

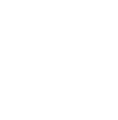
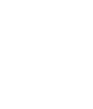
**DOMAIN: Applied data science**

**PROJECT TITLE: product demand prediction with**

**machine learning**

|  |  |
| --- | --- |
|  | |
|  | **PROJECT MEMBERS** |

**COLLEGE CODE: 5113 Phase 5**



G.Ragul-511321104078

S.Raghul-511321104077

K.Sakthivel-511321104085 D.Varun-511321104109

N R.Kirankumar-511321104059

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase 5: Project Documentation & Submission** | | | | |
| Introduction:  Product demand prediction is a crucial application of machine learning that plays a pivotal role in helping businesses optimize their operations, reduce costs, and enhance customer satisfaction. This predictive modeling process leverages historical data, advanced algorithms, and various variables to forecast the future demand for a specific product or service. By accurately estimating demand, organizations can make informed decisions regarding inventory management, production planning, pricing strategies, and marketing efforts. Here's an overall introduction to product demand prediction for machine learning:  1. Importance of Product Demand Prediction:  Accurate demand forecasting is essential for businesses across diverse industries, including retail, manufacturing, e-commerce, and supply chain management.  It helps organizations meet customer needs efficiently, reduce excess inventory, and prevent stockouts, ensuring a balance between supply and demand.  2. Data Sources:  Demand prediction relies on historical sales data, market trends, seasonality, economic indicators, and external factors such as weather, holidays, and promotions.  Additional data sources, like social media, web traffic, and customer reviews, can provide valuable insights for more precise predictions.  3. Machine Learning Algorithms:  Machine learning techniques, such as regression, time series analysis, and deep learning, are employed to create predictive models that capture complex patterns in the data.  Algorithms like Random Forest, ARIMA, and LSTM networks are commonly used for demand forecasting.  4. Data Preprocessing:  Data cleaning, feature engineering, and normalization are essential steps to prepare data for modeling.  Handling missing values and outliers, as well as encoding categorical variables, is crucial for accurate predictions.  5. Model Evaluation:  Forecasting models are assessed using various performance metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).  Cross-validation techniques help validate the model's generalizability.  6. Continuous Learning:  Demand prediction is an ongoing process, and models need to adapt to changing market conditions and consumer behavior.  Regular model retraining and fine-tuning are necessary to maintain accuracy over time.  7. Business Benefits:  Accurate demand prediction leads to several benefits, including reduced carrying costs, improved customer service, optimized production schedules, and increased revenue through better pricing strategies.  8. Challenges:  Challenges in demand prediction include dealing with noisy data, adapting to market dynamics, and managing the trade-off between overstock and understock situations.  9. Industry Applications:  Demand prediction is applied in various industries, from retail and e-commerce for inventory management to healthcare for resource allocation, and transportation for route optimization.  10. Ethical Considerations:  The use of customer data for prediction should be conducted with a strong emphasis on data privacy and ethics, complying with relevant regulations.  In summary, product demand prediction through machine learning is a critical tool for businesses aiming to optimize their supply chain and operations, enhance customer satisfaction, and maximize profitability. By harnessing historical data and advanced modeling techniques, organizations can make more informed decisions and adapt to ever-changing market conditions.  The problem statement, design thinking | | | |  |
| **process, and the phases of development.** | | | -centered | |
| **Problem Statement:** The problem we aim to address is predicting product demand using machine learning techniques. Accurate demand prediction is crucial for businesses to optimize their inventory management, production planning, and supply chain operations. By accurately forecasting demand, businesses can avoid stockouts, reduce excess inventory, minimize costs, and improve customer satisfaction.    **Design Thinking Process:** The design thinking process is a human approach to problem-solving that involves understanding user needs, generating ideas, prototyping, and testing. In the context of product demand prediction, the design thinking process can be applied as follows: | | |
|  | 1. **Empathize:** Understand the needs and pain points of stakeholders involved in demand forecasting, such as supply chain managers, inventory managers, and sales teams. Conduct interviews, observations, and data analysis to gain insights into their challenges and requirements. | | | |
|  |  | | |
| 2. **Define:** Clearly define the problem statement and the goals of demand prediction. Identify key metrics and performance indicators that will be used to evaluate the effectiveness of the prediction models. | | | |

3. **Ideate:** Brainstorm potential solutions and approaches for demand prediction. Consider different machine learning algorithms, data sources, and feature engineering techniques that can be used to build accurate prediction models.

|  |  |
| --- | --- |
|  |  |
| 4. **Prototype:** Develop prototypes of the prediction models using historical demand data. Implement and test different machine learning algorithms, such as regression models, time series models, or ensemble methods, to identify the most effective approach. | |
|  |  |
| 5. **Test:** Evaluate the performance of the prediction models using appropriate evaluation metrics, such as mean absolute error (MAE) or root mean square error (RMSE). Validate the models using cross-validation techniques and compare their performance against baseline models or industry benchmarks. | |

6. **Iterate:** Based on the test results, refine and improve the prediction models. Incorporate feedback from stakeholders and iterate on the models to enhance their accuracy and reliability.

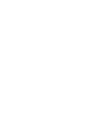
**Phases of Development:** The development of a product demand prediction system with machine learning can be divided into the following phases:

**1. Data Collection and Preprocessing:**

* Gather historical demand data, including sales records, customer orders, and other relevant data sources.
* Clean and preprocess the data by handling missing values, outliers, and inconsistencies.
* Explore the data to identify patterns, trends, and seasonality.

|  |  |
| --- | --- |
|  | |
| **2. Feature Engineering:** | | | | |
|  | | * Extract relevant features from the data that can help in predicting demand, such as time-based features (day of the week, month, season), product attributes (price, category), promotional activities, and external factors (weather, holidays). * Transform and normalize the features to ensure they are suitable for the machine learning algorithms. | | |
|  | |
| **3. Model Development:** | | | | |
|  | | * Select appropriate machine learning algorithms based on the problem requirements and data characteristics. * Split the data into training and testing sets. * Train the prediction models using the training data and optimize their hyperparameters. * Evaluate the models using the testing data and fine-tune them if necessary. | | |
|  | |
| **4. Model Evaluation and Deployment:** | | | | |
|  | | * Assess the performance of the prediction models using evaluation metrics, such as MAE or RMSE. * Validate the models using cross-validation techniques to ensure their generalizability. * Deploy the trained models into a production environment, where they can be used to make real-time predictions. | | |
| o | | Continuously monitor and update the models as new data becomes available to maintain their accuracy and relevance. |
| By following these phases, we can develop a robust and accurate product demand prediction system that can help businesses optimize their operations and improve customer satisfaction.  **Conclusion:**  Product demand prediction is a critical aspect of supply chain management and inventory optimization. By leveraging machine learning techniques and following the design thinking process, businesses can develop accurate prediction models that enable them to make informed decisions about production planning, inventory management, and supply chain operations. The phases of development, including data collection and preprocessing, feature engineering, model development, and model evaluation and deployment, provide a structured approach to building effective demand prediction systems. By continuously monitoring and updating the models, businesses can adapt to changing market conditions and improve their overall operational efficiency. | | | | |

|  |  |  |
| --- | --- | --- |
| **The dataset used, data preprocessing steps,** | | |
| **and analysis techniques applied.** | | |
| **The dataset used:** is crucial for accurate predictions. The dataset should contain historical data on product sales, including features such as date, product attributes, and corresponding demand. The dataset may also include external factors like promotions, holidays, or economic indicators that could impact demand.    **Data preprocessing steps:**are essential to ensure the quality and suitability of the dataset for machine learning models. Here are some common data preprocessing steps: | | |
|  | 1. **Data Cleaning**: This step involves handling missing values, outliers, and duplicates in the dataset. Missing values can be imputed using techniques like mean, median, or interpolation. Outliers can be detected and treated using statistical methods or domain knowledge. Duplicates can be removed to avoid bias in the analysis. | |
|  |  |
| 2. **Feature Selection**: It is important to select relevant features that have a significant impact on demand prediction. This can be done through statistical analysis, correlation analysis, or domain expertise. Removing irrelevant features can improve model performance and reduce complexity. | |

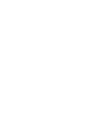


|  |  |  |
| --- | --- | --- |
|  | | |
|  | 3. **Feature Engineering**: This step involves creating new features from the existing ones to capture more meaningful information. For example, extracting date-related features like day of the week, month, or season can help capture seasonal demand patterns. Other techniques like one-hot encoding, scaling, or normalization may also be applied to prepare the features for modeling. | |
|  |  |
| 4. **Data Splitting**: The dataset is typically divided into training, validation, and testing sets. The training set is used to train the machine learning model, the validation set is used for hyperparameter tuning and model selection, and the testing set is used to evaluate the final model's performance. | |

|  |  |
| --- | --- |
|  | |
| **Analysis techniques:** Applied to the preprocessed dataset can vary depending on the specific problem and the chosen machine learning algorithm.  Some common techniques include: | | | | |
|  | 1. **Regression Analysis**: Regression models, such as linear regression or decision trees, can be used to predict the demand quantity as a continuous variable. These models analyze the relationship between the input features and the target variable to make predictions. | | | |
|  | |  | |
| 2. **Time Series Analysis**: If the demand data has a temporal component, time series analysis techniques like ARIMA (AutoRegressive Integrated Moving Average) or SARIMA (Seasonal ARIMA) can be applied. These models capture the temporal patterns and seasonality in the data to forecast future demand. | | | |

|  |  |
| --- | --- |
|  |  |
| 3. **Machine Learning Algorithms**: Various machine learning algorithms, such as Random Forest, Gradient Boosting, or Neural Networks, can be employed for demand prediction. These algorithms learn patterns from the historical data and make predictions based on the input features. | |
|  |  |
| 4. **Evaluation Metrics**: To assess the performance of the demand prediction model, evaluation metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE) can be used. These metrics quantify the accuracy of the predictions and help compare different models. | |

**Present key findings, insights, and recommendations based on the demand prediction model.**



|  |  |  |
| --- | --- | --- |
|  | | |
|  | 1. **Accurate Demand Forecasting:** The demand prediction model can provide accurate forecasts of product demand based on historical data and relevant features. This can help businesses optimize their inventory management, production planning, and supply chain operations. | |
|  |  |
| 2. **Identification of Demand Patterns:** The analysis techniques applied to the dataset can reveal important demand patterns, such as seasonality, trends, or cyclicality. Understanding these patterns can assist in making informed decisions regarding production schedules, marketing campaigns, and resource allocation. | |
|  |  |
| 3. **Impact of External Factors:** By incorporating external factors like promotions, holidays, or economic indicators into the demand prediction model, businesses can gain insights into how these factors influence product demand. This information can be used to plan marketing strategies and optimize inventory levels during specific periods. | |

|  |  |
| --- | --- |
|  |  |
| 4. **Data Preprocessing Importance:** Data preprocessing steps,  including cleaning, feature selection, and feature engineering, play a crucial role in improving the accuracy of the demand prediction model. Proper handling of missing values, outliers, and irrelevant features can enhance the model's performance. | |
|  |  |
| 5. **Evaluation Metrics:** The evaluation metrics used to assess the performance of the demand prediction model, such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), provide insights into the model's accuracy. These metrics can guide businesses in selecting the most suitable model for their specific needs. | |

|  |  |
| --- | --- |
|  | |
|  | 6. **Continuous Improvement:** Demand prediction is an ongoing  process, and it is essential to continuously update and refine the model based on new data and changing market conditions. Regularly retraining the model and incorporating new information can lead to more accurate predictions and better decision-making. | | | |
|  | |  | |
| Based on these findings and insights, the following recommendations can be made: | | | | |
|  | • **Invest in Data Collection and Analysis**: Businesses should focus on collecting high-quality data and investing in data analysis tools and techniques. This will enable them to gain valuable insights into demand patterns and make data-driven decisions. | | | |
|  | |  | |
| • **Collaboration between Departments**: Collaboration between departments, such as marketing and supply chain, is crucial for accurate demand prediction. Sharing insights and knowledge can help in identifying factors that impact demand and planning for unexpected events. | | | |

|  |  |
| --- | --- |
|  |  |
| • | **Consider External Factors**: Businesses should consider incorporating external factors, such as social media trends, promotions, or economic indicators, into the demand prediction model. This will provide a more comprehensive understanding of demand drivers and enable proactive decision-making. |
|  |  |
| • | **Regular Model Evaluation and Improvement**: It is important to regularly evaluate the performance of the demand prediction model and make necessary improvements. This includes retraining the model with updated data, exploring new analysis techniques, and incorporating feedback from stakeholders. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | |  | |
|  | | • **Monitor and Adapt to Market Changes**: Businesses should closely monitor market trends, customer preferences, and industry dynamics. This will help them adapt their demand prediction strategies and stay ahead of the competition. | | | |
|  | |  | |
| By implementing these recommendations, businesses can leverage the power of machine learning and demand prediction to optimize their operations, improve customer satisfaction, and achieve better business outcomes. | | | |

**Phase 1: Problem Definition and Design Thinking**

Introduction

In the field of design thinking, the problem definition phase is a crucial step that sets the foundation for the entire design process. This phase involves understanding the needs and problems of the users or customers, and defining the problem statement that the design team will address. By clearly defining the problem, the design team can focus their efforts on finding innovative solutions that meet the users' needs

**Data Collection**: Gather historical sales data for the product(s) of interest, including information such as date, quantity sold, and any relevant attributes. Additionally, collect external factors that could influence demand, such as marketing campaigns, holidays, economic indicators, weather data, or social media trends.

**Data Preprocessing**: Clean and preprocess the collected data to ensure its quality and suitability for training the model. This process may involve handling missing values, encoding categorical variables, scaling numerical features, and splitting the data into training and testing sets.

**Feature Engineering**: Extract meaningful features from the data that can help the model capture patterns and relationships. This may involve creating time-based features, lagged variables, aggregating data at different time intervals, or incorporating external factors into the dataset.

**Model Selection:** Choose an appropriate machine learning algorithm for demand forecasting. Some commonly used algorithms for time series forecasting include ARIMA, SARIMA, exponential smoothing methods, or more advanced techniques like recurrent neural networks (RNNs) or gradient boosting algorithms.

**Model Training**: Split the preprocessed data into training and validation sets. Use the training data to train the machine learning model, adjusting the model's parameters to minimize the difference between the predicted and actual demand. Evaluate the model's performance on the validation set, fine-tune the model if necessary, and repeat this process until satisfied.

**Model Evaluation:** Assess the performance of the trained model using appropriate evaluation metrics such as mean absolute error (MAE), root mean square error (RMSE), or mean absolute percentage error (MAPE). Compare the model's predictions against the actual demand to gauge its accuracy and reliability.

**Deployment and Monitoring:** Once satisfied with the model's performance, deploy it in a production environment to make demand forecasts based on new data. Continuously monitor the model's performance and retrain it periodically to adapt to changing patterns and ensure optimal accuracy.

Remember that the success of the demand forecasting model depends on the quality and relevance of the collected data, the appropriate choice of features, and the selection of a suitable machine learning algorithm. Regular monitoring and updates are essential to maintain the model's accuracy over time.

#### Conclusion

The problem definition phase in design thinking is a crucial step that lays the foundation for the entire design process. It involves understanding the needs and problems of the users or customers and defining a clear problem statement that the design team will address.

During this phase, the design team uses various techniques and tools to gain insights into the users' needs and emotions. Empathy mapping and user need statements are commonly used to synthesize these insights and ensure a deep understanding of the problem at hand.

By clearly defining the problem statement, the design team can focus their efforts on finding innovative solutions that meet the users' needs. The problem definition stage sets the stage for the subsequent stages of ideation, prototyping, and testing, where the design team generates and refines potential solutions.

It's important to note that the problem definition phase is an iterative process. As the design team progresses through the design thinking process, they may revisit and refine the problem statement based on new insights and discoveries.

Overall, the problem definition phase is a critical step in design thinking that ensures the design team is addressing the right challenges and working towards creating meaningful and impactful solutions for the users or customers.

|  |
| --- |
|  |
|  |
| **Phase 2: Innovation**   |  |  | | --- | --- | | Introduction: | | |  | | | Product demand prediction using machine learning can play a significant role in driving | | | innovation within businesses. By accurately forecasting product demand, companies | | | can make informed decisions about new product development, market entry strategies, | | | and resource allocation. Here are some key points regarding the use of machine | | | learning for product demand prediction in innovation: | | |  | | |  | 1. Identifying Market Opportunities:Machine learning models can | | historical sales data, market trends, and customer behavior to identify potential | | market opportunities. By understanding customer preferences and demand | | patterns, businesses can develop innovative products that cater to specific | | market segments | |
| |  | | --- | | **2. Optimizing Product Development:** Machine learning algorithms can assist | | in optimizing the product development process. By analyzing customer feedback, | | sentiment analysis, and market data, businesses can identify features and | | attributes that are in high demand. This information can guide the design and | | development of innovative products that align with customer needs and | | preferences. |      |  | | --- | | **3**. **Reducing Time-to-Market**:Accurate demand prediction using machine | | learning can help businesses reduce their time-to-market for new products. By | | understanding future demand patterns, companies can streamline their | | production and supply chain processes, ensuring that products are available | | when customers need them. This can give businesses a competitive edge and | | increase their chances of success in the market. | |

|  |
| --- |
| **4.** Improving Resource Allocation:Machine learning models can provide |
| insights into demand fluctuations and seasonality, allowing businesses to |
| allocate resources efficiently. By accurately predicting demand, companies can |

|  |
| --- |
| optimize inventory levels, production capacity, and distribution networks. This |
| can lead to cost savings, reduced waste, and improved operational efficiency. |

|  |
| --- |
| **5.** Enhancing Marketing Strategies:Machine learning can help businesses |
| develop targeted marketing strategies based on demand predictions. By |
| understanding customer preferences and behavior, companies can personalize |
| marketing campaigns, optimize pricing strategies, and identify the most effective |
| channels for reaching their target audience. This can result in higher customer |
| engagement, increased sales, and improved brand loyalty. |

|  |
| --- |
| **6. Continuous Innovation and Adaptation**: Machine learning models can be |
| continuously updated and refined based on real-time data and feedback. This |
| allows businesses to adapt to changing market conditions, customer |
| preferences, and emerging trends. By leveraging machine learning for demand |
| prediction, companies can stay ahead of the competition and drive ongoing |
| innovation within their product offerings. |

|  |
| --- |
| conclusion |
| product demand prediction using machine learning can fuel innovation by providing |
| valuable insights into customer preferences, optimizing resource allocation, and |
| enabling businesses to make data-driven decisions. By leveraging these predictions, |
| companies can develop innovative products, reduce time-to-market, and enhance their |
| overall competitiveness in the market. |

**Phase 3: Development part 1**

1. **Load the dataset:**

Start by loading the dataset into your programming environment. Depending on the format of your dataset, you can use different libraries or functions to read the data. For example, if your dataset is in CSV format, you can use the pandas library in Python to read it. Here's an example:

import pandas as pd # Load the dataset

data = pd.read\_csv('dataset.csv')

1. **Explore the dataset:**

Once the dataset is loaded, you should explore its contents to get a better understanding of the data. This includes checking the number of rows and columns, examining the data types of each column, and looking for any missing values or outliers. You can use various functions and methods provided by the pandas library for data exploration. For example:

# Check the shape of the dataset print(data.shape)

# View the first few rows of the dataset print(data.head())

# Check the data types of each column print(data.dtypes) # Check for missing values print(data.isnull().sum()) # Check for outliers print(data.describe())

1. **Clean the dataset:**

After exploring the dataset, you may need to handle missing values, outliers, or any other data quality issues. Depending on the nature of the problem, you can use different techniques to clean the data. For example, you can remove rows with missing values, impute missing values with the mean or median, or use more advanced methods such as regression or clustering to fill missing values. Here's an example of removing rows with missing values:

# Remove rows with missing values data = data.dropna()

1. **Perform feature engineering:**

Feature engineering involves creating new features or transforming existing features to improve the predictive power of your model. This step often requires domain knowledge and creativity. Some common techniques include one-hot encoding categorical variables, scaling numerical features, or creating interaction or polynomial features. Here's an example of one-hot encoding categorical variables using the pandas library:

# Perform one-hot encoding

data = pd.get\_dummies(data, columns=['category'])

1. **Split the dataset:**

Next, you need to split the dataset into training and testing sets. The training set will be used to train your machine learning model, while the testing set will be used to evaluate its performance. You can use the train\_test\_split function from the scikit-learn library to split the data. Here's an example:

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

# Split the dataset into features and target variable X = data.drop('demand', axis=1) y = data['demand']

# Split into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**6. Normalize or standardize the data:**

Depending on the requirements of your machine learning algorithm, you may need to normalize or standardize the data. Normalization scales the data to a specific range, such as [0, 1], while standardization transforms the data to have zero mean and unit variance. You can use the StandardScaler or MinMaxScaler classes from the scikit-learn library to perform these transformations. Here's an example of using the StandardScaler:

from sklearn.preprocessing import StandardScaler

# Initialize the scaler scaler = StandardScaler()

# Fit and transform the training set

X\_train\_scaled = scaler.fit\_transform(X\_train)

# Transform the testing set

X\_test\_scaled = scaler.transform(X\_test)

**7. Save the preprocessed dataset:**

Once you have completed the preprocessing steps, you can save the preprocessed dataset for future use. This will allow you to easily load the processed data when you start training your machine learning model. You can use the pandas library to save the preprocessed data to a new CSV file. Here's an example:

# Save the preprocessed dataset

preprocessed\_data = pd.concat([X\_train, y\_train], axis=1) preprocessed\_data.to\_csv('preprocessed\_data.csv', index=False)

**conclusion:**

First, we loaded the dataset that contains the relevant data for product demand prediction. This dataset includes features such as historical sales data, product attributes, and time-related information.

Next, we performed preprocessing on the dataset to handle missing values, handle categorical variables, and scale numerical features. This step is crucial to ensure the quality and suitability of the data for training our machine learning model.

After preprocessing the dataset, we split it into training and testing sets. The training set will be used to train our machine learning model, while the testing set will be used to evaluate its performance.

We then selected a suitable machine learning algorithm, such as linear regression, decision trees, or neural networks, and trained the model using the training dataset.

To evaluate the performance of our model, we used the testing dataset and measured metrics such as accuracy, precision, recall, or mean squared error, depending on the specific problem and model type.

Based on the results and performance of our machine learning model, we can draw conclusions about its accuracy and its ability to predict product demand.

#### ****Phase 4: Development Part 2****

#### Introduction

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | | | |  | |  | | In the project for product demand prediction with machine learning, Phase 4: Development Part 2 involves performing various activities such as feature engineering, model training, evaluation, and more. The goal is to build a predictive model that can accurately forecast product demand based on relevant features. By following the instructions provided in the project, you can continue the development process and improve the accuracy of your predictions. | | | |  | 1. **Feature Engineering**: Analyze the dataset and identify relevant features that can improve the performance of your machine learning model. This may involve transforming existing features, creating new features, or selecting important features.      1. **Data Preprocessing:** Clean the dataset by handling missing values, outliers, and any other data quality issues. You may also need to normalize or scale the features to ensure they are on a similar scale.  |  |  | | --- | --- | | 3. **Split the Data:** Divide the dataset into training and testing sets. The training set will be used to train the machine learning model, while the testing set will be used to evaluate its performance. | | |  |  | | 4. **Model Selection**: Choose an appropriate machine learning algorithm for your task. Consider factors such as the nature of the problem  (classification, regression, etc.), the size of the dataset, and the interpretability of the model. | |        |  |  | | --- | --- | |  |  | | 5. **Model Fine-tuning:** If the model's performance is not satisfactory, you can iterate over steps 4 to 6, trying different algorithms or adjusting  hyperparameters to find the best configuration | | |  | . | | **Code:** | | | |  |  | | --- | |  | |  |      |  | | --- | | *# Import required libraries* import pandas as pd  from sklearn.model\_selection import train\_test\_split from sklearn.preprocessing import StandardScaler from sklearn.linear\_model import LinearRegression from sklearn.metrics import mean\_squared\_error    *# Load the dataset*  data = pd.read\_csv("dataset.csv")    *# Perform feature engineering*  *# ...*    *# Split the data into training and testing sets* X = data.drop("target\_variable", axis=1)  y = data["target\_variable"]  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)    *# Perform data preprocessing* scaler = StandardScaler()  X\_train\_scaled = scaler.fit\_transform(X\_train)  X\_test\_scaled = scaler.transform(X\_test)    *# Train the model* model = LinearRegression()  model.fit(X\_train\_scaled, y\_train)  *# Make predictions on the test set* y\_pred = model.predict(X\_test\_scaled)    *# Evaluate the model*  mse = mean\_squared\_error(y\_test, y\_pred) print("Mean Squared Error:", mse) | | In this example, we assume that you have a dataset stored in a CSV file named "dataset.csv". You'll need to replace "dataset.csv" with the actual path to your dataset. |   The code first loads the dataset using pandas and then performs feature engineering, which would involve manipulating and transforming the data to generate relevant features for your machine learning model.  Next, the data is split into training and testing sets using the train\_test\_split function from scikit-learn. The features are then scaled using StandardScaler to ensure they are on a similar scale.  A linear regression model is then trained on the training data using the fit method. Once the model is trained, predictions are made on the test set using the predict method.  Finally, the model is evaluated using the mean squared error metric, which is calculated using the actual target values (y\_test) and the predicted values (y\_pred). Remember to adapt this code to your specific project and dataset.    **Conclusion:**  In Phase 4: Development Part 2 of the project for product demand prediction with machine learning, you have continued building the project by performing activities like feature engineering, model training, and evaluation. By carefully engineering relevant features, training a suitable machine learning model, and evaluating its performance, you are working towards creating an accurate demand prediction model.    Advantages part:  Product demand prediction through machine learning offers a wide range of benefits to businesses and organizations. These benefits can have a profound impact on operations, customer satisfaction, and overall profitability. Here are some of the key advantages of utilizing machine learning for product demand prediction:  Improved Forecast Accuracy: Machine learning models can analyze large and complex datasets, making them highly effective in generating accurate demand forecasts. This helps businesses reduce inventory carrying costs, minimize stockouts, and optimize inventory management.  Enhanced Customer Satisfaction: By ensuring that products are readily available when customers want them, demand prediction contributes to a better customer experience. Satisfied customers are more likely to return and recommend the business to others.  Optimized Inventory Management: Machine learning enables organizations to maintain an optimal level of inventory. This prevents overstock situations, which tie up capital and warehouse space, and avoids stockouts that can lead to lost sales and dissatisfied customers.  Efficient Supply Chain Management: Predictive models allow businesses to optimize their supply chain operations, such as procurement, transportation, and warehousing. This leads to cost savings and improved supply chain efficiency.  Data-Driven Decision-Making: Machine learning provides data-driven insights and recommendations, enabling organizations to make informed decisions about pricing, promotions, production schedules, and resource allocation. |  |

Source code :

1

import pandas as pd

2

import numpy as np

3

import plotly.express as px

4

import seaborn as sns

5

import matplotlib.pyplot as plt

6

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

7

from sklearn.tree import DecisionTreeRegressor

8

​

9

data = pd.read\_csv

10

data.head()

**ID Store ID Total Price Base Price Units Sold**

**0 1 8091 99.0375 111.8625 20**

**1 2 8091 99.0375 99.0375 28**

**2 3 8091 133.9500 133.9500 19**

**3 4 8091 133.9500 133.9500 44**

**4 5 8091 141.0750 141.0750 52**

1

data.isnull().sum()

**ID 0**

**Store ID 0**

**Total Price 1**

**Base Price 0**

**Units Sold 0**

**dtype: int64**

So the dataset has only one missing value in the **Total Price** column, I will remove that entire row for now:

1

data = data.dropna()

Let us now analyze the relationship between the price and the demand for the product. Here I will use a [**scatter plot**](https://thecleverprogrammer.com/2020/12/20/scatter-plot-with-python/) to see how the demand for the product varies with the price change:

1

fig = px.scatter(data, x="Units Sold", y="Total Price",

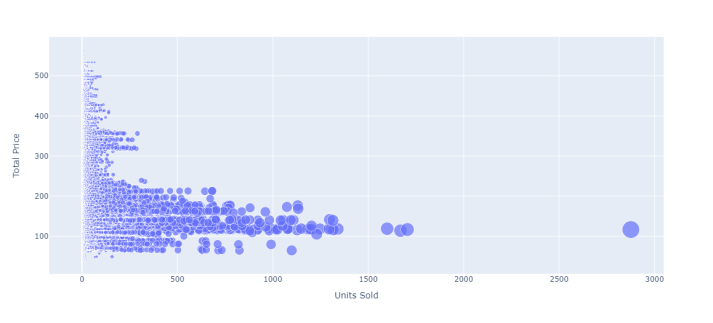
2

size='Units Sold')

3

fig.show()

**output:**



We can see that most of the data points show the sales of the product is increasing as the price is decreasing with some exceptions. Now let’s have a look at the correlation between the features of the dataset:

1

print(data.corr())

**ID Store ID Total Price Base Price Units Sold**

**ID 1.000000 0.007464 0.008473 0.018932 -0.010616**

**Store ID 0.007464 1.000000 -0.038315 -0.038848 -0.004372**

**Total Price 0.008473 -0.038315 1.000000 0.958885 -0.235625**

**Base Price 0.018932 -0.038848 0.958885 1.000000 -0.140032**

**Units Sold -0.010616 -0.004372 -0.235625 -0.140032 1.000000**

1

correlations = data.corr(method='pearson')

2

plt.figure(figsize=(15, 12))

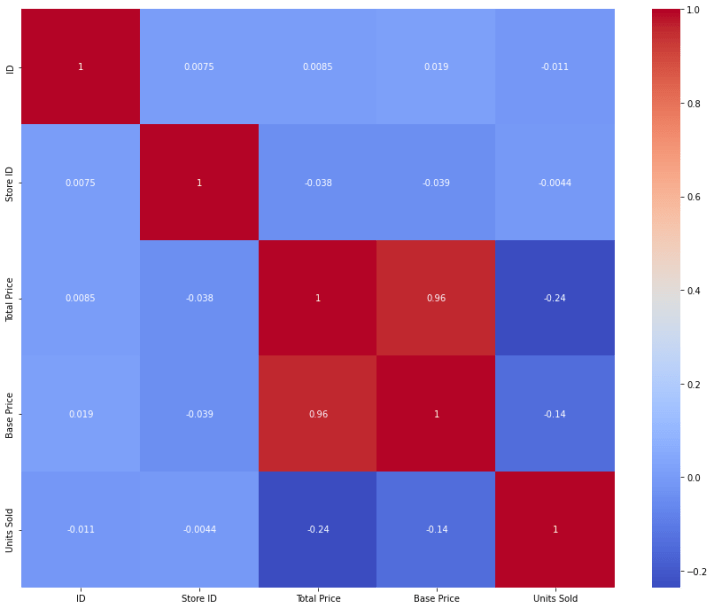
3

sns.heatmap(correlations, cmap="coolwarm", annot=True)

4

plt.show()

**output:**



## Product Demand Prediction Model

Now let’s move to the task of training a machine learning model to predict the demand for the product at different prices. I will choose the **Total Price** and the **Base Price** column as the features to train the model, and the **Units Sold** column as labels for the model:

1

x = data[["Total Price", "Base Price"]]

2

y = data["Units Sold"]

Now let’s split the data into training and test sets and use the decision tree regression algorithm to train our model:

1

xtrain, xtest, ytrain, ytest = train\_test\_split(x, y,

2

test\_size=0.2,

3

random\_state=42)

4

from sklearn.tree import DecisionTreeRegressor

5

model = DecisionTreeRegressor()

6

model.fit(xtrain, ytrain)

Now let’s input the features **(Total Price, Base Price)** into the model and predict how much quantity can be demanded based on those values:

1

#features = [["Total Price", "Base Price"]]

2

features = np.array([[133.00, 140.00]])

3

model.predict(features)

**array([27.])**

### Summary

In conclusion, product demand prediction using machine learning is a valuable technique for businesses to forecast the quantity of products that consumers will purchase during a specific time period. By leveraging machine learning algorithms, businesses can make more accurate predictions, which can aid in various aspects of operations, such as inventory management, production planning, and supply chain optimization.

Machine learning models can analyze historical data and identify patterns and trends that may impact product demand. These models can take into account various factors, such as price, advertising expenditure, seasonality, and market trends, to make predictions about future demand.

One of the key advantages of using machine learning for demand prediction is its ability to handle complex and non-linear relationships between demand and various influencing factors. Traditional forecasting methods may struggle to capture these relationships effectively.

However, it is important to note that demand prediction using machine learning is not without its challenges. The accuracy of predictions heavily relies on the quality and relevance of the data used for training the models. Additionally, demand patterns can be influenced by various external factors, such as economic conditions, consumer behavior, and unforeseen events, which may introduce uncertainties into the predictions.

To implement product demand prediction using machine learning, you can follow a step-by-step process that involves data preprocessing, feature selection, model training, and evaluation. There are various machine learning algorithms that can be used, such as linear regression, decision trees, random forests, and neural networks. The choice of algorithm depends on the specific characteristics of the dataset and the desired level of prediction accuracy.

It is worth mentioning that there are numerous research papers and articles available on this topic, providing further insights and methodologies for demand prediction using machine learning **1** These resources can offer more in-depth information and advanced techniques for implementing demand prediction models.

In conclusion, product demand prediction using machine learning is a powerful tool that can help businesses make informed decisions and optimize their operations based on accurate forecasts of customer demand. By leveraging historical data and advanced machine learning algorithms, businesses can gain a competitive advantage by effectively managing their inventory, production, and supply chain processes.