# Time Series Analysis for Car Sales Forecasting Using Prophet with IBM Cloud

## 1 Introduction:

Timeseries analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data.
Timeseries forecasting is the use of a model to predict future values based on previously observed values. Time series analysis can be applied to real-valued, continuous data.

### 1.1 Overview:

By this Time Series Analysis, we can easily predict the sales of car. Because we are using the robust model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

### 1.2 Purpose:

The overall purpose of the study was to prove that it's possible to efficiently forecast car sales using a simple statistical model. During our research we were able to prove that the PROPHET-based approach has acceptable outcomes.

Such models can be easily implemented with various statistical software and their computational complexity is acceptable. Also, the approach has well-studied statistical properties.

Forecasting or Prediction 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.

## 2 Literature Survey:

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. Predicting the sale value helps the investors to invest in such a time where profits can be maximum.

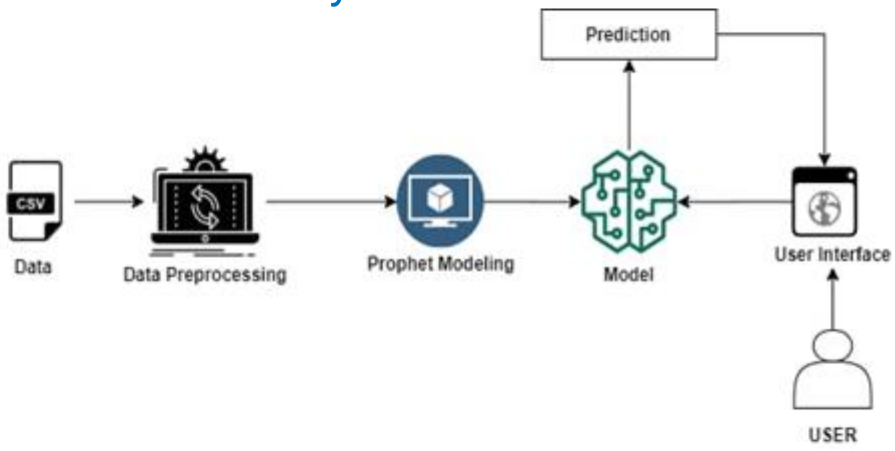
### 2.1 Existing problem:

The main problem in this time series analysis for car sales forecasting is difference to predict the sales. And it is different to investors calculate the several seasons of historical data. And how to improve the cars sales per monthly or per yearly.

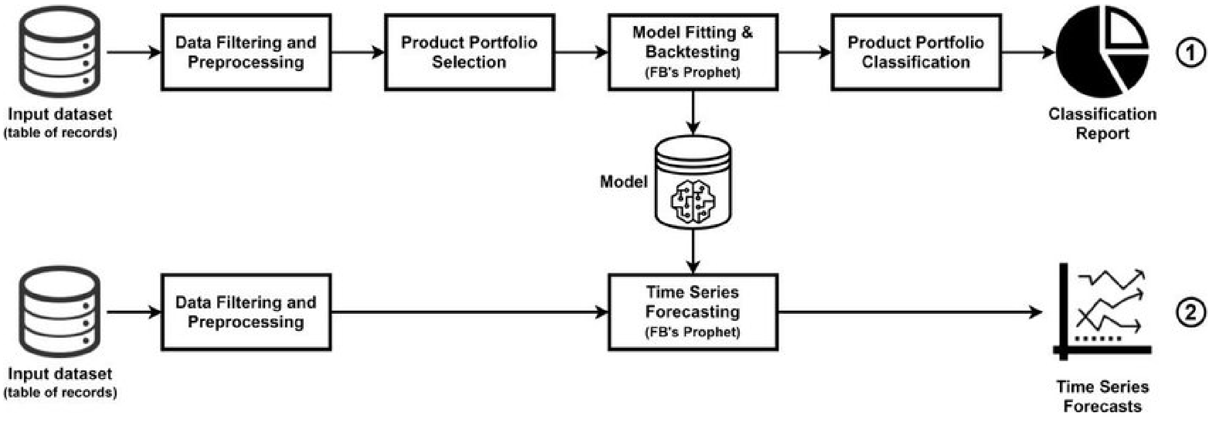
## 2.2 Proposed Solution:

To solve the above Existing problem we were using the 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. Because we are using the robust model.

# 3 Theoretical Analysis:



## 3.1 Block Diagram:



## 3.2 Hardware/Software desiging:

**Hardware Designing:**

| | | |
|---|---|---|
| Windows | - | 10 |
| Processor | - | intel core i3 or above or equivalent. |
| Hard disk | - | 20 GB |
| RAM | - | 1 GB or more |

**Software Designing:**

- 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 so very nice tools like JupyterLab, Jupyter Notebook,QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

- Consistent, simple and extensible API.
- Minimal structure - easy to achieve the result without any frills.
- It supports multiple platforms and backends.
- It is user friendly framework which runs on both CPU and GPU.
- Highly scalability of computation.

- Python 3.7 :

Python is broadly utilized universally and is a high-level programming language. It was primarily introduced for prominence on code, and its language structure enables software engineers to express ideas in fewer lines of code. Python is a programming language that gives you a chance to work rapidly and coordinate frameworks more effectively.

- Pandas:

Pandas is a Python library used for working with data sets.It has functions for analyzing, cleaning, exploring, and manipulating data.The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
Pandas allows us to analyze big data and make conclusions based on statistical theories.Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

- Matplotlib:

Matplotlib is a low level graph plotting library in python that serves as a visualization utility.Matplotlib was created by John D. Hunter.Matplotlib is open source and we can use it freely.Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

- Fbprophet:

Trend models non periodic changes in the value of the time series.Seasonality is 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.

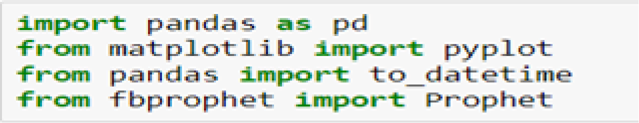
# 4 Experimental Investigations:

- First step is collect or create the dataset.
- After collecting the data set we to apply the Data Preprocessing Techniques because we want pure data without any impurities.
- Data Pre-processing includes the following main tasks:

a. Importing the required libraries
b. Importing the dataset
c. Analysis the data
d. Resampling the dataset
e. Preprocessing the data
f. Taking care of Missing Data
g. Prophet Library naming convention
h. Data visualization

- Open the jupyter notebook.

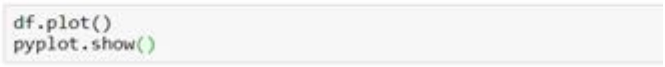
- For the Time series analysis for car sales forecasting main important libraries are pandas and by pandas we can import the datetime by using the to_datetime and matplotlib and Fbprophet.



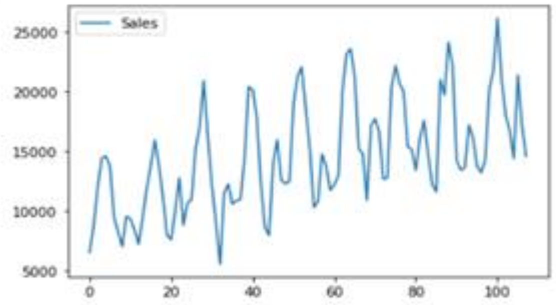
```
import pandas as pd
from matplotlib import pyplot
from pandas import to_datetime
from fbprophet import Prophet
```

- After importing the dataset which was performed under the Data Preprocessing Techniques.

- Visualize the sale trend using matplotlib

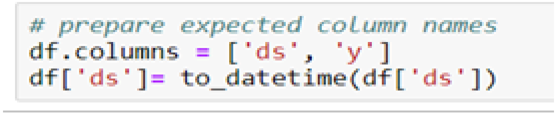


```
df.plot()
pyplot.show()
```



.

➤ Convert the date from string to date using to_datetime.



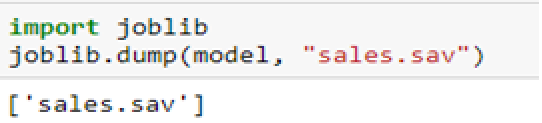
```
# prepare expected column names
df.columns = ['ds', 'y']
df['ds']= to_datetime(df['ds'])
```

➤ 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

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

➤ After this we have to save the model.

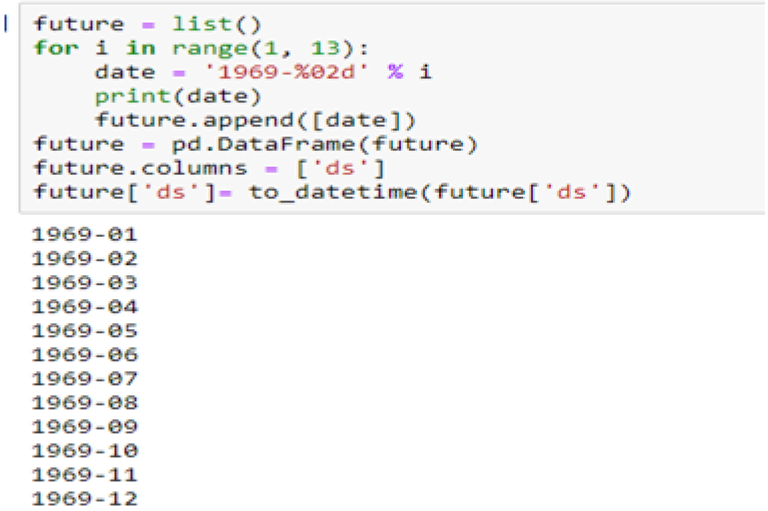


```
import joblib
joblib.dump(model, "sales.sav")
```

```
['sales.sav']
```

➤ 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.

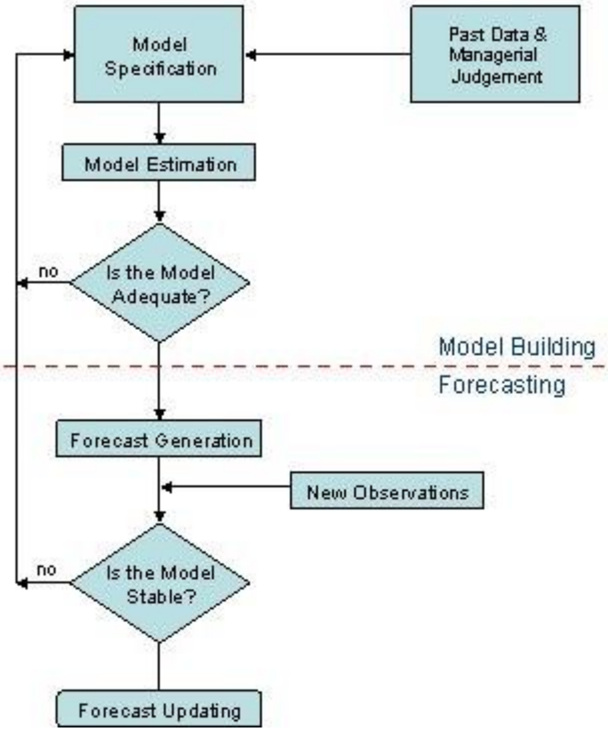
➤ We use the predict method to make future predictions. This will generate a data frame with an yhat column that will contain the predictions



```
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'])
```

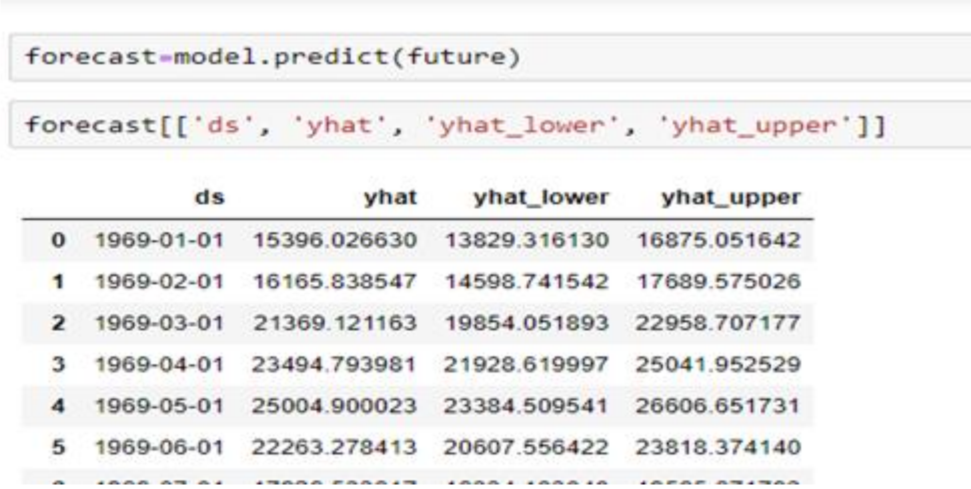
```
1969-01
1969-02
1969-03
1969-04
1969-05
1969-06
1969-07
1969-08
1969-09
1969-10
1969-11
1969-12
```

## 5 Flowchart:



# 6 Result:

➢ If we check the head for our forecast data frame. yhat is our predicted forecast, yhat_lower is the lower bound for our predictions and yhat_upper is the upper bound for our predictions

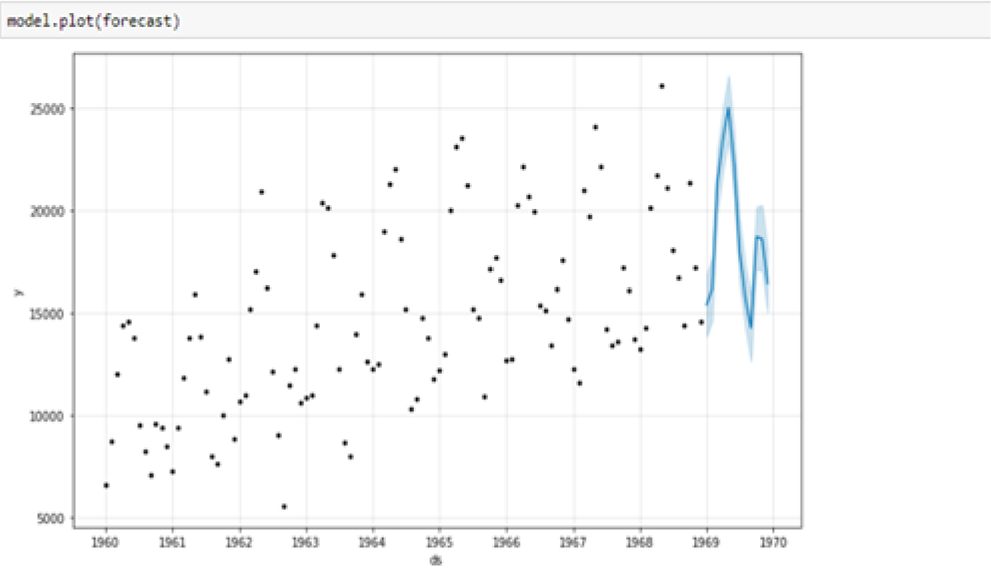


```
forecast=model.predict(future)
```

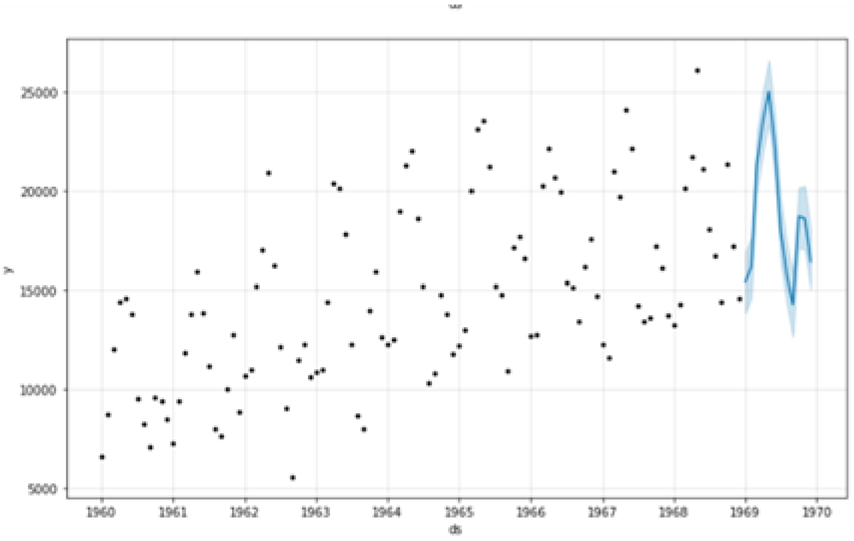
```
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
```

|   | ds | yhat | yhat_lower | yhat_upper |
|---|---|---|---|---|
| **0** | 1969-01-01 | 15396.026630 | 13829.316130 | 16875.051642 |
| **1** | 1969-02-01 | 16165.838547 | 14598.741542 | 17689.575026 |
| **2** | 1969-03-01 | 21369.121163 | 19854.051893 | 22958.707177 |
| **3** | 1969-04-01 | 23494.793981 | 21928.619997 | 25041.952529 |
| **4** | 1969-05-01 | 25004.900023 | 23384.509541 | 26606.651731 |
| **5** | 1969-06-01 | 22263.278413 | 20607.556422 | 23818.374140 |

➢ Now, let’s visualize the forecast using the below commands.



```
model.plot(forecast)
```



# 7 Advantages & Disadvantages :

## 7.1 Advantages:

- Accommodates seasonality with multiple periods
- Prophet is resilient to missing values
- Best way to handle outliers in Prophet is to remove them
- Fitting of the model is fast
- Intuitive hyper parameters which are easy to tune

## 7.2 Disadavantages:

- Doesn't catch interactions between external features, which could improve the forecasting power of a model. In our case, these variables are Promo and CompetitionOpen.
- Multiplicative models cannot be accounted for using prophet.
- Data need to be feed in pre-defined format.

# 8 Applications:

- We discussed time series analysis which has given us the understanding that time series models first recognize the trend and seasonality from the existing observations and then forecast a value based on this trend and seasonality. Such analysis is useful in various fields such as –

✓ Financial Analysis – It includes sales forecasting, inventory analysis, stock market analysis, price estimation.

✓ Weather Analysis – It includes temperature estimation, climate change, seasonal shift recognition, weather forecasting.

✓ Network Data Analysis – It includes network usage prediction, anomaly or intrusion detection, predictive maintenance.

✓ Healthcare Analysis – It includes census prediction, insurance benefits prediction, patient monitoring.

# 9 Conclusion:

➢ The conclusion of the project is:

- know fundamental concepts and techniques of 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.

# 10 Future Scope:

➢ A time series is a sequence of observations over a certain period.  Forecast time series data with the help of various statistical and machine learning models in elaborate and easy to understand way.

➢ Times series analysis we can Prediction the sales of cars for every seasonal period  not only the car sales it is also used for supermaskets,etc.

➢ But this predicted data it is helpful for investors for increasing the products and how to  attract the customers by their new ideas .

# 11 Bibliography:

1. An Introduction to Time Series Forecasting with Prophet in Exploratory
2. Forecasting Time Series With Prophet in Python - BLOCKGENI
3. Facebook's open source forecasting tool, Prophet
4. Time Series Analysis and Forecasts with Prophet | Kaggle
5. Prophet - Reviews, Pros & Cons | Companies using Prophet
6. Application of Facebook's Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data

# 12 Appendix:

## 12.1 Source code

➢ In this section, we will be building a web application that is integrated into the model we built.

➢  A UI is provided for the uses where he has uploaded the dataset.

➢ The uploaded dataset is given to the saved model and prediction is showcase on the user interface.

➢ This section has the following tasks

➢ Building HTML Pages

➢ Building server-side script

➢ Let us build the flask file ‘app.py’ which is a web framework written in python for server-side scripting.

➢ Let’s see step by step procedure for building the backend application:

➢ App starts running when the “__name__” constructor is called in main.

➢ render_template is used to return HTML file.

➢ “GET” method is used to take input from the user.

➢ “POST” method is used to display the output to the user.

➢ "Pandas" is the most popular python library that is used for **data analysis**. It provides highly optimized performance with back-end source code is purely written in C or Python.

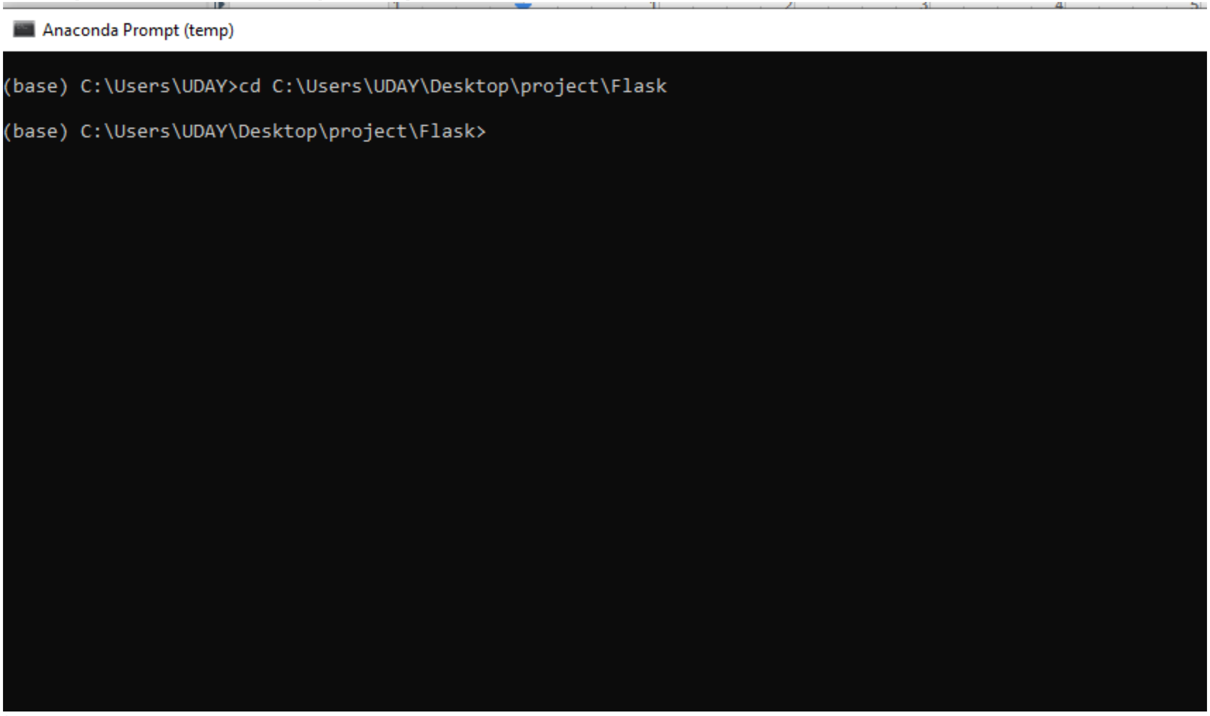
➢ Series is one dimensional (1-D) array defined in pandas that can be used to store any data type.

> ➢ Python library which facilitates parallel processing in Python.
>
> ➢ "Joblib" uses Python’s multiprocessinglibrary to run functions in parallel, but it also allows users to define their own “back end” for parallel computation.
>
> ➢ The Civis Python API client takes advantage of this to let you easily run your own code in parallel

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

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

@app.route('/')
def home():
    return render_template('predict.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('predict.html',output="The sale value on selected date is {} thousands".format(output))
    return render_template("predict.html")
port=os.getenv('VCAP_APP_PORT','8080')

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

## 13 UI output Screenshot:

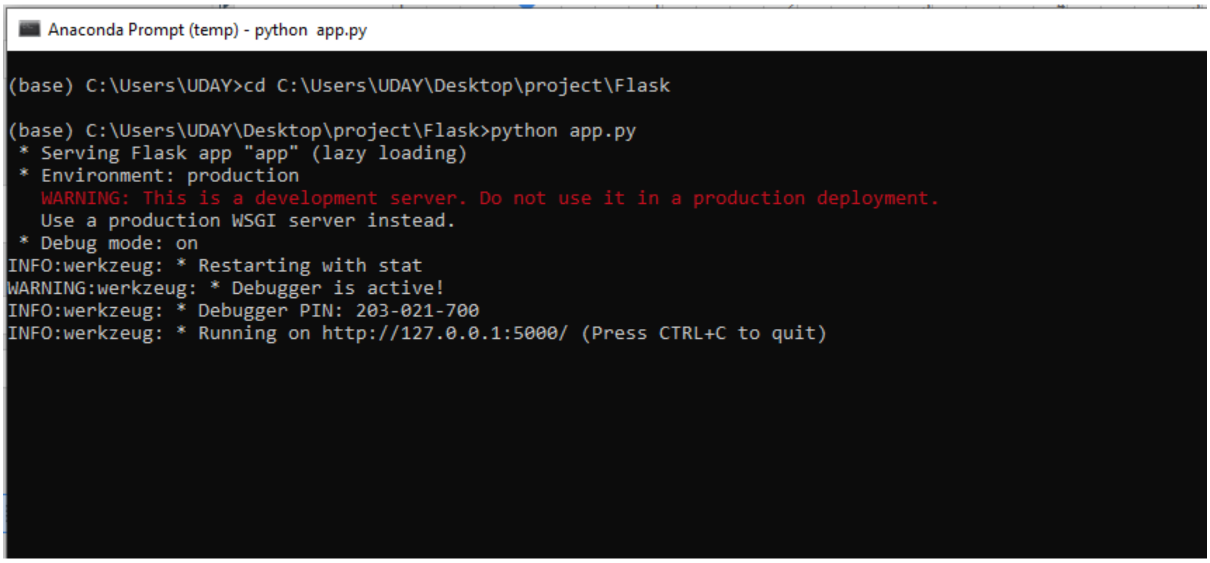
- Open anaconda prompt from the start menu



Anaconda Prompt (temp)

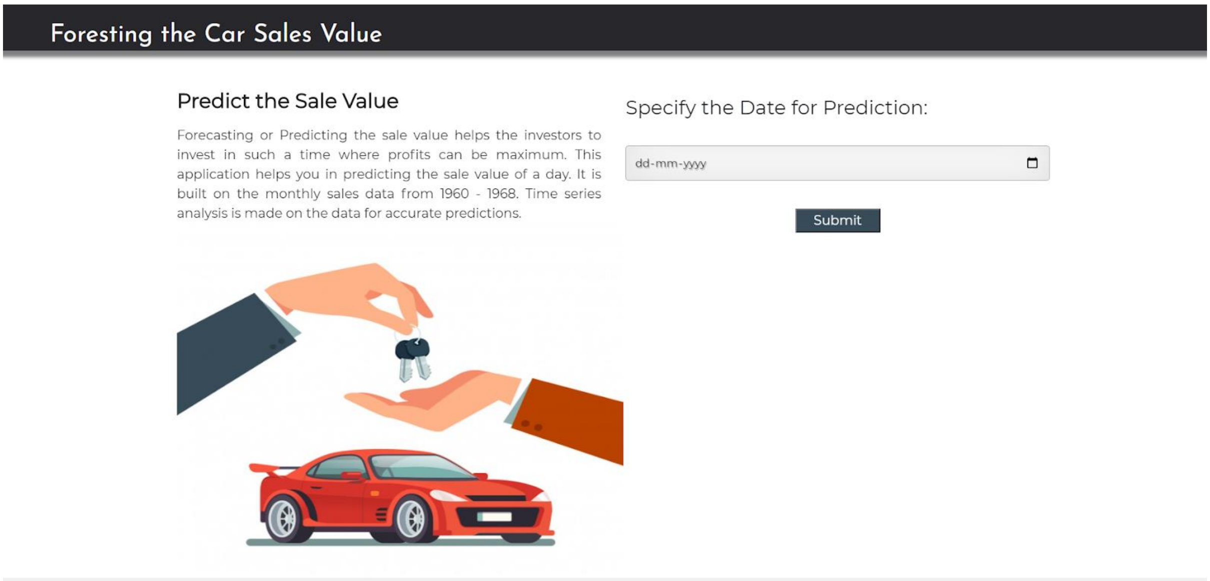
```
(base) C:\Users\UDAY>cd C:\Users\UDAY\Desktop\project\Flask

(base) C:\Users\UDAY\Desktop\project\Flask>
```

- Navigate to the folder where your python script is.
- Now type “python app.py” command



```
(base) C:\Users\UDAY>cd C:\Users\UDAY\Desktop\project\Flask

(base) C:\Users\UDAY\Desktop\project\Flask>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
INFO:werkzeug: * Restarting with stat
WARNING:werkzeug: * Debugger is active!
INFO:werkzeug: * Debugger PIN: 203-021-700
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

- After getting the URL "http://127.0.0.1:5000/" we have to copy the url and navigate to the google paste the url.
- After we views the html page



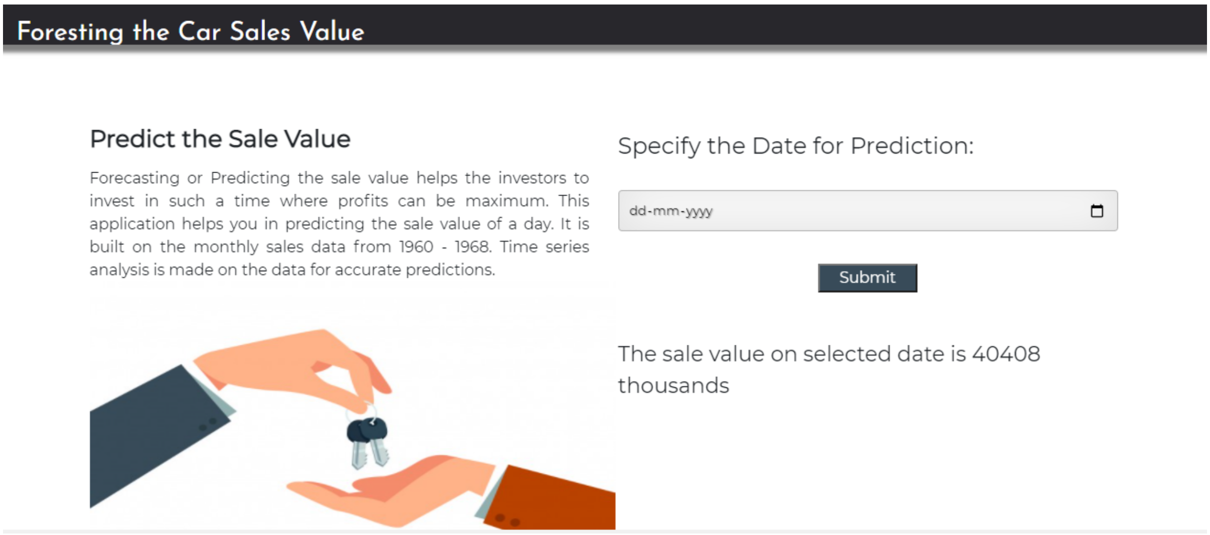
Foresting the Car Sales Value

Predict the Sale Value

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.

Specify the Date for Prediction:

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



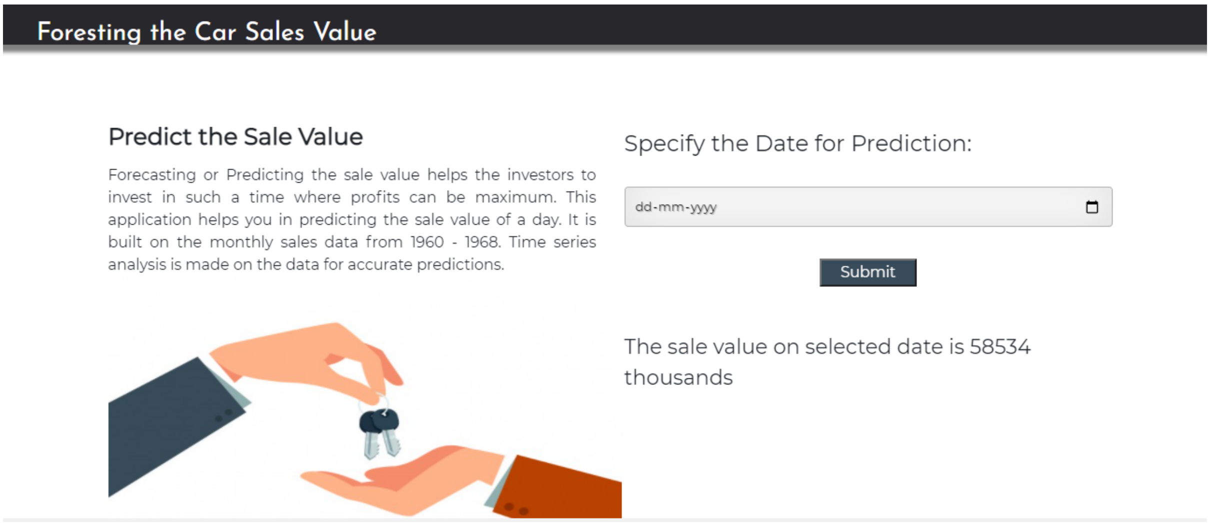
Foresting the Car Sales Value

Predict the Sale Value

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.

Specify the Date for Prediction:

The sale value on selected date is 40408 thousands



Foresting the Car Sales Value

Predict the Sale Value

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.

Specify the Date for Prediction:

The sale value on selected date is 58534 thousands