**PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING**

**Phase- 5**

**Development and Submission**

**Submitted By :**

**Murali Keshava**

**au723921104024**

[keshava33377@gmail.com](mailto:keshava33377@gmail.com)

Introduction to Product Demand Prediction with Machine Learning:

Product demand prediction is a critical aspect of supply chain management and business operations. It involves using machine learning techniques and data analysis to forecast how many units of a product customers are likely to purchase over a specific time frame. Accurate demand prediction is essential for optimizing inventory levels, ensuring customer satisfaction, and maximizing profitability. In this introduction, we'll explore the fundamentals of product demand prediction with machine learning.

Key Concepts:

1.Historical Data:

Demand prediction relies on historical sales, order, and other relevant data. This data provides a foundation for building machine learning models that can recognize patterns and relationships.

2.Features:

Features are the variables or attributes used in the prediction process. They can include product characteristics, pricing, promotional activities, seasonality, customer demographics, and more.

3. Supervised Learning:

Demand prediction typically falls under supervised machine learning, where the model learns from historical data with known outcomes (e.g., sales numbers). It's trained to make predictions based on the relationships it discovers.

4. Time Series Analysis:

Many product demand prediction scenarios involve time series data, where observations are recorded at regular intervals. Time series analysis helps capture trends, seasonality, and periodic patterns.

Design Thinking Process Steps in Product Demand Prediction:

1. Data Collection:

Gather historical sales data, customer orders, and any relevant external factors that may influence demand, such as holidays or market trends.

2.Data Pre-Processing :

Clean and prepare the data by handling missing values, outliers, and encoding categorical variables. Data preprocessing ensures that the dataset is suitable for machine learning.

3. Feature Engineering:

Select and create features that are likely to impact product demand. Feature engineering can involve aggregating data by time, creating lag features, and extracting relevant information.

4. Model Selection:

Choose the appropriate machine learning model for demand prediction. Common models include regression models, decision trees, random forests, and more advanced techniques like neural networks.

5. Model Training:

Train the selected model on the historical data, using a portion of the data for training and another for validation. The model learns to make predictions by identifying patterns and relationships in the data.

6. Evaluation and Validation:

Assess the model's accuracy and performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE). Validation helps ensure the model's reliability.

7. Deployment:

Once a satisfactory model is developed, it can be integrated into the company's systems to generate real-time or periodic demand predictions.

8.Monitoring and Feedback Loop:

Continuously monitor the model's performance and make necessary updates. Feedback from operations teams and customers can be used to improve the model over time.

Benefits of Machine Learning in Demand Prediction:

Accuracy Machine learning models can capture complex relationships and patterns in the data, leading to more accurate demand forecasts.

Efficiency Automated prediction processes save time and resources compared to manual forecasting methods.

Adaptability Machine learning models can adapt to changing market conditions and customer preferences.

Scalability Models can be scaled to handle large datasets and real-time prediction needs.

In conclusion, product demand prediction with machine learning is a data-driven approach that enhances businesses' ability to manage inventory efficiently, optimize pricing strategies, and meet customer expectations. It is a valuable tool for decision-making and improving overall supply chain management.

**Description Of Data :**

A dataset for product demand prediction with machine learning is a collection of structured data that contains historical information about product sales, orders, and related factors. The dataset serves as the foundation for training and testing machine learning models to forecast future product demand accurately. Here's a description of the key components of such a dataset:

**1.Date/Time Stamp:**

Each data point should include a date or timestamp indicating when a particular event occurred. This is essential for time series analysis and capturing trends, seasonality, and periodic patterns.

**2.Product Information:**

Details about the products being sold, including product ID, category, brand, and any relevant attributes. This information is essential for associating product characteristics with demand.

**3.Customer Information:**

Information about the customers, which may include customer ID, location, demographics, or other characteristics. Understanding customer behavior and preferences can help in predicting demand.

4. Demand Data:

The most critical component, this includes information on how many units of a product were sold or ordered. It's the target variable that the machine learning model aims to predict.

**5. Pricing Information:**

Historical data on product prices, discounts, and promotions. Pricing data is essential for assessing price elasticity and its impact on demand.

**6. External Factors:**

Any external factors that could influence demand. This may include data on holidays, weather conditions, economicindicators, or other variables that might affect purchasing behavior.

**7. Stock Levels:**

Data on the availability of products in inventory at different time points. Low stock levels can lead to stockouts, affecting demand.

**8. Competitor Data:**

Information about competitor activities, such as price changes or promotions, can provide insights into the competitivelandscape and its impact on demand.

**9. Geographical Data:**

**If applicable, location-related data, such as store locations, customer locations, or regional data, can help in regional demand prediction.**

10. Order Lead Times :

Information about the lead times for order fulfillment, which can affect demand. Longer lead times might discourage customers.

**11. Marketing Activities**: Historical data on marketing campaigns, advertising efforts, and their timing. This data helps in understanding the effect of marketing on demand.

**12. Customer Reviews and Feedback** :

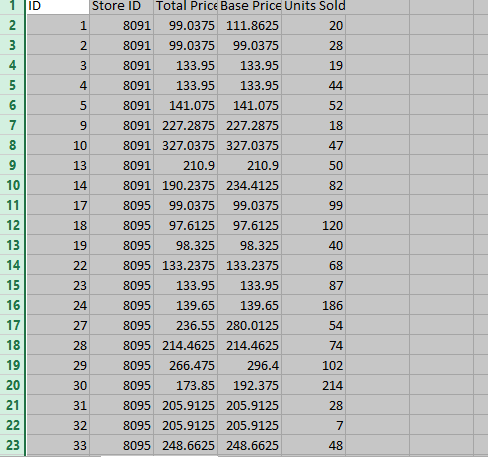
If available, customer reviews and feedback data can provide additional insights into productdemand. Sentiment analysis can be applied to this data.

**Data Size:**

The dataset's size can vary significantly based on the company's scale and data availability. Larger datasets with more historical data points typically result in more accurate machine learningmodels.

A well-structured and clean dataset is essential for building robust machine learning models for product demand prediction. The quality and relevance of the data directly impact the accuracy of predictions and, consequently, the success of inventory management and business operations.

GIVEN DATASET :



**Data Pre-Processing** :

Loading and preprocessing datasets are crucial steps in product demand prediction with machine learning.

⚫Data Integrity and Consistency:

Loading ensures the data is properly read into the system, maintaining its integrity and consistency for accurate analysis.

⚫Data Understanding:

Preprocessing involves exploring and understanding the data, identifying missing values, outliers, and inconsistencies that may impact prediction models.

⚫Feature Engineering:

Preprocessing allows for feature extraction, transformation, and selection, enhancing the quality and relevance of features for predicting product demand.

⚫Normalization and Scaling:

Preprocessing aids in normalizing and scaling features, ensuring uniformity and preventing bias towards certain features during model training.

⚫Handling Missing Values:

Preprocessing addresses missing data through imputation or removal, enabling effective utilization of available information.

⚫Dealing with Categorical Data:

Conversion of categorical variables to numerical representations facilitates their integration into machine learning models.

⚫Dimensionality Reduction:

Techniques like PCA or feature selection help in reducing the dataset's dimensions while retaining essential information, improving model efficiency and performance.

⚫Data Splitting:

Preprocessing often involves dividing the dataset into training, validation, and test sets, ensuring an unbiased evaluation of the model's performance.

loading and preprocessing datasets pave the way for cleaner, more meaningful data, ultimately enhancing the accuracy and effectiveness of machine learning models in predicting product demand

**Python Program :**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv("Historical Product Demand.csv")

data

data.head(10)

data.tail(10)

data.isna().sum()

data.dropna(axis = 0, inplace = True)

data.isna().sum()

data.describe(include= 'all')

data1 = pd.read\_csv("PoductDemand.csv")

data1

data1.mean()

data1.corr()

sns.heatmap(data1.corr())

plt.show()

plt.boxplot(data1)

data1.groupby('Units Sold').corr()

from sklearn import preprocessing

sc = preprocessing.LabelEncoder()

from sklearn.preprocessing import MinMaxScaler

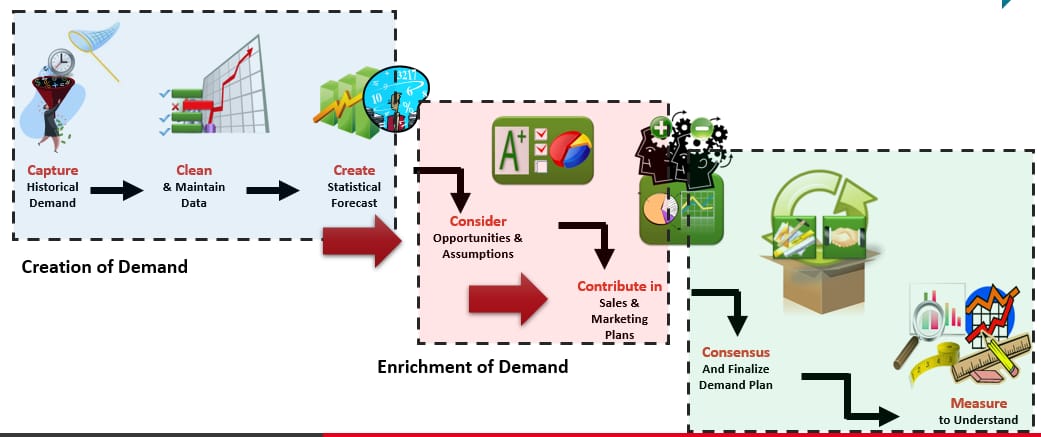
le = MinMaxScaler()

le.fit(x)

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.25, random\_state = 50 )

**ARCHITECTURE OF PRODUCT DEMAND PREDICTION :**



**EXPLANATION :**

Product demand prediction involves using data and statistical techniques to forecast the future demand for a product. This prediction helps businesses make informed decisions regarding production, inventory, and marketing strategies. Several steps are involved in creating a demand prediction modelData Collection: Gather historical sales data, customer preferences, market trends, economic indicators, and any other relevant data sources.

Feature Engineering: Identify and create relevant features (variables) that could impact product demand, such as price, promotions, seasonality, or marketing activities.Model Selection: Choose a suitable predictive model, such as regression, time series analysis, machine learning algorithms (e.g., decision trees, neural networks), or advanced forecasting techniques. Training the Model: Use historical data to train the chosen model, where the model learns the relationships between features and product demand. Validation and Testing Assess the model's performance using validation datasets or techniques like cross-validation to ensure it accurately predicts demand.

Parameter Tuning: Optimize the model by adjusting parameters to achieve the best predictive performance. Forecasting: Apply the trained and validated model to future data to generate predictions for product demand. Evaluation: Evaluate the model's predictive accuracy using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or others suitable for the specific problem. Implementation and Monitoring: Implement the model in the business operations and continually monitor and update the model to adapt to changing market dynamics. By accurately predicting product demand, businesses can optimize their supply chains, manage inventory efficiently, plan marketing campaigns effectively, and ultimately improve their overall business strategy.

**Product Demand prediction With Machine Learning :**

Predicting product demand with machine learning is a valuable application in various industries. To get started, here's a high-level overview of the steps involved:

Data Collection :

Gather historical data on product sales, including factors that may influence demand, such as pricing, promotions, seasonality, and external events.

Clean and pre-process the data. This involves handling missing values, outliers, and encoding categorical variables.

Feature Selection/Engineering : Identify relevant features that can affect product demand. You may need to create new features or transform existing ones.

Data Splitting : Split the data into training, validation, and test sets to evaluate the model's performance.

Model Selection : Choose the appropriate machine learning model. Common choices include linear regression, decision trees, random forests, or more advanced techniques like neural networks. Model Training : Train the selected model using the training data.

Model Evaluation : Evaluate the model's performance using the validation set. Common evaluation metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

Hyper-parameter Tuning : Fine-tune the model's hyper-parameters to optimize its performance.

Model Validation : Assess the model's generalization performance on the test set to ensure it can make accurate predictions on unseen data.

Deployment : Once satisfied with the model's performance, deploy it in a real-world environment to make predictions on future demand.

**Data Loading anc Pre-processing** :

Data Sources : Identify the sources of your data, whether it's stored in databases, spreadsheets, text files, or obtained from APIs. Ensure you have access to the data you need.

Data Retrieval : Use appropriate libraries or tools to load your data into your analysis environment. For example, in Python, you can use libraries like Pandas to read data from various file formats or SQL databases.

Data Inspection: Once you've loaded the data, inspect the first few rows to get an initial understanding of its structure and contents. This step helps you verify that the data was loaded correctly.

Handling Missing Values : Identify and handle missing data. You can choose to remove rows with missing values, fill them with suitable values (e.g., mean, median), or use more advanced imputation techniques.

Dealing with Duplicates : Check for and remove duplicate records if they exist in the dataset.

Data Encoding : Convert categorical variables into numerical format using techniques like one-hot encoding or label encoding. This is necessary for most machine learning algorithms that require numerical inputs.

**Visualization :**



Visualization is a powerful tool for understanding and communicating the results of product demand prediction. Here are some common types of visualizations you can use:

Time Series Plots : If your prediction involves time-dependent data, create time series plots to visualize historical demand and predicted demand over time. You can use line charts to show the actual demand and forecasted demand on the same graph.

Actual vs. Predicted Plots : Compare actual product demand with your model's predictions. Scatter plots or line charts can help you assess how closely your model's predictions align with the real data.

Residual Plots : Plot the residuals (the differences between actual and predicted values) over time. This can help you identify patterns or trends in prediction errors.

Histograms and Density Plots : Visualize the distribution of prediction errors to assess their normality and identify potential bias or skew in the predictions.

Box Plots : Use box plots to visualize the spread and distribution of errors, including outliers. This can help you identify areas where the model performs exceptionally well or poorly

**Evaluation Performance :**

Evaluation Metrics : These are quantitative measures used to assess how well a model or algorithm performs on a particular task. Common evaluation metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), accuracy, precision, recall, F1 score, and many others. The choice of metric depends on the problem you are trying to solve. For example, classification tasks often use metrics like accuracy, while regression tasks use metrics like RMSE.

Training and Testing : In machine learning, you typically split your dataset into a training set and a testing set. The model is trained on the training set and then evaluated on the testing set to assess how well it generalizes to new, unseen data.

**Loading and Importing the libraries**

import numpy as np

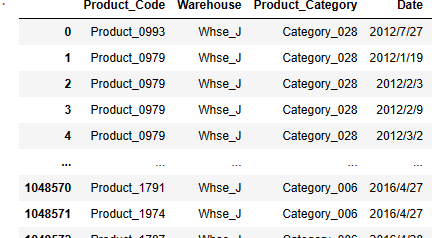
import pandas as pd

import matplotlib.pyplot as plt

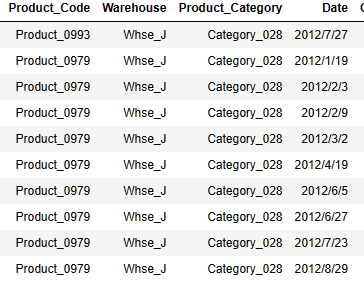
import seaborn as sns

data = pd.read\_csv("Historical Product Demand.csv")

data



data.head(10)



data.tail(10)



data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1048575 entries, 0 to 1048574

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Product\_Code 1048575 non-null object

1 Warehouse 1048575 non-null object

2 Product\_Category 1048575 non-null object

3 Date 1037336 non-null object

4 Order\_Demand 1048575 non-null object

dtypes: object(5)

memory usage: 40.0+ MB

data.isna().sum()

Product\_Code 0

Warehouse 0

Product\_Category 0

Date 11239

Order\_Demand 0

dtype: int64

data.dropna(axis = 0, inplace = True)

data.isna().sum()

Product\_Code 0

Warehouse 0

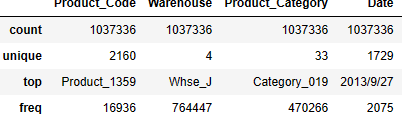
Product\_Category 0

Date 0

Order\_Demand 0

dtype: int64

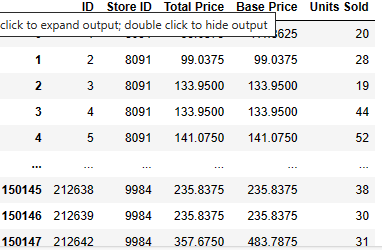
data.describe(include= 'all')



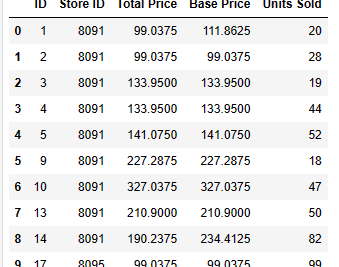
**Loading The Data:**

data1 = pd.read\_csv("PoductDemand.csv")

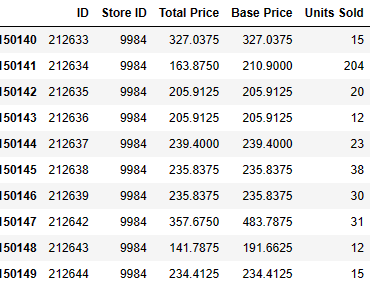
data1



data1.head(10)



data1.tail(10)



**Future Engineering and pre-processing visualization:**

data1.isna().sum()

ID 0

Store ID 0

Total Price 1

Base Price 0

Units Sold 0

dtype: int64

data1.dropna(axis = 0, inplace = True)

data1.isna().sum()

ID 0

Store ID 0

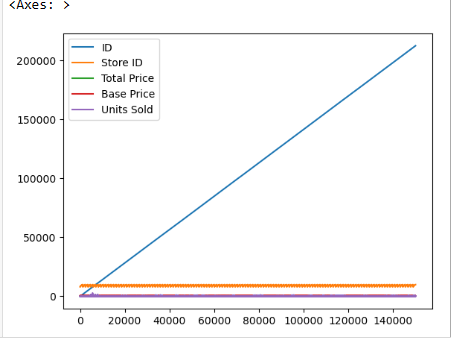
Total Price 0

Base Price 0

Units Sold 0

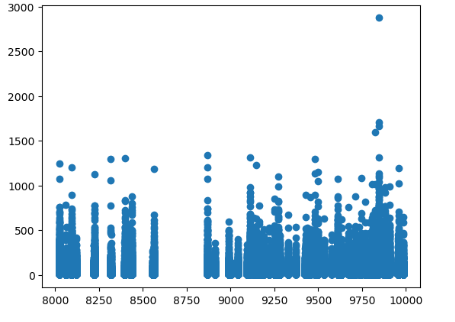
dtype: int64

data1.plot()



plt.scatter(data1["Store ID"], data1["Units Sold"])

plt.show()



data1.mean()

ID 106270.971795

Store ID 9199.420935

Total Price 206.626751

Base Price 219.424262

Units Sold 51.674543

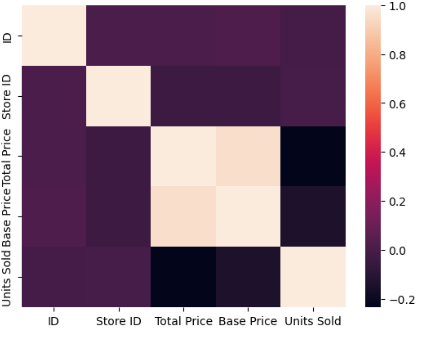
dtype: float64

data1.corr()

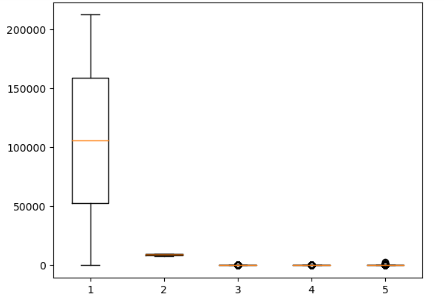


sns.heatmap(data1.corr())

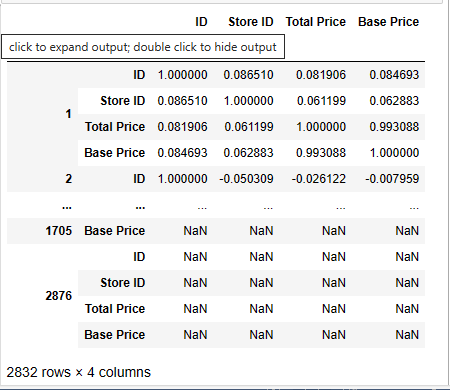
plt.show()



plt.boxplot(data1)



data1.groupby('Units Sold').corr()



data1.mean()

ID 106270.971795

Store ID 9199.420935

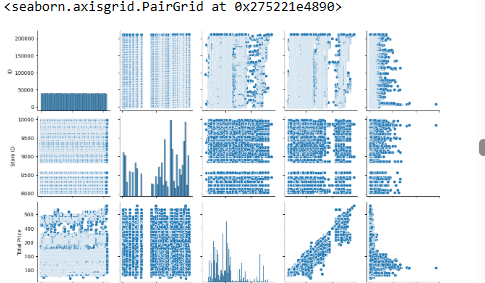
Total Price 206.626751

Base Price 219.424262

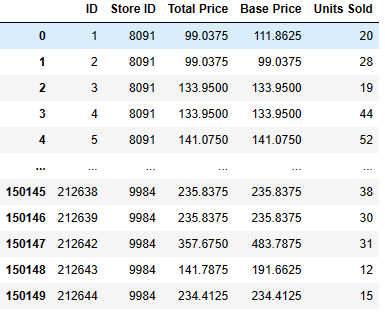
Units Sold 51.674543

dtype: float64

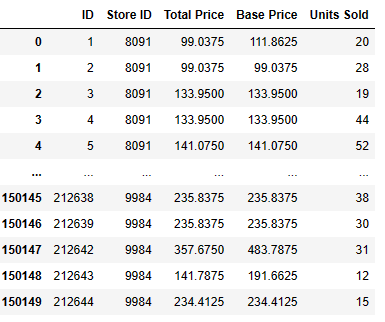
sns.pairplot(data1)



data1.dropna()



data1.fillna(0)



**Linear Regression:**

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

model = lr.fit(x\_train, y\_train)

model.intercept\_

66.30203634254558

model.coef\_

array([ -4.39017685, 0.95330134, -342.45636915, 268.59705875])

y\_pred = model.predict(x\_test)

from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import mean\_absolute\_error

from sklearn.metrics import r2\_score

mean\_absolute\_error(y\_test, y\_pred)

33.233948162416986

mean\_squared\_error(y\_test, y\_pred)

2935.7936428617845

r2\_score(y\_test, y\_pred)

0.1607003705784842

y\_pred.mean()

51.573538982561075

def generateX(N):

x = np.random.random(N)\*2\*np.pi

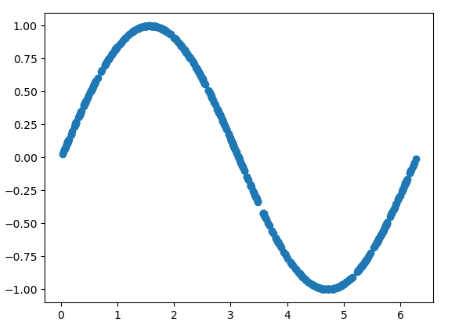
yd = np.sin(x)

return x, yd

x, y = generateX(500)

plt.scatter(x, y)

plt.show()



def plotmodel(x, y, yd):

i = x.argsort()

plt.figure()

plt.plot(x[i], y[i], "g-o")

plt.plot(x[i], yd[i], "r-o")

plt.ylabel("F(X)")

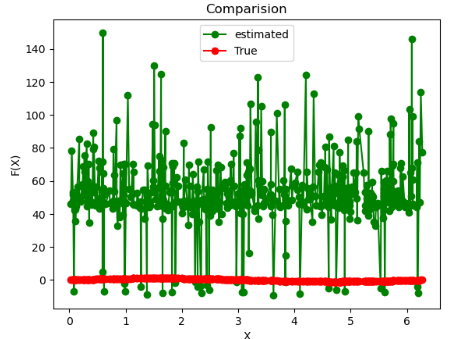
plt.xlabel("X")

plt.legend(["estimated", "True"])

plt.title("Comparision")

plt.show()

plotmodel(x, y\_pred, y)



CONCLUSION:

**Conclusion leveraging machine learning for product demand prediction offers significant potential for businesses. By analyzing historical data and utilizing advanced algorithms, accurate demand forecasts can be generated, aiding in informed decision-making, optimized inventory management, and enhanced customer satisfaction. However, it's crucial to continually refine and update models, incorporate new data sources, and consider market dynamics to ensure the predictions remain reliable and effective in a dynamic business environment.**