# TIME SERIES ANALYSIS AND SALES FORECASTING FOR AUTOMATIVE USING IBM CLOUD SERVICES

A MINI PROJECT REPORT

Submitted to

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

*In partial fulfillment of the requirements for the award of the degree of*

## BACHELOR OF TECHNOLOGY

In

## COMPUTER SCIENCE AND ENGINEERING

BY

| | |
|---|---|
| **B. DIVYA SRI** | **(18UK1A05C4)** |
| **K. ADITHYA** | **(18UK1A0582)** |
| **K. ANJALI** | **(18UK1A05E7)** |
| **SALAHUDDIN** | **(18UK1A05B1)** |

Under the esteemed guidance of

## Mr.B NIRANJAN

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VAAGDEVI ENGINEERING COLLEGE**

Affiliated to JNTU, HYDERABAD

BOLLIKUNTA, WARANGAL-506005.

2018-2022

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTU, HYDERABAD

BOLLIKUNTA, WARANGAL-506005.

2018-2022



# CERTIFICATE

This is to certify that the Mini Project Report entitled "**TIME SERIES ANALYSIS AND SALES FORECASTING FOR AUTOMOTIVE USING IBM CLOUD SERVICES**" is being submitted by **B. DIVYA SRI (I8UK1A05C4), K.ADITHYA(18UK1A0582), K.ANJALI(18UK1A05E7) ,S. SALAHUDDIN(18UK1A05B1),** in partial fulfillment of the requirement for the award of the Degree in Bachelor of Technology in computer science and engineering during the engineering during the academic year 2018-2022.

**Mr. NIRANJAN**                                                    **Dr. R. NAVEEN KUMAR**

Project Guide                                                          Head of The Department

**EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this mini project in the institute.

We extend our heartfelt thanks to **Dr. R. Naveen Kumar,** Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the mini project.

We express heartfelt thanks to the Mini Project Coordinator, **Ms. G. Arunakranthi,** Assistant Professor, Department of CSE for her constant support and giving necessary guidance for completion of this mini project.

We express heartfelt thanks to the guide, **Mr. Niranjan,** Assistant Professor, Department of CSE for her constant support and giving necessary guidance for completion of this mini project.

We express our sincere thanks and gratitude to **The Smart Bridge,** for providing internship.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experiencing throughout thesis.

# ABSTRACT

Forecasting or Predicting the sale value helps the investors to invest in such a time where profits can be maximum. This project provides guidance to individuals who are willing to invest or buy a car and help them in knowing the price of a day using the prophet library. It is built on the monthly sales data from 1960 - 1968. Time series analysis is made on the data for accurate predictions.

# CONTENTS

# INTRODUCTION

## 1.1 INTRODUCTION

Forecasting or Predicting the sale value helps the investors to invest in such a time where profits can be maximum. This project provides guidance to individuals who are willing to invest or buy a car and help them in knowing the price of a day using the prophet library. It is built on the monthly sales data from 1960 - 1968. Time series analysis is made on the data for accurate predictions.Time series forecasting is the process of analyzing time series data using statistics and modeling to make the predictions and inform strategic decision making.

## 1.2 OVERVIEW

Sales forecasting is the process of estimating future sales. Accurate sales forecasts enable companies to make informed business decisions and predict short-term and long-term performance. Companies can base their forecasts on past sales data, industry-wide comparisons, and economic trends.

## 1.3 PURPOSE

Sales forecasting is the process of estimating future sales. Accurate sales forecasts enable companies to make informed business decisions and predict short-term and long-term performance.Sales forecasting gives insight into how a company should manage its workforce, cash flow, and resources.

# PROJECT OBJECTIVES

**By the end of this project:**

- Know fundamental concepts and techniques of the time series forecasting.

- Gain a broad understanding of time series data.

- Gain knowledge on fbprophet library.

- You will be able to know how to find the accuracy of the model

- You will be able to build web applications using the Flask Framework

# LITERATURESURVEY

## EXISTING PROBLEM

Sales people not having sufficient knowledge of the details of specific deals, and/or (nearly as bad) failing to enter that information into the sales forecasting system .

A lack of personal accountability on the part of individual sales people as to their responsibilities for accurate sales forecasting

Sales people displaying over-confident, conservative or sandbagging behaviours in their personal forecasting

A general inability to understand or calculate the realistic probabilities and closing dates for current deals

## PROPOSED SOLUTION

Most of the sales people face many problems in prediction of sales so, we are with a solution in which the Forecasting or Predicting the sale value helps the investors to invest in such a time where profits can be maximum. This application helps you in predicting the sale value of a day. It is built on the monthly sales data from 1960 - 1968. Time series analysis is made on the data for accurate predictions. Which is achieved by a Flask application and used packages like fbprophet and pandas etc .

# THEORITICAL ANALYSIS

BLOCK DIAGRAM
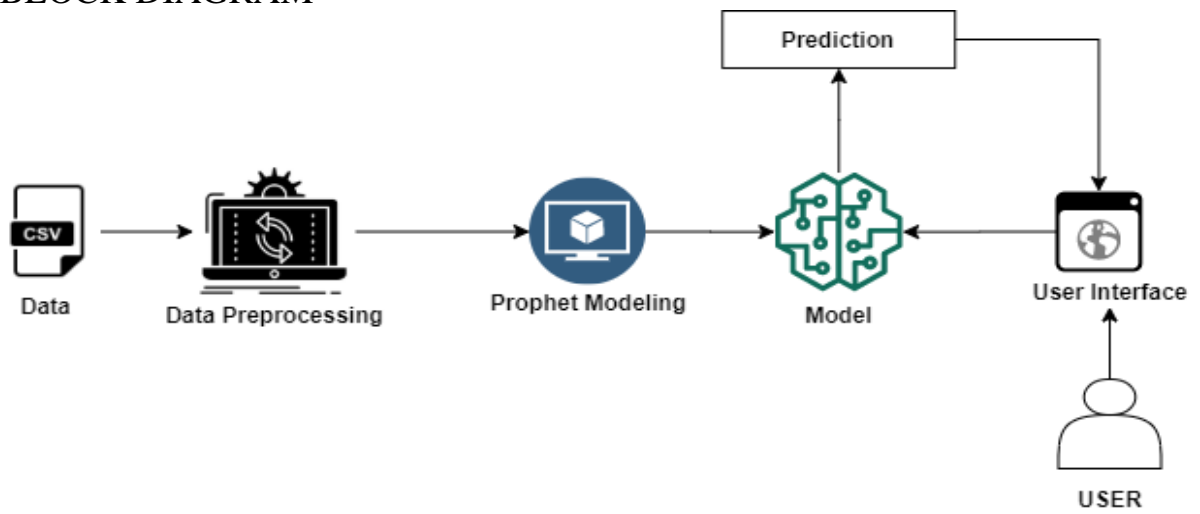


HARDWARE / SOFTWARE DESIGNING

- Laptop
- Vscode , jupyter Notebook ,IBM watson

# EXPERIMENTAL INVESTIGATIONS

PROJECT FLOW

User interacts with the UI (User Interface) to give the date as input.

The date is analyzed by the model which is integrated Once the model analyses the uploaded date, the prediction is showcased on the UI.

To accomplish this, we have to complete all the activities and tasks listed below

1. Installation of Prerequisites.
   - Installation of Anaconda IDE / Anaconda Navigator.
   - Installation of Python packages.
2. Data Collection.
   - Create or Collect the dataset.
3. Data Pre-processing.
   - Importing of Libraries.
   - Importing of Dataset &Visualisation.
4. Model Building.
   - Fitting the prophet library.
   - Evaluation of the model.
   - Save the model.

5. Application Building

PREREQUISITES

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

To build Machine learning models we require the following packages

Numpy:

It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations

FbProphet:

FbProphet is a powerful time-series analysis package released by Core Data Science Team at Facebook. It is a simple and easy to go package for performing time-series analytics and forecasting at scale. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

## INSTALLATION OF PACKAGES

1. Open anaconda prompt.
2. Type "pip install requests" and click enter.
3. Type "pip install Flask" and click enter.
4. Type "conda install -c  conda-forge fbprophet" and press enter

## PROJECT STRUCTURE

| Name | Size | Type |
|---|---|---|
| ▼ 📂 Flask | | File Folder |
|    ▼ 📂 templates | | File Folder |
|       📄 predict.html | 3 KB | html File |
|    📄 app.py | 802 bytes | py File |
|    📄 sales.sav | 1.1 MB | sav File |
| 📄 Car Sales Testing.ipynb | 1 KB | ipynb File |
| 📄 Car Sales Training.ipynb | 97 KB | ipynb File |
| 📄 monthly-car-sales.csv | 1 KB | csv File |
| 📄 sales.sav | 1.1 MB | sav File |

In order to proceed with this milestone, arrange all your project files in the below format.

- All the above files will be used to develop a flask application.
- In the templates, you will store all the rendering files and HTML pages.
- The sales.sav is the saved model file
- app.py is the python script (flask file)

## DATA COLLECTION

ML depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

## COLLECT THE DATASET

You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository etc.Please refer to the link given below to download the data set and to know about the dataset It contains weather forecast from 1960 to 1968.

This dataset contains two columns many columns. But we are interested in two columns for time series analysis

- Month
- Sales

## DATA PREPROCESSING

As we have collected data, let's start training, but before training the machine make sure the data is in proper structure. So first let's clean the collected data initially.

Data Pre-processing includes the following main tasks:

- Importing the required libraries
- Importing the dataset
- Analyze the data
- Resampling the dataset
- Preprocessing the data
- Taking care of Missing Data

- Prophet Library naming convention
- Data visualization

Create a jupyter file and start coding it.Watch the youtube link given in the prerequisites section to know more about the usage of jupyter.

## IMPORT LIBRARIES

Import the below libraries to import the dataset, for data pre-processing etc…

```python
import pandas as pd
from matplotlib import pyplot
from pandas import to_datetime
from fbprophet import Prophet
```

## IMPORT DATASET AND VISUALIZATION

Import the dataset, using the below command

```python
# load data
df = pd.read_csv('monthly-car-sales.csv')
# summarize shape
df.shape
```

```
(108, 2)
```

- List the first five rows of the dataset using head().
- Visualize the sale trend using matplotlib.
- Convert the date from string to date using to_datetime.

```python
# prepare expected column names
df.columns = ['ds', 'y']
df['ds']= to_datetime(df['ds'])
```

- With this, you have successfully completed the milestone.

## MODEL BUILDING

Now that our data is ready for training let's train the model

Model Building Includes:

- Model Fitting
- Making Future Predictions
- Obtaining the Forecasts
- Plotting the Forecasts
- Model Evaluation
- Saving the model

## FITTING THE PROPHET LIBRARY

- Import the prophet library, create the instance and fit it to the dataset using the below commands.

```python
# define the model
model = Prophet()
# fit the model
model.fit(df)
```

Note: It will take a few minutes to fit the model.

- Making Future Predictions

The next step is to prepare our model to make future predictions. This is achieved using the Prophet.make_future_dataframe method or considering a few dates.

```
future = list()
for i in range(1, 13):
    date = '1969-%02d' % i
    print(date)
    future.append([date])
future = pd.DataFrame(future)
future.columns = ['ds']
future['ds']= to_datetime(future['ds'])
```

- Obtaining the Forecasts

We use the predict method to make future predictions. This will generate a data frame with an yhat column that will contain the predictions.

If we check the head for our forecast data frame we'll notice that it has very many columns. However, we are mainly interested in ds, yhat, yhat_lower and yhat_upper. yhat is our predicted forecast, yhat_lower is the lower bound for our predictions and yhat_upper is the upper bound for our predictions.

```
forecast=model.predict(future)
```

OBTAINING THE SUMMARY AND PLOTTING THE RESULTS

- Get the summary of the forecast using the below command.

```
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
```

|    | ds         | yhat         | yhat_lower   | yhat_upper   |
|----|------------|--------------|--------------|--------------|
| 0  | 1969-01-01 | 15376.752451 | 13729.421592 | 16995.966021 |
| 1  | 1969-02-01 | 16138.731483 | 14484.420546 | 17703.478388 |
| 2  | 1969-03-01 | 21351.103785 | 19819.830527 | 22810.013611 |
| 3  | 1969-04-01 | 23479.854771 | 21971.888265 | 24913.918607 |
| 4  | 1969-05-01 | 24992.388293 | 23548.883629 | 26625.440222 |
| 5  | 1969-06-01 | 22254.527429 | 20671.033329 | 23796.754674 |
| 6  | 1969-07-01 | 17901.835593 | 16371.452543 | 19499.790600 |
| 7  | 1969-08-01 | 15730.040830 | 14151.520766 | 17300.068331 |
| 8  | 1969-09-01 | 14235.917670 | 12717.947618 | 15794.608600 |
| 9  | 1969-10-01 | 18672.473959 | 17094.533468 | 20307.941880 |
| 10 | 1969-11-01 | 18578.551829 | 17076.341641 | 20130.639843 |
| 11 | 1969-12-01 | 16388.664407 | 14807.094795 | 17849.068471 |

Now, let's visualize the forecast using the below commands.

```
model.plot(forecast)
```
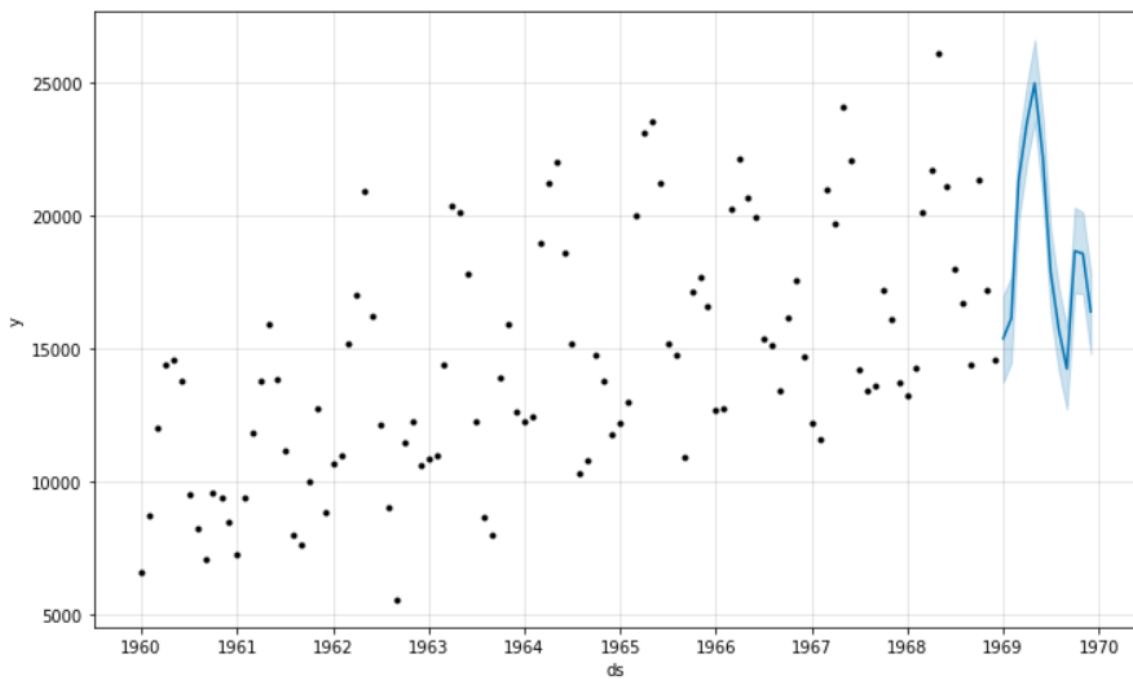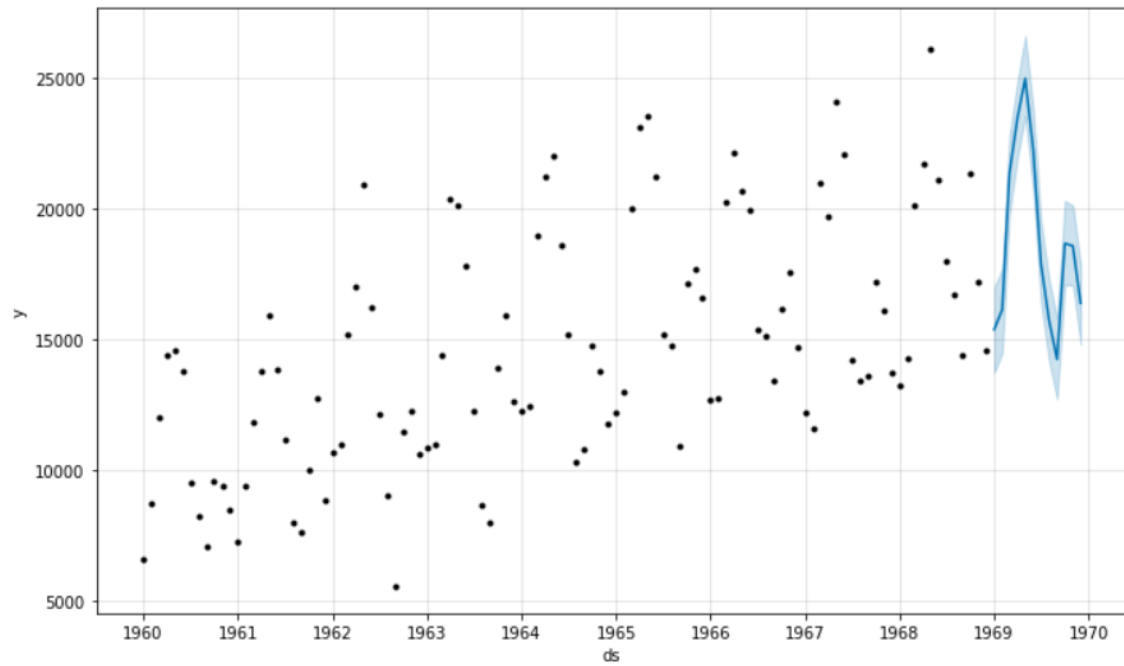
## SAVE THE MODEL

This is the final activity of this milestone, here you will be saving the model to integrate into the web application.

- Follow the commands to save your model.

```
import joblib
joblib.dump(model, "sales.sav")

['sales.sav']
```

- With this, you have successfully completed the milestone.

## APPLICATION BUILDING

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

IMPORTING OF LIBRARIES AND ROUTING THE HTMLPAGES

writing backend We will be using python for server-side scripting. Let's see the step by step process for code.

Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument Pickle library to load the model file.

- Import the following libraries

```
import joblib
import pandas as pd
from flask import Flask, request, render_template
```

- Rendering to HTML page

Here we will be using a declared constructor to route to the HTML page which we have created earlier.

In the above example, the '/' URL is bound with the predict.html function. Hence, when the home page of a web server is opened in the browser, the HTML page will be rendered. Whenever you enter the values from the HTML page the values can be retrieved using the POST Method. Load the saved model.

```
app = Flask(__name__)
model = joblib.load('sales.sav')
```

- Rendering of home page HTML.

```
@app.route('/')
def home():
    return render_template('predict.html')
```

- Routing the prediction to the home page.

```
@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('predict.html',output="The sale value on selected date is {} thousands".format(output))
    return render_template("predict.html")
```

- Calling of Main Function

- This is used to run the application in the localhost.

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

Note: Ensure that you have written the code properly without indentation errors.

## RUNNING OF FLASK APPLICATION

- Open the anaconda prompt from the start menu.
- Navigate to the folder where your app.py resides.
- Now type the "python app.py" command.
- It will show the local host where your app is running on http://127.0.0.1.5000/
- Copy that local host URL and open that URL in the browser. It does navigate you to where you can view your web page.

```
(base) F:\Projects\Car Sales Time Series\Flask>python app.py
```

- The UI will look like

- Select the date you would like to predict and click on submit.
- The output prediction will be like.



Check the predictions for the different inputs.

- With this you have successfully completed the project.

# RESULT

The output prediction will be like



# APPLICATIONS

Forecasting has applications in a wide range of fields where estimates of future conditions are useful. Not everything can be forecast reliably, if the factors that relate to what is being forecast are known and well understood and there is a significant amount of data that can be used very reliable forecasts can often be obtained. If this is not the case or if the actual outcome is affected by the forecasts, the reliability of the forecasts can be significantly lower.

# CONCLUSION

**A sales** forecast helps every business make better business decisions. It helps in overall business planning, budgeting, and risk management. ... Sales forecasting also helps businesses to estimate their costs and revenue accurately based on which they are able to predict their short-term and long-term sales.

# FUTURE SCOPE

For the sales rep, as well as the entire organization, a sales forecast aims to predict future sales and is used as the basis of planning time and resources. A good forecast should have several objectives, all directed at identifying what you will sell, when you will sell it and to whom.

# BIBILOGRAPHY

https://en.m.wikipedia.org/wiki/Forecasting

https://www.inflexion-point.com/Blog/bid/90329/Why-is-accurate-sales-forecasting-such-a-challenge

https://www.tableau.com/learn/articles/time-series-forecasting

https://smartinternz.com/Student/guided_project_info/5799#