**Forecast Commuters Inflow For Airline Industry**

**Using Prophet Model With IBM Cloud**

**INDEX:**

1. Introduction

a. Overview

b. Purpose

2. Literature Survey

a. Existing problem

b. Proposed solution

3. Theoretical Analysis

a. Block diagram

b. Hardware / Software designing

4. Experimental Investigations

5. Flowchart

6. Result

7. Advantages & Disadvantages

8. Applications

9. Conclusion

10. Future Scope

11. Bibliography

12. Appendix

a. Source code

b. UI output Screenshot.

**1.Introduction:**

Air passenger traffic forecast is of great importance for airlines and civil aviation authorities. For airlines, accurate forecasts play an increasingly important role in revenue management. It helps to reduce the airlines’ risk by objectively evaluating the demand of the air transportation business. For civil aviation authorities, air passenger traffic forecast provides a concrete basis for planning decisions in air transport infrastructure. The main objective of this project is to build a prophet time series model that forecasts the passenger traffic for a given date

**a.Overview:**

1.Data Collection

2.Preprocessing or clean the data

3.Train the model

4.Test the model

5.Save the model

6.Build a web application using flask

**b.Purpose:**

We know, that passenger numbers appears to be highest sometimes after which we see dip in numbers due to implicit seasonality and trend in the context of air passenger traffic.

This leads to raise a questions like-

What capacity aircraft should they use?

When should they fly?

How many air hostesses and pilots do they need?

How much food should they stock in their inventory?

**2. Literature Survey:**

**a.Existing system:**

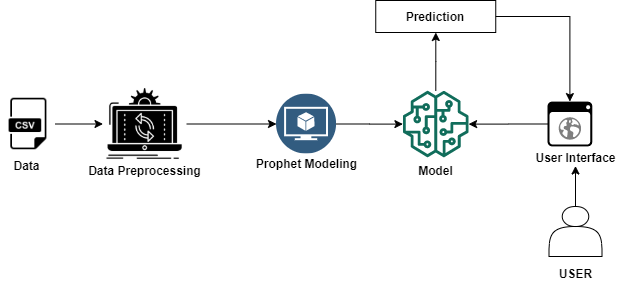
In existing system, for air passengers traffic forcast there are time series analysis with python and ARIMA model for time series forecasting may not be accurate in the case of seasonalilty and holiday effects

**b.Proposed system:**

# In this paper,we use Time Series Forecasting with Prophet.Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data.

**3.Theoretical Analysis:**

**a. Block diagram:**



**b.Software designing:**

In order to develop this project we need to install the following software/packages:

**Anaconda Navigator :**

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with great tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code.

**Python:**

* Python is a propular programing language
* It is used for
* Web development
* software development
* system scripting

**HTML AND CSS:**

HTML is used to create the actual content of the page,such as written text

CSS is responsible for the design or style of the website, including the layout,visual effects and backgruound color

**Packages:**

To complete this project, we need to install the following packages:

1.Numpy: This package is used to perform numerical computations.

2.Pandas: Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures, and data analysis tools.

3.Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

4.Scikit-learn: This is a machine learning library for the Python programming language.

5.Flask: Flask is a lightweight WSGI web application framework.

6.Requests: Allows you to send HTTP requests Using Python

7.Pystan: Pystan is a python interface to Stan,a package for Bayesian interface

8.FBProphet: FBProphet uses time as a regressor and tries to fit several linear and nonlinear function of time as components.

Install the packages in the anaconda environment.

Open anaconda prompt as administrator

* Type “pip install numpy” and click enter.
* Type “pip install pandas” and click enter.
* Type “pip install matplotlib” and click enter.
* Type “pip install scikit-learn” and click enter.
* Type “pip install Flask” and click enter.
* Type “pip install requests” and click enter.
* Type “pip install pystan” and click enter.
* Type “conda install -c conda-forge fbprophet” and click enter.

**4. Experimental Investigations :**

**Time Series Forecast:**

Time series forecasting is a technique for the prediction of events through a sequence of time. It predicts future events by analyzing the trends of the past, on the assumption that future trends will hold similar to historical trends.

**Prophet:**

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

**5. Flowchart :**

We will go through all the steps mentioned below to complete the project-

* Installation of Pre-requisites.
  + Installation of Anaconda IDE / Anaconda Navigator.
  + Installation of Python packages.
* Data Collection.
  + Create or Collect the dataset.
* Data Pre-processing.
  + Importing of Libraries.
  + Importing of Dataset & Visualisation.
* Model Building.
  + Fitting the prophet library.
  + Cross validation of the model.
  + Evaluation of the model.
  + Save the model.
  + Application Development.

**6. Result :**

We will be building a Web application where

* The user selects the date from User Interface(UI)
* The passenger traffic for the selected date is analysed by the model
* The count of passengers for the selected date is displayed on UI

**7. Advantages & Disadvantages :**

**Advantages:**

These forecasts will be a key element in the airport planning process. They are used for determining future airport requirements, analyzing alternative development plans, assessing the possible environmental effects of proposed plans, and determining the economic implications of future growth and development.

**Disadvantages:**

Error terms is what is not explained by the model.

**8. Applications:**

* Airlines
* Civil aviation authorities.

**9. Conclusion:**

An airline company has the data of the number of passengers that have travelled with them on a particular route for the past few years. Using this data,a prophet time series model that forecasts the passenger traffic for a given date

**10. Future Scope :**

For airlines, accurate forecasts play an increasingly important role in revenue management. It helps to reduce the airlines’ risk by objectively evaluating the demand of the air transportation business. For civil aviation authorities, air passenger traffic forecast provides a concrete basis for planning decisions in air transport infrastructure.

**11. Bibliography:**

We used dataset from Kaggle website to train the model-

https://www.kaggle.com/rakannimer/air-passenger-prediction

We use References of work in website-

https://towardsdatascience.com/

we also use Reference videos in youtube-

<https://www.youtube.com/watch?v=wGUV_XqchbE&t=13s>

https://www.youtube.com/watch?time\_continue=9&v=lj4I\_CvBnt0&feature=emb\_logo

**12. Appendix:**

**a. Source code :**

import numpy as np

import pandas as pd

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

import pickle

import os

from gevent.pywsgi import WSGIServer

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

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

@app.route('/')

def home():

return render\_template('home.html')

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

def y\_predict():

if request.method == "POST":

ds = request.form["Date"]

a={"ds":[ds]}

ds=pd.DataFrame(a)

prediction = model.predict(ds)

print(prediction)

output=round(prediction.iloc[0,15])

print(output)

return render\_template('home.html',prediction\_text="Commuters Inflow on selected date is. {} thousands".format(output))

return render\_template("home.html")

port = os.getenv('VCAP\_APP\_PORT','8080')

if \_\_name\_\_ == "\_\_main\_\_":

app.secret\_key = os.urandom(12)

app.run(debug=True, host='0.0.0.0',)

**b. UI output Screenshot:**



Select the date you would like to predict and click on submit.

The output prediction will be like-

