CHIKKANNA GOVERNMENT ARTS COLLEGE

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION

TIRUPUR-641602

(AFFILIATED TO BHARATHIAR UNIVERSITY)

TEAM MEMBERS NAME :

VIJAYASINGAM A (2022J0059)

MURUGESAN S (2022J0046)

AJAY KUMAR P (2022J0011)

MAHESH KUMAR B (2022J0040)

Flight Delay Prediction For Aviation Industry Using Machine Learning

1.INTRODUCTION

1.1 OVERVIEW :

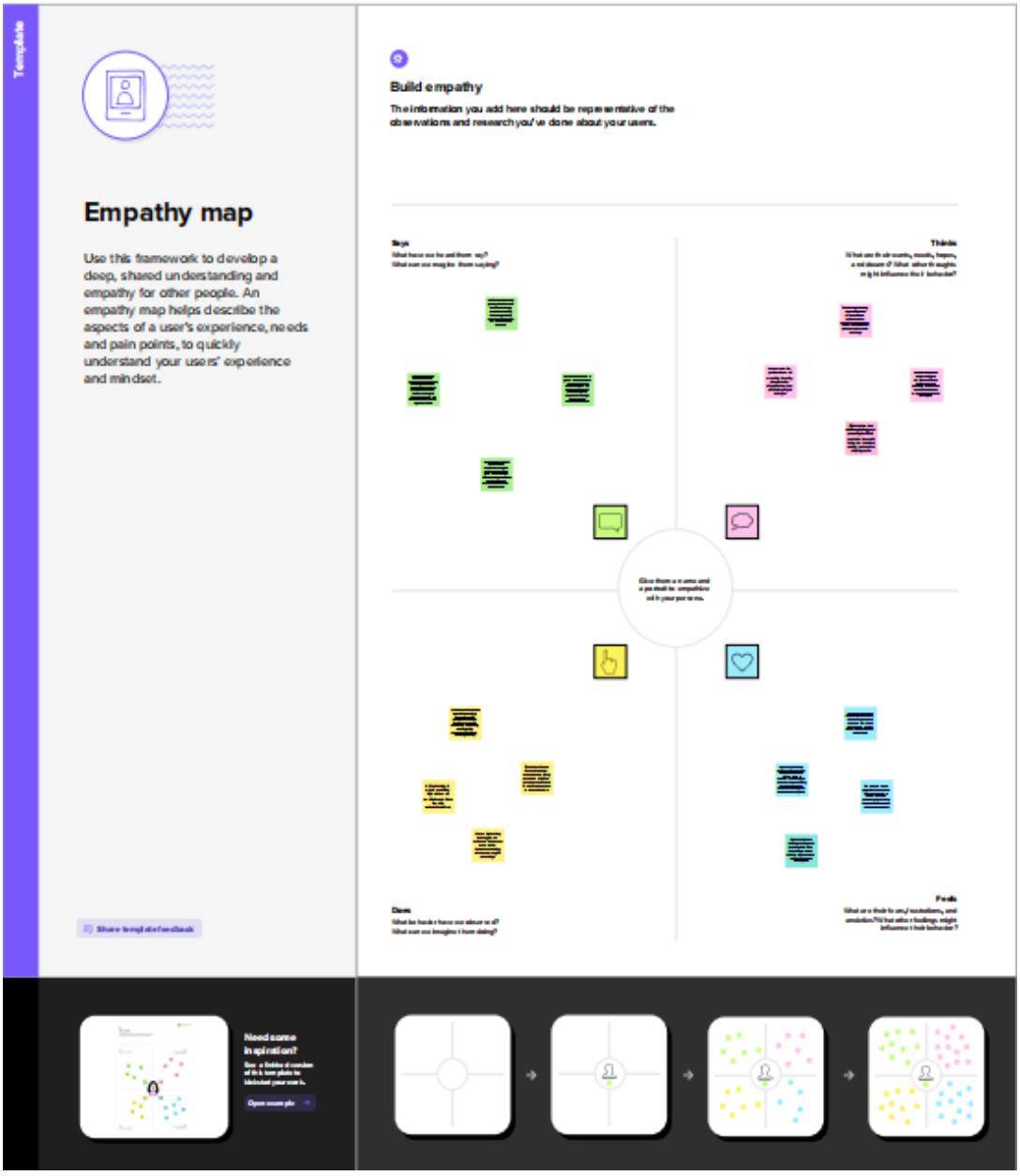
In the present world, the major components of any transportation system include passenger airline, cargo airline, and air traffic control system. With the passage of time, nations around the world have tried to evolve numerous techniques of improving the airline transportation system. This has brought drastic change in the airline operations. Flight delays occasionally cause inconvenience to the modern passengers . Every year approximately 20% of airline flights are canceled or delayed, costing passengers more than 20 billion dollars in money and their time.

1.2. PURPOSE :

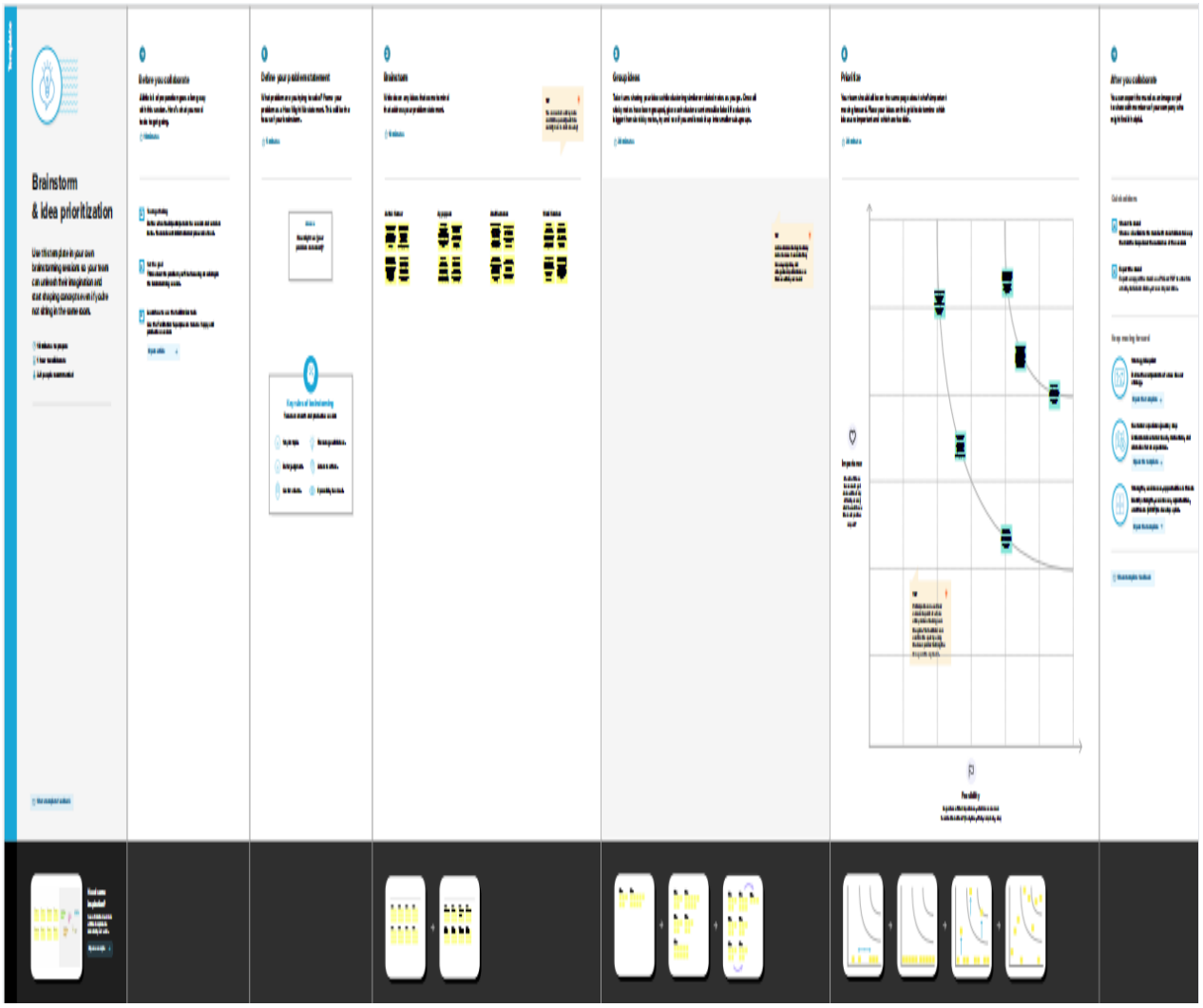
Average aircraft delay is regularly referred to as an indication of airport capacity. Flight delay is a prevailing problem in this world. It's very tough to explain the reason for a delay. A few factors responsible for the flight delays like runway construction to excessive traffic are rare, but bad weather seems to be a common cause. Some flights are delayed because of the reactionary delays, due to the late arrival of the previous flight. It hurts airports, airlines, and affects a company's marketing strategies as companies rely on customer loyalty to support their frequent flying programs.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1. EMPATHY MAP



2.2 IDEATION & BRAINSTORMINGS MAP



3. RESULT


Prediction of Flight Delay
Enter the Flight Number :
Enter Flight Number
Month :
Enter Month
Day of Month :
Enter Day of Month
Day of Week :
Enter Day of Week
Origin :
MSP
Destination :
MSP
Scheduled Departure Time :
Enter Departure Time
Scheduled Arrival Time :
Enter Arrival Time
Actual Departure Time :
Enter Actual Time
Submit

4.ADVANTAGES & DISADVANTAGES

ADVANTAGES :

- There are no federal laws requiring airlines to provide passengers with money or other compensation when their flights are delayed.
- Each airline has its own policies about what it will do for delayed passengers.
- If your flight is experiencing a long delay,ask airline staff if they will pay for meals or a hotel room.

DISADVANTAGES :

- Carriers attribute flight delays to several causes such as bad weather conditions, airport congestion, and use of smaller aircraft by airlines.
- These delays and cancellations tarnish the airlines reputation, often resulting in loss of demand by passengers.
- Cause a decrease in efficiency, an increase in capital costs, reallocation of flight crews and aircraft, and additional crew expenses.

5. APPLICATION

The problem of flight delay prediction is approached most often by predicting a delay class or value. However, the aviation industry can benefit greatly from probabilistic delay predictions on an individual flight basis, as these give insight into the uncertainty of the delay predictions. Therefore, in this study, two probabilistic forecasting algorithms, Mixture Density Networks and Random Forest regression, are applied to predict flight delays at a European airport. The algorithms estimate well the distribution of arrival and departure flight delays with a Mean Absolute Error of less than 15 min. To illustrate the utility of the estimated delay distributions, we integrate these probabilistic predictions into a probabilistic flight-to-gate assignment problem. The objective of this problem is to increase the robustness of flight-to-gate assignments. Considering probabilistic delay predictions, our proposed flight-to-gate assignment model reduces the number of conflicted aircraft by up to 74% when compared to a deterministic flight-to-gate assignment model. In general, the results illustrate the utility of considering probabilistic forecasting for robust airport operations' optimization.

6. CONCLUSION

In this project, we use flight data, weather, and demand data to predict flight departure delay. Our result shows that the Random Forest method yields the best performance compared to the SVM model. Somehow the SVM model is very time consuming and does not necessarily produce better results. In the end, our model correctly predicts 91% of the non-delayed flights. However, the delayed flights are only correctly predicted 41% of time. As a result, there can be additional features related to the causes of flight delay that are not yet discovered using our existing data sources. In the second part of the project, we can see that it is possible to predict flight delay patterns from just the volume of concurrently published tweets, and their sentiment and objectivity. This is not unreasonable; people tend to post about airport delays on Twitter; it stands to reason that these posts would become more frequent, and more profoundly emotional, as the delays get worse. Without more data, we cannot make a robust model and find out the role of related factors and chance on these results. However, as a proof of concept, there is potential for these results. It may be possible to routinely use tweets to ascertain an understanding of concurrent airline delays and traffic patterns, which could be useful in a variety of circumstances.

7. FUTURE SCOPE

This project is based on data analysis from year 2008. A large dataset is available from 1987-2008 but handling a bigger dataset requires a great amount of preprocessing and cleaning of the data. Therefore, the future work of this project includes incorporating a larger dataset. There are many different ways to preprocess a larger dataset like running a Spark cluster over a server or using a cloud-based services like AWS and Azure to process the data. With the new advancement in the field of deep learning, we can use Neural Networks algorithm on the flight and weather data. Neural Network works on the pattern matching methodology. It is divided into three basic parts for data modelling that includes feed forward networks, feedback networks, and selforganization network. Feed-forward and feedback networks are generally used in the areas of prediction, pattern recognition, associative memory, and optimization calculation, whereas self-organization networks are generally used in cluster analysis. Neural Network offers distributed computer architecture with important learning abilities to represent nonlinear relationships. Also, the scope of this project is very much confined to flight and weather data of United States, but we can include more countries like China, India, and Russia. Expanding the scope of this project, we can also add the flight data from international flights and not just restrict our self to the domestic flights.

8. APPENDIX

A.SOURCE CODE :

```
import numpy as np
import pickle
import matplotlib.pyplot as plt
%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("flightdata.csv")


print(dataset.head())
```

```
print(dataset.info())


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


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


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


print(dataset['DEP_DEL15'].mode())


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


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


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

```
#print(dataset['ORIGIN'])

print(dataset.head(5))

print(dataset['ORIGIN'].unique())

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

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

print(x)

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

print(z)
print(t)

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

print(dataset.describe())

print(sns.distplot(dataset.MONTH))
```

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

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

print(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,text_x,train_y,test_y=train_test_split(dataset.drop('ARR_DEL15',axis=1),dataset['ARR_
DEL15'],test_size=0.2,random_state=0)

print(x_test.shape)

print(x_train.shape)

print(y_test.shape)

print(y_train.shape)

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

x_train=sc.fit_transform(x_train)

x_test=sc.transform(x_test)
```

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

x_test=np.nan_to_num(x_test)
decisiontree=classifier.predict(x_test)

print(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')

print(rfc.fit(x_train,y_train))

y_predict=rfc.predict(x_test)

import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

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'))

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

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

y_pred=classifier.predict([[129,99,1,0,0,1]])

print(y_pred)

classification.save('flight.h5')

y_pred=classification.predict(x_test)

print(y_pred)

y_pred=(y_pred>0.5)
print(y_pred)

def predict_exist(sample_value):
  sample_value=np.array(sample_value)
  sample_value=sample_value.reshape(1,-1)
  sample_value=sc.transform(sample_value)
  return classifier.predict(sample_value)
```

```
test=classification.predict([[1,1,121.000000,36.0,0,0]])
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',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:
    kfold=model_selection.KFold(n_splits=2,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)

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

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_predict)
print(cm)

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

print(desacc)

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,decisiontree)

print(cm)

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))

from sklearn.metrics import confusion_matrix
```

```
cm=confusion_matrix(y_test,y_pred)
print(cm)

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]
}

RCV=RandomizedSearchCV(estimator=rfc,param_distributions=parameters,cv=10,n_iter=4)

print(RCV.fit(x_train,y_train))

bt_params=RCV.best_params_
bt_score=RCV.best_score_

print(bt_params)

print(bt_score)

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

y_predict_rfc=RCV.predict(x_test)
```

```
RFC=accuracy_score(y_test,y_predict_rfc)
print(RFC)

import pickle
pickle.dump(RCV,open('flight.pkl','wb'))
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__,template_folder='template')

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

@app.route('/prediction',methods=['POST'])
def predict():
  name=request.form['name']
  month=request.form['month']
  dayofmonth=request.form['dayofmonth']
  dayofweek=request.form['dayofweek']
  origin=request.form['origin']
  if (origin=='map'):
    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.form['destination']
  if (destination=='map'):
    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['arrtime']
  actdept=request.form['actdept']
  dep15=int(dept)-int(actdept)

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

```
  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()
```