VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018, Karnataka, India

A Mini Project On

"Price Elasticity Demand"

Submitted in partial fulfillment of the Mini Project requirement for sixth semester Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

A Sahana

(1TJ21CS002)

Under the guidance of

Dr. Praynlin E

Associate Professor

T. JOHN INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University)

#86/1, Gottigere, Bannerghatta Road, Bengaluru-560083

2023-2024

(Affiliated to Visvesvaraya Technological University)
Approved by AICTE, Govt. of India, New Delhi.
No. 86/1, Gottigere, Bannerghatta Road, Bengaluru-560083
2023-2024

DEPARTMENT OF COMPUTER SCIENCE
& ENGINEERING

CERTIFICATE

Certified that the mini project work entitled “Price Elasticity Demand” carried out by **A Sahana (1TJ21CS002)**, bonafide student of **T. John Institute of Technology** in partial fulfillment for sixth semester of Bachelor of Engineering Computer Science and Engineering of Visvesvaraya Technological University, Belagavi,during the year 2023- 24. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

GUIDE	HOD
Dr.Praynlin E	Dr. Suma R
Associate Professor	Assoc. Professor & Head

Internal Examiner	**External Examiner**

DECLARATION

A Sahana, sixth semester students declare that the mini project entitled **"Price Elasticity Demand"** has been carried out and submitted by me in partial fulfillment of sixth semester of Bachelor of Engineering in Computer Science and Engineering, Visvesvaraya Technological University, Belagavi during the academic year 2023-2024.I also declare that, to the best of my knowledge and belief, the work reported here is accepted and satisfied.

A SAHANA
(1TJ21CS002)

ABSTRACT

An “**Price Elasticity Demand”** Prediction of future movement of stock prices has always been a challenging task for researchers. While the advocates of the efficient market hypothesis (EMH) believe that it is impossible to design any predictive framework that can accurately predict the movement of stock prices, there are seminal work in the literature that have demonstrated that the seemingly random movement patterns in the time series of a stock price can be predicted with a high level of accuracy. The design of such predictive models requires the choice of appropriate variables, right transformation methods of the variables, and tuning of the parameters of the models. In this paper, we present a very robust and accurate framework of stock price prediction that consists of an agglomeration of statistical, machine learning, and deep learning models. We use daily stock price data, collected at five minutes intervals of time, of a very well-known company that is listed in the National Stock Exchange (NSE) of India. The granular data is aggregated into three slots in a day, and the aggregated data is used for training and building the forecasting models. We contend that the approach of model building that uses a combination of statistical, machine learning, and deep learning approaches, can very effectively learn from the volatile and random movement patterns in stock price data. This effective learning will lead to the building of very robust training of the models that can be deployed for short-term forecasting of stock prices, and prediction of stock movement patterns. We build eight classification and eight regression models based on statistical and machine learning approaches. In addition to these models, two deep learning-based regression models using a long-and-short-term memory (LSTM) network have also been built. Extensive results have been presented on the performance of these models, and results are critically analyzed.

ACKNOWLEDGEMENT

The project report on "**Price Elasticity Demand**" is the outcome of guidance, moralsupport and knowledge imparted on me, throughout my work. For this I acknowledge andexpress immense gratitude to all those who have guided and supported me during the preparation of this mini project.

I take this opportunity to express my gratefulness to everyone who has extended their support for helping me in the mini project completion.

First and foremost, I thank **Dr. Thomas P. John**, Chairman of T. John Group of Institutions and **Dr. P Suresh Venugopal**, Principal, T. John Institute of Technology for giving me this opportunity to studying this prestigious institute and also providing me with best of facilities.

I would like to show my greatest appreciation to **Dr. Suma R**, HOD, Dept of CSE and **Dr. Praynlin E,** Mini Project Guide, Dept. of CSE for constantly guiding us throughout the project.

I would also like to thank to all teaching and non-teaching staff of Computer Science and Engineering Department for directly or indirectly helping me in completion of Mini Project.

Lastly and most importantly I convey my gratitude to my parents who have been the source of inspiration and also for instrumental help in successful completion of project.

A Sahana

(1TJ21CS002)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Prediction of future movement patterns of stock prices has been a widely researched area in the literature. While there are proponents of the efficient market hypothesis who believe that it is impossible to predict stock prices, there are also propositions that demonstrated that if correctly formulated and modeled, the prediction of stock prices can be done with a fairly high level of accuracy. The latter school of thought focused on the construction of robust statistical, econometric, and machine learning models based on the careful choice of variables and appropriate functional forms or models of forecasting. There are propositions in the literature that are based on time series analysis and decomposition for forecasting future values of stocks. In this regard, several propositions have been presented in the literature for stock price forecasting following a time series decomposition approach. There is also an extent of literature that deals with various technical analysis of stock price movements. These approaches provide the user with visual manifestations of the indicators which help the ordinary investors to understand which way stock prices are more likely to move soon. In this thesis, we propose a granular approach to forecasting stock price and the price movement pattern by combining several statistical, machine learning, and deep learning methods of prediction on technical analysis of stock prices. We present several approaches for short-term stock price movement forecasting using various classification and regression techniques and compare their performance in the prediction of stock price movement and stock price values. We believe this approach will provide several useful information to the investors in the stock market who are particularly interested in short-term investments for profit.

2.1 GOAL OF THE PROJECT

The primary goal of predicting machine states is to enhance the efficiency, reliability, and overall performance of industrial operations. By accurately forecasting the various operational states of machinery, businesses can achieve several key objectives:

1. **Minimize Downtime:**Predictive insights allow for timely interventions, reducing unplanned downtime and ensuring continuous production.
2. **Optimize Maintenance:**Enable predictive maintenance by identifying potential issues before they escalate, thus extending the lifespan of machinery and reducing maintenance costs.

3. **Increase Efficiency:**Optimize resource allocation and operational workflows by understanding and anticipating machine states, leading to more efficient production processes.
4. **Enhance Safety:** Early detection of error states can prevent accidents and ensure a safer working environment.
5. **Improve Decision-Making:**Provide actionable data for informed decision-making, enabling proactive management of machinery and operations.
6. **Reduce Operational Costs:**Lower costs associated with unexpected breakdowns, emergency repairs, and inefficient use of resources.
7. **Support Real-Time Monitoring:**Facilitate real-time monitoring of machine states, allowing for immediate responses to changing conditions.
8. **Enable Scalability:**Develop scalable solutions that can be applied across various machines and operations, supporting the growth and adaptability of the business.Predicting machine states is a critical component of modern industrial strategies, leverage data analytics and machine learning to drive operational excellence and sustainable growth.

2.2 SCOPE OF THE PROJECT

This project focuses on the comprehensive analysis and prediction of machine states to optimize industrial operations. The scope includes several key areas:

1. **Data Collection and Ingestion:**Systematically gather time series data from machine sensors, including operational parameters, performance metrics, and environmental conditions.
2. **Data Pre-processing:**Clean, normalize, and prepare the collected data to ensure accuracy and consistency, addressing issues such as missing values and noise.
3. **Time Series Analysis:**Apply statistical and analytical techniques to understand patterns, trends, and correlations in the time series data related to machine operations.
4. **Anomaly Detection:**Implement methods to identify deviations from normal operation, which could indicate potential issues or changes in machine state.
5. **Machine Learning Model Development:**Train and validate machine learning models to classify and predict different machine states (e.g., running, idle, maintenance, error, shutdown) based on historical data.

6. **Model Evaluation and Validation:**Assess the performance of the predictive models using metrics such as accuracy, precision, recall, and F1 score, ensuring reliability and robustness.
7. **Real-Time Monitoring:**Develop and deploy systems for real-time monitoring of machine states, enabling timely detection and response to changes in operational conditions.
8. **Integration and Implementation:**Integrate the predictive models and monitoring systems into existing industrial infrastructure, ensuring compatibility and seamless operation.
9. **Continuous Improvement:** Establish processes for ongoing model refinement and system updates based on new data, feedback, and evolving operational needs.

CHAPTER 2

PREDICTION OF MACHINE STATES

1. **Objective:** Predict machine states (running, idle, maintenance, error, shutdown) to enhance operational efficiency.
2. **Data Source:** Time-stamped sensor data from machinery, including operational metrics and performance indicators.
3. **Data Collection:** Gather comprehensive data from various machines for a robust analysis.
4. **Data Preparation:** Clean, normalize, and preprocess data to address issues like missing values and noise.
5. **Feature Engineering:** Extract relevant features from the raw data to improve model performance.
6. **Time Series Analysis:** Analyse temporal patterns and trends in the data to understand machine behaviour.
7. **Anomaly Detection:** Identify deviations from normal operation to flag potential issues early.
8. **Machine Learning Models:** Develop and train models to classify and predict machine states based on historical data.
9. **Model Evaluation:** Assess model accuracy and performance using metrics like precision, recall, and F1 score.
10. **Real-Time Monitoring:** Implement systems for real-time tracking of machine states and performance.
11. **Integration:** Integrate predictive models into existing industrial systems for seamless operation.
12. **Alerts and Notifications:** Provide real-time alerts and notifications for proactive maintenance and operational decisions.
13. **User Interface:** Develop user-friendly interfaces for visualization and interpretation of machine state predictions.
14. **Continuous Improvement:** Update and refine models based on new data and feedback for ongoing accuracy.

CHAPTER 3

MODULES AND LIBRARIES USED

1. **Pandas (pd):**

o**Purpose:** Essential for data manipulation and analysis.

o**Key Functions:**

▪ pd.read_excel(): Reads data from Excel files into Data Frames.

▪ pd.to_datetime(): Converts columns to datetime format for time series analysis.

▪ pd.DataFrame(): Creates and manipulates Data Frames.

2. **NumPy(np):**

o **Purpose:** Provides support for numerical operations and array handling.

o **Key Functions:**

▪ np.array(): Converts lists to arrays for efficient numerical operations.

3. **Scikit-learn (sklearn):**

o **Purpose:** Offers machine learning and data preprocessing tools.

o **Key Modules:**

▪ sklearn.preprocessing: Functions for scaling and normalizing data.

4. **Matplotlib:**

o**Purpose:** For creating static, animated, and interactive plots.

o **Key Functions:**

▪ plt.plot(): Generates line plots to visualize trends and patterns.

▪ plt.show(): Displays the plot, making it easier to analyse visual data.

5. **Seaborn:**

o**Purpose:** Enhances Matplotlib with additional features for statistical plotting

o**Key Functions:** sns.lineplot():creates refined line plots with better aesthetic

CHAPTER 4

IMPLEMENTATION

5.1 Define Objectives and Requirements:

- **Objective:** Predict machine states (running, idle, maintenance, error, shutdown) to enhance operational efficiency.
- **Requirements:** Determine data sources, define machine states, and establish success metrics.

5.2 Data Collection and Ingestion:

- **Source Data:** Collect time series data from machine sensors. Ensure it includes relevant metrics like operational parameters and performance indicators.
- **Load Data:** Use libraries such as Pandas to load and manage data.

```python
import pandas as pd

# Load the dataset from the Excel file
df = pd.read_excel('path/to/your/dataset.xlsx')
```

Fig 5.2:Data Collection and Ingestion

5.3. Data Preprocessing:

- **Convert Data Types:** Ensure columns like timestamps are in the correct format.

```python
df['_time'] = pd.to_datetime(df['_time'])
```

Fig 5.3.1:Data Processing

- **Handle Missing Values:** Fill or impute missing values to avoid issues in analysis.

```python
df.fillna(method='ffill', inplace=True)  # Forward fill missing values
```

Fig 5.3.2: Handle missing values

- **Feature Engineering:** Extract and create features relevant to predicting machine states.

```python
df['rolling_mean'] = df['_value'].rolling(window=5).mean()
```

Fig 5.3.3: Extracting data

5.4 Exploratory Data Analysis (EDA):

- **Visualize Data:** Use Matplotlib and Seaborn to explore trends and patterns.

```python
import matplotlib.pyplot as plt
import seaborn as sns

sns.lineplot(data=df, x='_time', y='_value')
plt.show()
```

Fig 5.4: Data Analysis •

Analyse Trends: Identify trends and anomalies in the data.

5.5 Develop Machine Learning Models:

- **Split Data:** Divide the data into training and testing sets.

```python
from sklearn.model_selection import train_test_split

X = df[['feature1', 'feature2']]  # Replace with relevant features
y = df['state']  # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
```

Fig 5.5.1:Developing Model

Train Models: Use Scikit-learn or TensorFlow/Kera's to build and train classification models.

•

```python
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier()
model.fit(X_train, y_train)
```

Fig 5.5.2: Train Model

• **Evaluate Models:** Assess model performance using metrics like accuracy, precision, recall, and F1 score.

```python
from sklearn.metrics import classification_report

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Fig 5.5.3: Extract Model

5.6 Real-Time Monitoring and Deployment:

• **Build APIs:** Use Flask or Fast API to create an API for real-time predictions.

```python
from flask import Flask, request, jsonify
app = Flask(__name__)

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    prediction = model.predict([data['features']])
    return jsonify({'prediction': prediction[0]})

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

Fig 5.6.1: Data Monitoring

Deploy Models: Use Docker and cloud platforms (AWS, GCP, Azure) to deploy the model.

•

dockerfile Copy code

```dockerfile
# Dockerfile example
FROM python:3.8
WORKDIR /app
COPY requirements.txt ./
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "app.py"]
```

Fig 5.6.2: Deploy Models

5.7 Monitor and Improve:

- **Track Performance:** Use tools like Flow to monitor model performance and track experiments.
- **Update Models:** Regularly update models with new data and refine them based on performance feedback.

5.8 Documentation and Reporting:

- **Document the Process:** Keep comprehensive documentation of the project, including methodologies, model performance, and deployment steps.
- **Report Findings:**Provide clear reports and dashboards to stakeholders using visualization libraries or business intelligence tools.

Following these steps, you can systematically implement a project to predict machine states, levering data analytics and machine learning to improve operational efficiency and decision making.

CHAPTER 5

RESULTS

6.1 Screenshots

6.1.1 Add Required Data

This figure represents the data we have uploaded in our file.

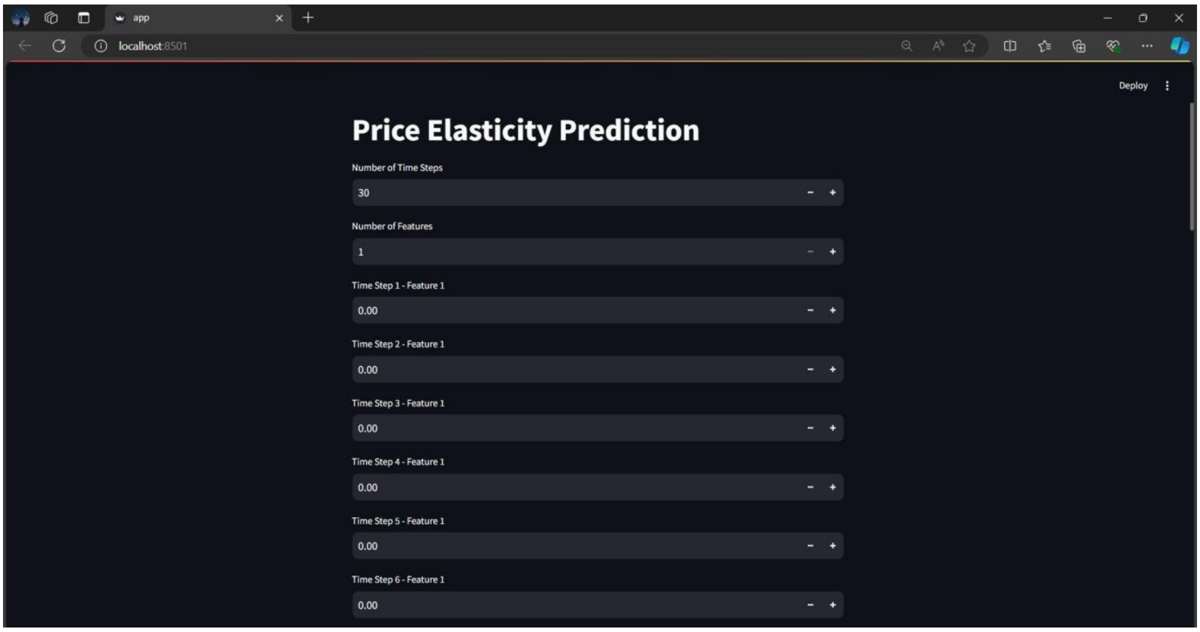

Fig 6.1.1: Add required data

6.1.2 Predicted Results

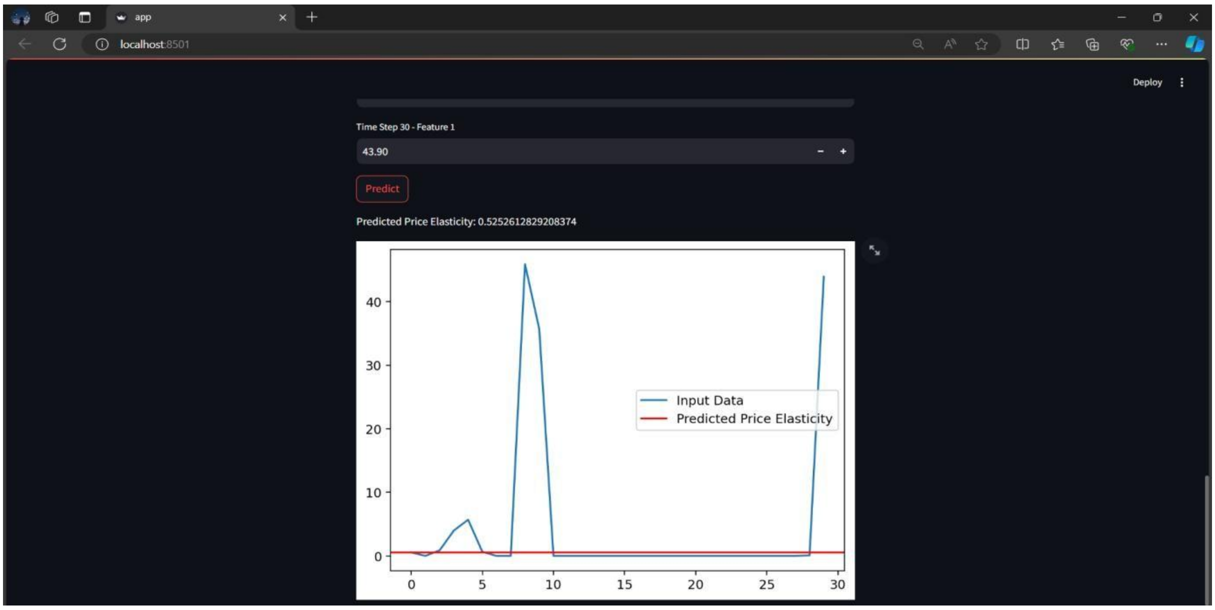

Fig 6.1.2: Prediction Graph

CONCLUSION

The provided script demonstrates an effective method to visualize time series data from a CSV file. By leveraging pandas for data manipulation and matplotlib along with seaborn for plotting, the script generates clear, aesthetically pleasing subplots for each column in the dataset. The use of seaborn's dark grid style enhances the readability of the plots, making trends and patterns in the data more apparent. This visualization technique is essential for initial data exploration, allowing quick identification of any irregularities or notable trends. The script's straightforward approach ensures that it can be easily adapted to different datasets with minimal modification. Such visualizations are crucial in predictive maintenance projects, aiding in the analysis of machine states and operational conditions. Overall, this script provides a solid foundation for further time series analysis and model development.

REFERENCES

1) https://link.springer.com/article/10.1007/s00542-019-04552-w

2) https://www.researchgate.net/publication/