TIME SERIES ANALYSIS FOR CAR SALES PREDICTION USING PROPHET

1 INTRODUCTION

1.1 Overview

The objective of the proposed project is to develop an accurate and reliable model that can forecast future sales for cars based on historical data. The entire process starts from data collection to model evaluation. The model uses the Prophet library, developed by Facebook's core data science team offering valuable features including trend capturing and seasonality modelling which are essential for analyzing and predicting operations. The report also highlights the steps taken to train and fine-tune the model, including feature engineering and parameter optimization. The performance of the model is evaluated using various evaluation metrics, enabling a comprehensive understanding of its predictive capabilities. Ultimately, this project aims to provide insights into car sales trends and offer a reliable tool for car manufacturers and dealerships to make informed decisions and plan their strategies accordingly.

1.2 Purpose

The purpose of this project is to utilize time series analysis techniques, specifically employing the Prophet forecasting model, for predicting car sales. The project aims to address the need for accurate and reliable predictions in the automotive industry, where understanding sales patterns and forecasting future demand is crucial for effective decision-making. By leveraging historical car sales data, the project seeks to develop a robust forecasting model that can capture underlying trends, seasonality, and other relevant factors influencing car sales. The primary purpose is to provide car manufacturers, dealerships, and stakeholders with a valuable tool for anticipating sales fluctuations, optimizing inventory management, and formulating targeted marketing strategies. The project report will outline the process, methodologies, and results of the analysis, ultimately contributing to the advancement of time series analysis in the context of car sales prediction.

2 LITERATURE SURVEY

2.1 Existing Problem

Emir Zunic et al. proposed the Application of Facebook’s prophet algorithm for successful sales forecasting based on real-world data [1] wherein the approach is claimed to be useful for companies operating in the retail industry due to the use of prophet and backtesting strategy. The proposed framework generates monthly and quarterly sales forecasts, creating a product portfolio along with expected reliability.

A Raiyani et al. proposed Usage of time series forecasting model in Supply chain sales Prediction [2] wherein a representation of a study related to supply chain operation data is made. The data consists of details of 100 different store items from 10 stores using 5 years of history. Optimal performance is provided by the model hybridized using ARIMA (Auto Regressive Integrated Moving Average) and Prophet. Proposed methodology requires prior knowledge or experience of forecasting time series analysis data since the seasonal trends beneath the data are calculated automatically and parameters which are easy to understand are also offered.

Bineet Kumar Jha et al. proposed Time Series Forecasting Model for Supermarket Sales using FB-Prophet [3] examining the efficiency of FB-Prophet tool for sales prediction of supermarket data. The research concludes that FB Prophet is a better prediction model in terms of low error, better prediction, and better fitting when compared with ARIMA. Since scalability of the dataset could turn out to be a significant issue, a transfer learning approach can be used along with FB Prophet and handle large datasets.

Costin-Gabriel Chiru et al. proposed Time Series Analysis for Sales Prediction [4] as an attempt to forecast the daily quantity from each product to be sold by Vivre, a leading online retailer for Home and Lifestyle in Central and Eastern Europe. Besides using the most sought-for ARIMA algorithm, FBProphet was also tested to produce high quality forecasts for time series data that has multiple seasonality with linear or non-linear growth. FBProphet is reported to have performed better than ARIMA in the aspect of allowing multi variate dependencies.

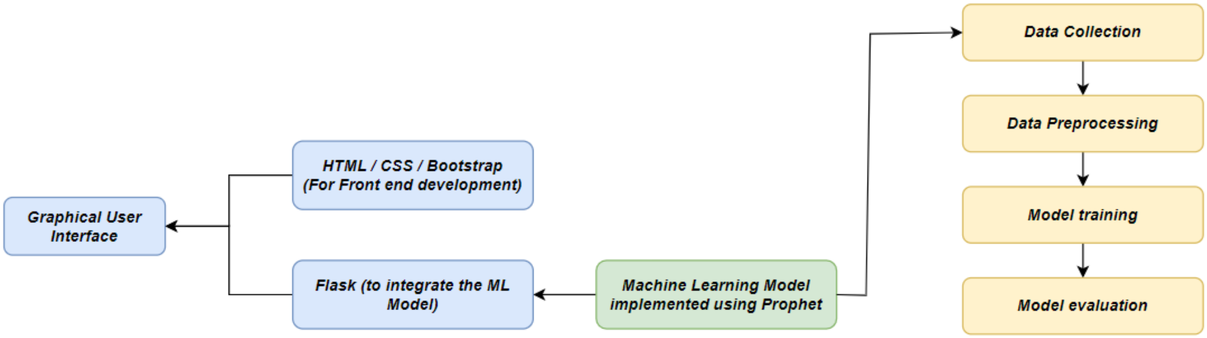
Manzoor Ansari et al. proposed An Intelligent IoT-Cloud-Based Air Pollution Forecasting Model Using Univariate Time-Series Analysis [5] wherein Prophet has been used as an advanced forecasting model. It has been implemented considering 3 main features: trend, seasonality, holidays, and demand for high-quality forecasting. Based on the Prophet model, the seasonality function of the daily, weekly, and annually AQI data can be represented. It has been concluded that the Prophet method is most effective if the historical data series has significant seasonal effects over several seasons.

2.2 Proposed Solution

- It has been observed that Prophet algorithm has not been used in specific car sales prediction in the existing works. As it is reported that prophet has the strenghts of handling historical data series for trend prediction while allowing multi variante dependencies, implementation of the same enhances the accuracy of results obtained
- It has been observed that car sales prediction has not been performed without the usage of more than 2 data fields. To address this issue, the complexity of the model has been reduced significantly by using 2 direct variables and the model has been constructed to produce outputs without having to compromise on the accuracy
- A user friendly GUI has been implemented in order to make the process much easier for the user and the results are produced in a layman format to boost the reliability.

3 THEORETICAL ANALYSIS

3.1 Block Diagram


Data Collection
Data Preprocessing
Model training
Model evaluation
HTML / CSS / Bootstrap (For Front end development)
Graphical User Interface
Flask (to integrate the ML Model)
Machine Learning Model implemented using Prophet

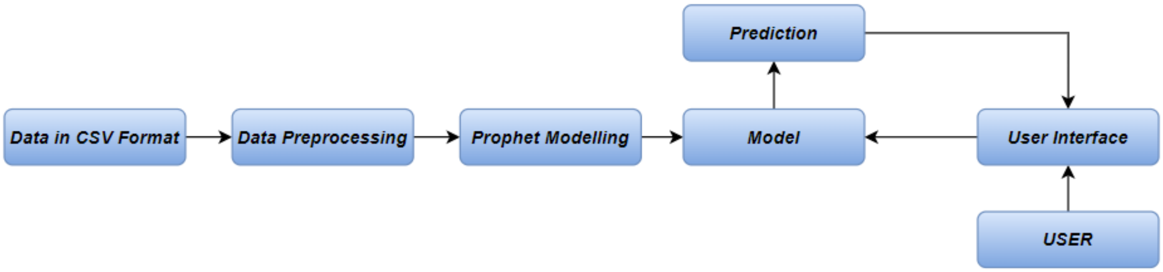
3.2 Software Designing

- HTML and CSS along with Bootstrap were the frameworks used to build the Graphical User Interface (GUI)
- Flask is a library in python which was used to connect the GUI to the backend Machine Learning Model
- Prophet Library was used for the primary task of time series forecast using 2 data fields from the dataset (ds and y)
- Model training was performed using Google colaboratory platform for cloud based implementation.

4 EXPERIMENTAL INVESTIGATIONS

- The GUI has been developed with precautions to make it user efficient and accurate at the same time. This milestone had exceptions with input and output format which was rectified consequently
- The model evaluation results reflect the performance of the model against ground truth. Even though the graph lines do not coincide, they converge at almost all significant points.

5 FLOWCHART



6 RESULTS

The model has been trained efficiently and the plots for various data fields has been mentioned below

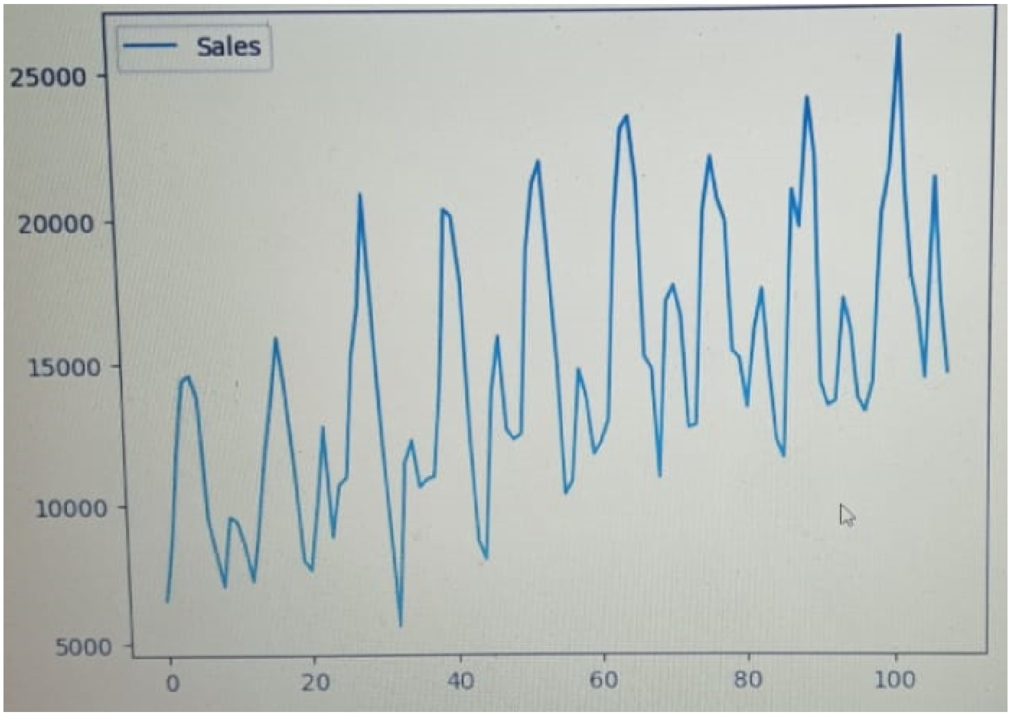


Fig 1. Year vs Revenue

Convergence of Predicted points against ground truths has been plotted

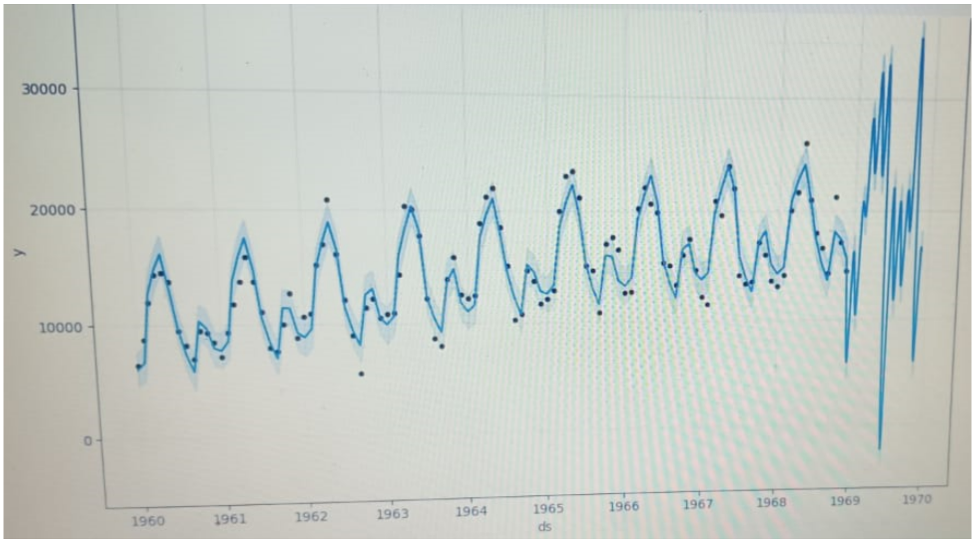


Fig 2. Ground truth vs Actual values

Model with Least Mean Squared Error (MSE) has been chosen as the final and best model

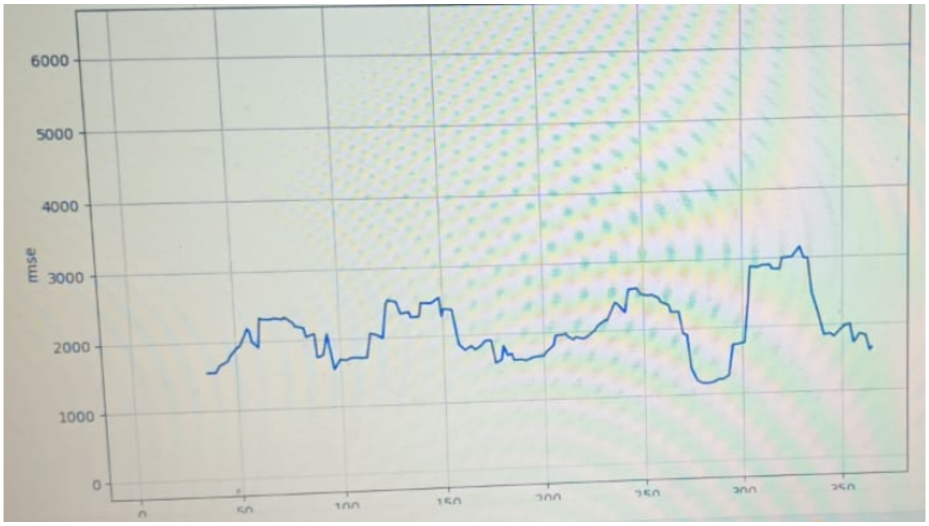


Fig 3. Graph of Model with Least MSE

Overall, graph of predicted values seems to converge with actual values at significant points thus affecting the accuracy at a least negotiable manner

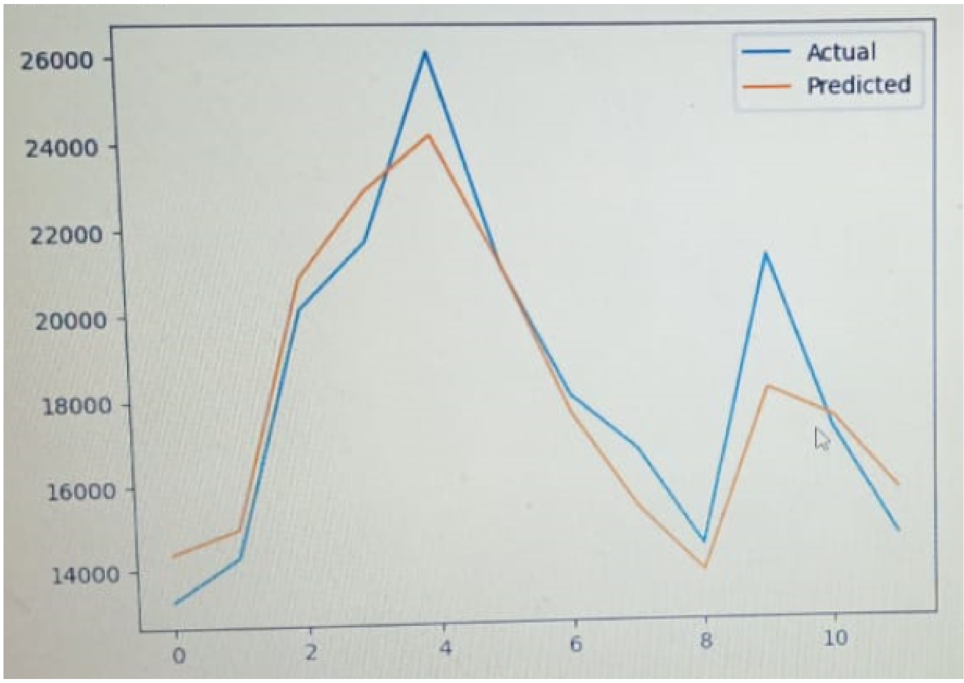
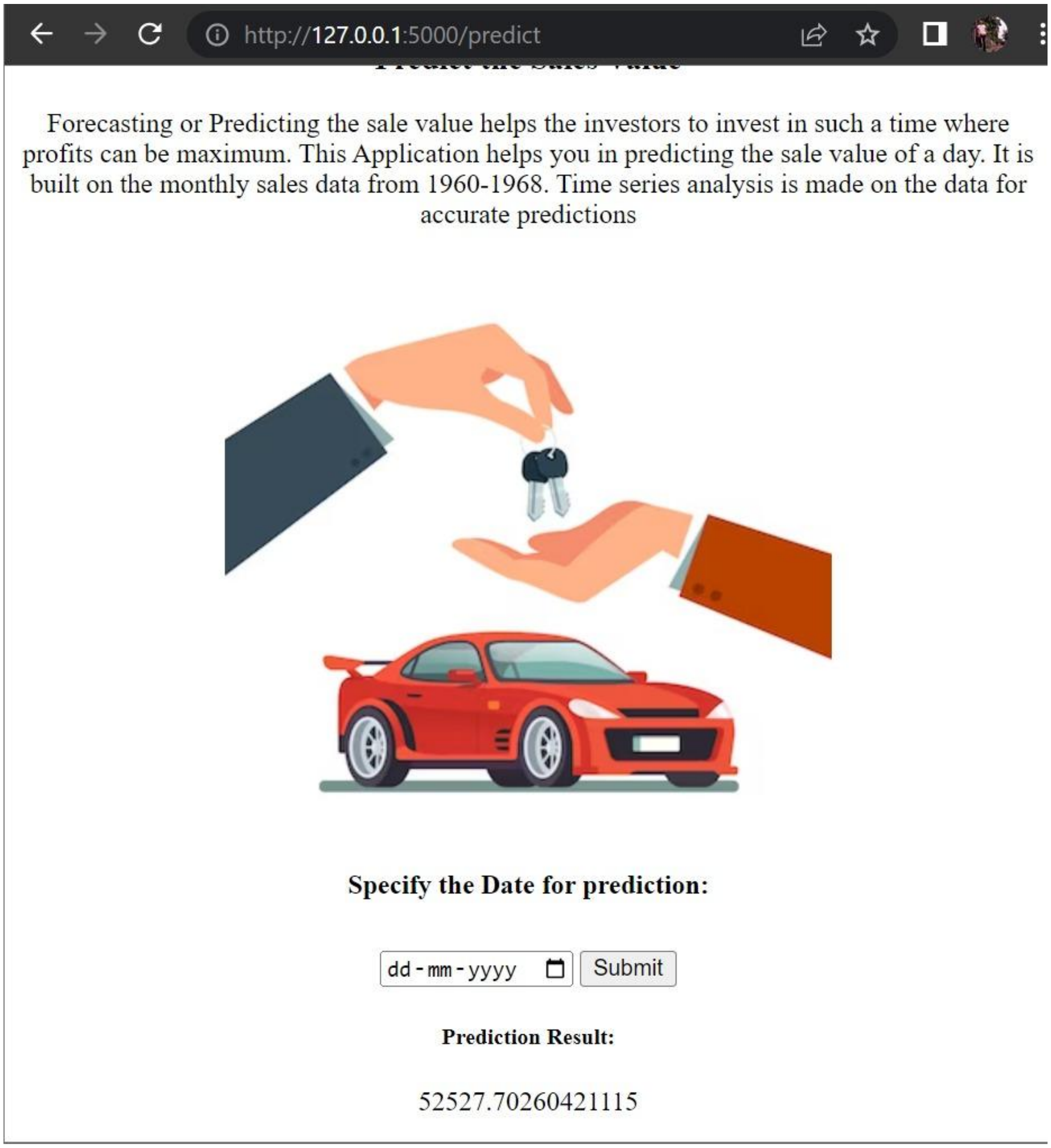


Fig 4. Actual Values vs Predicted Values



Predict the Sales 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:

Prediction Result:

52527.70260421115

Fig 5. Demonstration of Final Output

7 ADVANTAGES AND DISADVANTAGES

7.1 Advantages

- Prophet is a time series forecasting model that is designed to be accurate. It incorporates advanced modeling techniques to capture the various components of time series data, such as trend, seasonality, and holidays. This allows Prophet to make accurate predictions of car sales, which can help businesses make informed decisions based on reliable forecasts.

- Prophet is a time series forecasting model that is both flexible and easy to use. It has a user-friendly interface and simplified modeling approach, making it accessible to users with varying levels of expertise in time series analysis. Prophet also automates many of the complex tasks involved in time series forecasting, such as handling missing data, outliers, and changes in trend and seasonality. This allows analysts to focus on interpreting the results, rather than on the technical aspects of the modeling process.
- Prophet is a time series forecasting model that can handle seasonality and holidays. It has built-in functionality to model and incorporate various types of seasonal patterns and holiday effects in the forecasting process. This is important for car sales forecasting, as sales often exhibit recurring seasonal patterns influenced by factors such as holidays, weather, and economic cycles.

7.2 Disadvantages

- The simplified modeling approach may not be able to accommodate complex relationships or capture nuanced patterns in car sales data. Prophet assumes certain assumptions about the data, such as smoothness and additivity, which may not always hold true for every time series dataset.
- It does not inherently incorporate the impact of external factors or exogenous variables on car sales. While Prophet allows for the inclusion of additional regressors, the model's ability to capture complex interactions between the time series and external factors may be limited.
- Data preprocessing is an important step in the time series forecasting process. It involves cleaning, handling missing values, and addressing outliers and anomalies in the data. This is important because inadequate preprocessing can adversely impact the accuracy and reliability of the forecasts.

8 APPLICATIONS

- Time series analysis can help car manufacturers and dealerships gain insights into sales patterns and seasonality. This information can be used to develop effective sales and marketing strategies, such as targeted promotions, pricing adjustments, and advertising campaigns aligned with peak sales periods or seasonal trends.

- Accurate car sales predictions can help businesses mitigate risks associated with inventory management and financial planning. By anticipating fluctuations in demand, businesses can avoid excess inventory, which can lead to high carrying costs or potential write-offs. Additionally, accurate forecasts can help businesses plan their finances proactively, ensuring that they have sufficient cash flow and budget allocation

- Time series analysis can help businesses simulate and evaluate different scenarios to assess the potential impact on car sales. By adjusting key variables, such as economic indicators, market conditions, or marketing strategies, businesses can analyze the potential outcomes and make informed decisions based on these insights.

9 CONCLUSION

This project focusses on applying time series analysis techniques, specifically utilizing the Prophet forecasting model, for car sales forecasting. Through a comprehensive analysis of historical car sales data, the project aimed to develop accurate and reliable forecasts to assist car manufacturers and dealerships in making informed business decisions.

The project successfully demonstrated the effectiveness of Prophet in capturing and modeling the various components of time series data, such as trend, seasonality, and holiday effects. By leveraging the flexibility and ease of use offered by Prophet, the project provided valuable insights into car sales patterns and trends, enabling stakeholders to anticipate future demand and optimize their operations accordingly.

In conclusion, this project report serves as a valuable resource for car manufacturers, dealerships, and industry professionals interested in leveraging time series analysis and the Prophet model for accurate and reliable car sales forecasting. The insights and methodologies presented here provide a foundation for future research and application of time series analysis in the automotive industry, enabling businesses to navigate the complex dynamics of car sales with greater confidence and success.

10 FUTURE SCOPE

- Further exploration can be done to optimize the Prophet model by fine-tuning hyperparameters and incorporating additional features. Experimentation with different seasonality settings, trend flexibility, and parameter tuning techniques could enhance the forecasting accuracy.

- The inclusion of external factors or exogenous variables, such as economic indicators, marketing campaigns, or competitor data, could provide a more comprehensive analysis of car sales dynamics. Investigating the impact of these variables and incorporating them into the forecasting model may lead to improved accuracy and actionable insights.

- Integrating the developed forecasting model into decision support systems or business intelligence platforms can enable seamless integration and utilization of the forecasts within existing business workflows. This can enhance the practical applicability and adoption of the forecasting model in real-world business scenarios.

11 BIBILIOGRAPHY

[1] Zunic, E., Korjenic, K., Hodzic, K., & Donko, D. (2020). Application of Facebook's prophet algorithm for successful sales forecasting based on real-world data. arXiv preprint arXiv:2005.07575.

[2] Raiyani, A., Lathigara, A., & Mehta, H. (2021). Usage of time series forecasting model in Supply chain sales prediction. In IOP Conference Series: Materials Science and Engineering (Vol. 1042, No. 1, p. 012022). IOP Publishing.

[3] Jha, B. K., & Pande, S. (2021, April). Time series forecasting model for supermarket sales using FB-prophet. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 547-554). IEEE.

[4] Chiru, C. G., & Posea, V. V. (2018). Time series analysis for sales prediction. In Artificial Intelligence: Methodology, Systems, and Applications: 18th International Conference, AIMSA 2018, Varna, Bulgaria, September 12–14, 2018, Proceedings 18 (pp. 163-172). Springer International Publishing.

[5] Ansari, M., & Alam, M. (2023). An Intelligent IoT-Cloud-Based Air Pollution Forecasting Model Using Univariate Time-Series Analysis. Arabian Journal for Science and Engineering, 1-28.

APPENDIX

SOURCE CODE

App.py (Flask):

```
from flask import Flask, render_template, request
import pandas as pd
from prophet import Prophet
import pickle

app = Flask(__name__)


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

```
@app.route('/predict', methods=['POST'])
def predict():
    date = request.form['date']

    # Load the model
    with open('forecast.h5', 'rb') as file:
        model = pd.read_pickle(file)

    # Perform the prediction
    future_date = pd.DataFrame({'ds': [date]})
    forecast = model.predict(future_date)
    prediction = forecast['yhat'].values[0]

    return render_template('index.html', prediction=prediction)



if __name__ == '_main_':
    app.run()
```

Index.html:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
  <h2>Forecasting on the Car Sales Value</h2>

  <div class="container">
```

```
    <div class="row">
      <div class="col-sm">
        <h3>Predict the Sales Value</h3>
        <p>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</p>
        <img src="https://img.freepik.com/free-vector/buying-renting-new-used-speedy-sports-car_3446-651.jpg?w=360">
      </div>

      <div class="col-sm">
        <div class="form-group">
          <label for="dateInput">Specify the Date for prediction:</label>
          <form action="/predict" method="POST">
            <input type="date" class="form-control" id="dateInput" name="date" required>
            <button type="submit" class="btn btn-primary">Submit</button>
          </form>
        </div>
        {% if prediction %}
        <h5>Prediction Result:</h5>
        <p>{{ prediction }}</p>
        {% endif %}
      </div>
    </div>
  </div>
</body>
</html>
```

style.css

```
body {
   text-align: center;
}

h1 {
   color: #333;
}

form {
   margin-top: 20px;
}

label {
   font-weight: bold;
}

button {
   margin-top: 10px;
}

.result {
   margin-top: 20px;
   background-color: #f5f5f5;
   padding: 10px;
}
```

Best Model (forecast.h5):

```
import pandas as pd
import prophet
import matplotlib.pyplot as plt
%matplotlib inline
from google.colab import drive
drive.mount('/content/drive')

df=pd.read_csv('/content/drive/My Drive/Colab Notebooks/monthly-car-sales.csv')
df.head()
df.tail()
df.plot()

df.columns = ['ds','y']
df.head()
df.tail()

df['ds'] = pd.to_datetime(df['ds'])
df.head()
df.tail()

from prophet import Prophet
# Initialize the Model
model=Prophet()
Df.columns
df.dropna(axis=0,inplace=True)
df.isnull().sum()
df.head()
model.fit(df)
```

```
Import joblib
joblib.dump(model,"sales.sav")
import pickle
with open('forecast.h5', 'wb') as f:
   pickle.dump(model, f)


Model.component_modes
df.tail()

### Create future dates of 365 days
future_dates=model.make_future_dataframe(periods=365)
future_dates.tail()
future_dates.head()

prediction=model.predict(future_dates)
prediction[['ds','yhat','yhat_lower','yhat_upper']].head()
model.plot(prediction)

#### Visualize Each Components[Trends,yearly]
model.plot_components(prediction)
df.head()


from prophet.diagnostics import cross_validation
df_cv = cross_validation(model, initial='730 days', period='180 days', horizon = '365 days')
df_cv.head()


from prophet.diagnostics import performance_metrics
df_p = performance_metrics(df_cv)
df_p.head()
```

```
from prophet.plot import plot_cross_validation_metric
fig = plot_cross_validation_metric(df_cv, metric='rmse')

from sklearn.metrics import mean_absolute_error
from matplotlib import pyplot

future = list()
for i in range(1, 13):
        date = '1968-%02d' % i
        future.append([date])
future = pd.DataFrame(future)
future.columns = ['ds']
future['ds'] = pd.to_datetime(future['ds'])
# use the model to make a forecast
forecast = model.predict(future)
y_true = df['y'][-12:].values
y_pred = forecast['yhat'].values
mae = mean_absolute_error(y_true, y_pred)
print('MAE: %.3f' % mae)
# plot expected vs actual
pyplot.plot(y_true, label='Actual')
pyplot.plot(y_pred, label='Predicted')
pyplot.legend()
pyplot.show()
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
np.sqrt(mean_squared_error(y_true,y_pred))
```