**PROFESSIONAL TRAINING REPORT**

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

**N.SAI KRISHNA REDDY**

**REG. NO. 39110669**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE ANDTECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,CHENNAI – 600119, TAMILNADU**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  **[www.sathyabama.ac.in](http://www.sathyabama.ac.in)** |  |

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING PROFESSIONAL TRAINING REPORT

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **Ms.Mecnil Jenifer., (Reg. No: 39110669)** who carried out the project entitled “Forecast Commuters Inflow For Airline Industry Using Prophet Model” under my supervision from June 2022 to aipral 2021.

Internal Guide

**Ms.Mecnil Jenifer.,**

**Head of the Department**

**Dr. L. Lakshmanan, M.E., Ph.D.,**

Submitted for Viva voce Examination held on

**InternalExaminer**

**DECLARATION**

I, **N.SAI KRISHNA REDDY** hereby declare that the project report entitled **Analytics in cloud and implementation of cloud based on cloud-based analytics infrastructure** done by me under the guidance of **Ms.MECNIL JENIFER.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

**PLACE: SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. C.HEMALATHA.,M.E., Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

|  |  |  |
| --- | --- | --- |
| **CHAPTER No.** | **TITLE** | |
|  | ABSTRACT | |
|  |  | |
|  |  | |
|
|  |  | |
| **1** | **INTRODUCTION** | |
|  |  | |
|  | Purpose of the project | |
| **2.** | **Time series** |
|  |  | |  |
|  |  | |  |
|  |  | |  |

**3. LITERATURE SURVEY**

|  |  |  |
| --- | --- | --- |
|  | Existing problem | 7 |
|  | Proposed solution | 8 |
| **4.** | **THEORITICAL ANALYSIS**  4.1 Hardware and software requirements of the project  4.2 Project Flow  The Prophet Forecasting Model | **9** |
|  |
|  |
|  |
| **5** | **ADVANTAGES&DISADVANTAGES**  APPLICATIONS  **Conclusion** |
|  | **REFERENCES** | **28** |
|  | **APPENDIX** | 14 |
|  | A.SCREENSHOTS | 17  18  20  24  26  28 |
|  |  |  |
|  |  |  |

**ABSTRACT**

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

**INTRODUCTION**

Though we have sophisticated machine learning models to predict/ forecast a time sensitive / dynamic datasets, since the correlations between variables is time sensitive and the model has to train design algorithms based upon data relation form consecutive timestamp prophet model has an upper edge to handle time series datasets as it has a functionality of auto regression which most of the machine learning algorithms lack of.

**Purpose of the project**

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, they want to see if they can forecast the number of passengers for the next twelve months.

Making this forecast could be quite beneficial to the company as it would help them take some crucial decisions 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?

1. **Quantity:** Number of passengers
2. **Granularity:** Flights from city A to city B; i.e., flights for a particular route
3. **Frequency:** Monthly
4. **Horizon:** 1 year (12 months)

By the end of this project:

1. We will be able to know the fundamental concepts of time series forecasting.
2. Working with Prophet library
3. Flask Application Development.

**Time Series:**

A time series is a collection of numerical data points in successive order. A time series tracks the movement of the chosen data points, such as stock's price, over a specified period of time with data points recorded at regular intervals. An example of time series data is monthly electricity bill collected in chronological order over a year. This will be oneyear monthly electricity bill time series. A time series data collected over same variable is univariate time series and a time series data collected over more than one variable is multivariate time series.

In general time series can be decomposed into 3 components:

1. Trend: A long term increase or decrease in data, it does not have to be linear.

2. Seasonal: It’s a time series effected by seasonal factors of time like a month in a year.

3. Cyclic: When data exhibits rises and falls that are not a fixed frequency.

4. Noise: An optional variability in the observation that cannot be explained by the model

**LITERATURE SURVEY**

**Existing problem**

We’ll be working with the Box and Jenkins (1976) Airline Passengers dataset, which contains time series data on the monthly number of airline passengers between 1949 and 1960.

**Proposed solution**

Time Series Forecasting with Prophet

Here we are using prophet model for forcasting the data

What is 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.

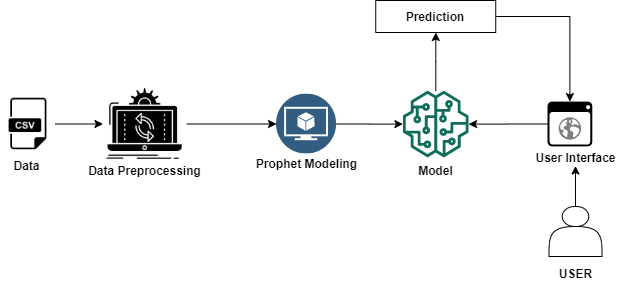
*Prophet is open source software released by Facebook’s Core Data Science team. It is available for download on CRAN and PyPI.*

* So, prophet is the facebooks’ open source tool for making time series predictions.
* Prophet decomposes time series data into trend, seasonality and holiday effect.
* **Trend** models non periodic changes in the time series data.
* **Seasonality** is caused due to the periodic changes like daily, weekly, or yearly seasonality.
* **Holiday effect** which occur on irregular schedules over a day or a period of days.
* **Error terms** is what is not explained by the model.

**THEORITICAL ANALYSIS**

**Hardware and software requirements of the project**

**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. 

For this project, we will be using **Jupyter** notebook and **Spyder**

To make a responsive python script you must require the following packages

**Requests**: Allows you to send HTTP requests using Python.

**Flask:** Web framework used for building Web applications.

**Project Flow**

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

To accomplish this, complete all the milestones & activities listed below.

* 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.

**ADVANTAGES & DISADVANTAGES**

**Advantages of Prophet**

Prophet has several advantages associated with it. These are given below:-

* **1. Accurate and fast** - Prophet is accurate and fast. It is used in many applications across Facebook for producing reliable forecasts for planning and goal setting.
* **2. Fully automatic** - Prophet is fully automatic. We will get a reasonable forecast on messy data with no manual effort.
* **3. Tunable forecasts** - Prophet produces adjustable forecasts. It includes many possibilities for users to tweak and adjust forecasts. We can use human-interpretable parameters to improve the forecast by adding our domain knowledge.
* **4. Available in R or Python** - We can implement the Prophet procedure in R or Python.
* **5. Handles seasonal variations well** - Prophet accommodates seasonality with multiple periods.
* **6. Robust to outliers** - It is robust to outliers. It handles outliers by removing them.
* **7. Robust to missing data** - Prophet is resilient to missing data.

Understanding time based patterns is critical for any business. Questions like how much inventory to maintain, how much footfall do you expect in your store to how many people will travel by an airline – all of these are important time series problems to solve.

This is why time series forecasting is one of the must-know techniques for any data scientist. From predicting the weather to the sales of a product, it is integrated into the data science ecosystem and that makes it a mandatory addition to a data scientist’s skillset.

If you are a beginner, time series also provides a good way to start working on real life projects. You can relate to time series very easily and they help you enter the larger world of machine learning.

Prophet is an open source library published by Facebook that is based on **decomposable (trend+seasonality+holidays) models**. It provides us with the ability to make time series predictions with good accuracy using simple intuitive parameters and has support for including impact of custom seasonality and holidays!

In this article, we shall cover some background on how Prophet fills the existing gaps in generating fast reliable forecasts followed by a demonstration using Python. The final results will surprise you!

**Table of Contents**

1. What’s new in Prophet?
2. The Prophet Forecasting Model
   * Trend
     + Saturating growth
     + Changepoints
   * Seasonality
   * Holidays and events
3. Prophet in action (using Python & R)
   * Trend Parameters
   * Seasonality and Holiday Parameters
   * Predicting passsenger traffic using Prophet

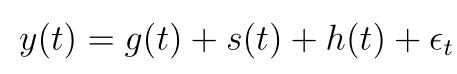
**What’s new in Prophet?**

When a forecasting model doesn’t run as planned, we want to be able to tune the parameters of the method with regards to the specific problem at hand. Tuning these methods requires a thorough understanding of how the underlying time series models work. The first input parameters to automated ARIMA, for instance, are the maximum orders of the differencing, the auto-regressive components, and the moving average components. A typical analyst will not know how to adjust these orders to avoid the behaviour and this is the type of expertise that is hard to acquire and scale.

The Prophet package provides intuitive parameters which are easy to tune. Even someone who lacks expertise in forecasting models can use this to make meaningful predictions for a variety of problems in a business scenario.

**The Prophet Forecasting Model**

We use a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:



* **g(t)**: piecewise linear or logistic growth curve for modelling non-periodic changes in time series
* **s(t)**: periodic changes (e.g. weekly/yearly seasonality)
* **h(t)**: effects of holidays (user provided) with irregular schedules
* **εt**: error term accounts for any unusual changes not accommodated by the model

Using time as a regressor, Prophet is trying to fit several linear and non linear functions of time as components. Modeling seasonality as an additive component is the same approach taken by exponential smoothing in  . We are, in effect, framing the forecasting problem as a curve-fitting exercise rather than looking explicitly at the time based dependence of each observation within a time series.

* A major criticism against Prophet is that **its underlying assumptions are simplistic and weak**. In 2017, Facebook released Prophet, an open-source forecasting tool in Python and R. The demand for high-quality forecasts often outpaces the analysts producing them.

**APPLICATIONS**

This activity lets you create a Flask Web application where the user can select the specific date to forecast the passangers count on the selected date.  To accomplish the task you should build the required HTML pages and styling sheets as well as backend scripting files

please refer to the above-mentioned files in the link provided in the project structure milestone

**Conclusion**

* In this tutorial, we described how to use the Prophet library to perform time series forecasting in Python.
* We have been using out-of-the box parameters, but Prophet enables us to specify many more arguments.
* In particular, Prophet provides the functionality to bring your own knowledge about time series to the table.

**References**

The concepts and ideas in this notebook are tgaken from the following websites-

1.<https://facebook.github.io/prophet/>

2.<https://facebook.github.io/prophet/docs/quick_start.html>

3.<https://peerj.com/preprints/3190.pdf>

4.<https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-prophet-in-python-3>

**APPENDIX**

Source Code:-

#### Importing the libraries

import numpy as np

#import pandas

import pandas as pd

#import visualization library

import matplotlib.pyplot as plt

#### importing the dataset

#read the dataset

data=pd.read\_csv('AirPassengers.csv')

#check the first 5 rows of data

data.head()

#checking the shape of data

data.shape

#checking the null values

data.isnull().any()

#rename the #passengers column name

data.rename(columns={"#Passengers": "Passengers"},inplace=True)

data.head()

#### Data Visualization

#We can plot the data easily in Pandas by calling the plot() function on the DataFrame.

import matplotlib.pyplot as plt

data.plot()

plt.show()

import datetime

data['ds']=pd.to\_datetime(data['ds'], format='%Y-%m')

data.hedataad()

data.shape

#### Forecast airline passengers traffic flow With PropheT

#import datetime conversion

from pandas import to\_datetime

#prepare expected column names

data.columns = ['ds', 'y']

data['ds']= to\_datetime(data['ds'])

data.head()

#visualizing the ds column

#configure the figure size

plt.figure(figsize=(12,6))

plt.plot(data.set\_index(['ds']))

#### Model Building

#fit prophet model on the dataset

#import Prophet library from fbprophet

from fbprophet import Prophet

import logging

logging.getLogger('fbprophet').setLevel(logging.WARNING)

# define the model

model = Prophet( )

# fit the model

model.fit(data)

#### Prediction

future\_prediction = model.make\_future\_dataframe(periods=365,freq="D")

future\_prediction.tail()

#### Obtaining the forecastes

forecast=model.predict(future\_prediction)

forecast.head()

#### summarize the forecast

# summarize the forecast

print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].head())

#### plot forecast

model.plot\_components(forecast)

#### Cross Validation

from fbprophet.diagnostics import cross\_validation

cv = cross\_validation(model,initial = '530 days',period='180 days',horizon = '365 days')

cv

#### Evaluation of Model

from fbprophet.diagnostics import performance\_metrics

pm=performance\_metrics(cv)

pm.head()

#### Visualizing the performance Metrics

from fbprophet.plot import plot\_cross\_validation\_metric

fig = plot\_cross\_validation\_metric(cv,metric='rmse')

#### Save the Model

#import pickle

import pickle

pickle.dump(model,open('AirPassengers.pkl','wb'))