**Developing a Flight Delay Prediction Model using Machine Learning**

**TEAM LEADER** : SWATHI K

**TEAM MEMBER 1** : VINOTHINI S

**TEAM MEMBER 2** : ABITHA A

**TEAM MEMBER 3** : SUMITHRA S

**FLASK CODING**

from flask import Flask, request, jsonify, render\_template, url\_for , request

import pickle

from sklearn.preprocessing import LabelEncoder

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

import pandas as pd

# Import dataset

df = pd.read\_csv('Data/Processed\_data15.csv')

# Label Encoding

le\_carrier = LabelEncoder()

df['carrier'] = le\_carrier.fit\_transform(df['carrier'])

le\_dest = LabelEncoder()

df['dest'] = le\_dest.fit\_transform(df['dest'])

le\_origin = LabelEncoder()

df['origin'] = le\_origin.fit\_transform(df['origin'])

# Converting Pandas DataFrame into a Numpy array

X = df.iloc[:, 0:6].values # from column(years) to column(distance)

y = df['delayed']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.25,random\_state=61) # 75% training and 25% test

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

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

@app.route('/')

def home():

    return render\_template('index.html')

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

def predict():

    year = request.form['year']

    month = request.form['month']

    day = request.form['day']

    carrier = request.form['carrier']

    origin = request.form['origin']

    dest = request.form['dest']

    year = int(year)

    month = int(month)

    day = int(day)

    carrier = str(carrier)

    origin = str(origin)

    dest = str(dest)

    if year >= 2013:

        x1 = [year,month,day]

        x2 = [carrier, origin, dest]

        x1.extend(x2)

        df1 = pd.DataFrame(data = [x1], columns = ['year', 'month', 'date', 'carrier', 'origin', 'dest'])

        df1['carrier'] = le\_carrier.transform(df1['carrier'])

        df1['origin'] = le\_origin.transform(df1['origin'])

        df1['dest'] = le\_dest.transform(df1['dest'])

        x = df1.iloc[:, :6].values

        ans = model.predict(x)

        output = ans

    return render\_template('index.html', prediction\_text=output)

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

    app.run(debug=False)

# For mac, make 'app.run(debug=True)'