columns=['ORIGIN', 'DEST'])
dataset.head()
# In[6]:


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

# In[7]:

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)

# In[3]:

# define darafame
import pandas as pd
from sklearn.model_selection import train_test_split

# Load your dataset into a DataFrame
df = pd.read_csv(r"C:\Users\Gowthami\Flight Delay Prediction for aviation Industry using Machine Learning\Dataset\flightdata.csv")

# Split the dataset into training and testing sets
train_x, test_x, train_y, test_y = train_test_split(df.drop('ARR_DEL15', axis=1), df['ARR_DEL15'], test_size=0.2, random_state=0

# In[4]:

x_test = test_x
x_test.shape
# defferent ans you konw

# In[13

x_train.shape
# same like

# In[14

y_test.shape
```

```
# In[15]

y_train.shape


# In[ ]:

# Scaling the data

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


# In[1]:
# MODEL BUILDING
# In[15]:

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


# In[ ]:

decisiontree = classifier.predict(x_test)

# In[61]:


decisiontree
# In[ ]:

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

# In[2]:

# Random Forest Model

# In[64]:

from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=10,criterion='entropy')

# In[17]:

rfc.fit(x_train,y_train)
```

```
# In[ ]:
y_predict = rfc.predict(x_test)

# In[ ]:

# Importing the Keras Libraries and packages

import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
# In[ ]:
# 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'))


# In[ ]

# Compiling the ANN model

classification.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

# In[ ]:

# Training the model

classification.fit(x_train,y_train,batch_size=4,validation_split=0.2,epochs=100)

# In[18]:

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)

# In[19]:

##  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)

# In[ ]:
```

```
classification.save('flight.h5')

# In[ ]:

#testing the model

y_pred = classification.predict(x_test)

# In[23]:

y_pred

# In[24]

y_pred = (y_pred > 0.5)
y_pred

# In[80]:

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)


# In[25]:
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:
    print('Prediction: No chance of delay.')


# In[8]:


#Performance Testing & Hyperparameter Tuning
# In[82]:


from sklearn import model_selection
from sklearn.neural_network import MLPClassifier
```

```
# In[7]:
dfs = []
models =[
      ('RF', RandomForestClassifier()),
      ('DecisionTree',DecisionTreeClassifier()),
      ('ANN',MLPClassifier())
    ]
results = []
names = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted', 'roc_auc']
target_names = ['no delay', 'delay']
for name, model in models:
    kflod = model_selection.KFold(n_splits=5, shuffle=True, random_state=90210)
    cv_results = model_selection.cross_validate(model, x_train, y_train, cv=kfold,
scoring=scoring)
    clf = 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(name)
    this_df = pd.DataFrame(cv_results)
    this_df['model'] = name
    dfs.append(this_df)
final = pd.concat(dfs, ignore_index=True)
return final

# In[28]:


# RandomForest Accuracy

print('Training accuracy: ',accuracy_score(y_train,y_predict_train))
print('Testing accuracy: ',accuracy_score(y_test,y_predict))


# In[29]:

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm

# In[13]:

# Accuracy score of desicionTree

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

# In[30]:
```

```
desacc

# In[31]:

cm


# In[32]:
# 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))


# In[33]

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm


# In[22]:
# Giving some parameters 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' : [1,2,3,4,6,8,9,10]
}

# In[34]:

# performing the randomized cv
RCV = RandomizedSearchCV(estimator=rf,param_distributions=parametes,cv=10,n_iter=4)


# In[35]

RCV.fit(x_train,y_train)

# In[36]:

# getting the best paarmets from the giving list and best score from them
bt_params = RCV.best_params_
bt_score = RCV.best_score_

# In[37]:
```

```
bt_params
# In[38]:

bt_score

# In[39]:

model = RandomForestClassifier(verbose= 10, n_estimators= 120, max_features= 'log2',max_depth= 10,criterion= 'entropy')
RCV.fit(x_train,y_train)

# In[40]:

y_predict_rf = RCV.predict(x_test)

# In[41]:

RFC=accuracy_score(y_test,y_predict_rf)
RFC


# In[31]:
# MODEL DEPLOYMENT

# In[42]:

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

TEAM MEMBERS :

- Team Leader : JANANI K
- Team member 1 : SHALINI S
- Team member 2 : GOWTHAMI P
- Team member 3 : NANDHINI K