**Product Demand Prediction with machine learning**

**PHASE-5 SUBMISSION DOCUMENT**

**Project Title: product demand prediction**

**Phase 5:** Project Documentation & Submission

**Topic:** In this section we will document the complete

project and prepare it for submission.

**Introduction:**

* The research aimed to analyze and predict the demand for products, which is critical aspect for business to effectively manage.
* The project utilized various machine learning techniques, including time series analysis, feature engineering and model training to develop an accurate and reliable forecasting model.
* Product demand prediction is the critical aspect of supply chain management and inventory optimization for business in various industry.
* **In today dynamic marketing the ability to accurately forecast demand for product is essential to maintain efficient operations reduce the cast and satisfies the customer need.**

This introduction provides an overview of key concept and process involved in using machine learning for product demand prediction.

**Data collection and preprocessing:**

* Successful demand prediction relies on quality data.
* This involves gathering historical sales data, customer behavior and other relevant information.

**Data preprocessing:**

* Data preprocessing including cleaning normalization and feature engineering is essential to ensure data is suitable for modeling.

**Model selection:**

* Machine learning models such as linear regression decision tree random forest, and neural networks can be used for demand prediction.

**Training and testing:**

* The selected model is trained on historical data to learn pattern and relationships.
* This is followed by testing the model on a separate dataset to assess its performance.

**Evaluation Metrics:**

* Common metrics for evaluating the model performance includes mean absolute error (MAE), Mean square error (MSE) and Root Mean Squared Error (RMSE).

**Dataset Link:**

<https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning>

**GIVEN DATASET:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | STORE ID | TOTAL  PRICE | BASE  PRICE | UNITS  SOLD |
| 1 | 8091 | 99.0375 | 111.8625 | 20 |
| 2 | 8091 | 99.0375 | 99.0375 | 28 |
| 3 | 8091 | 133.95 | 133.95 | 19 |
| 4 | 8091 | 133.95 | 133.95 | 44 |
| 5 | 8091 | 141.075 | 141.075 | 52 |
| 9 | 8091 | 227.2875 | 227.2875 | 18 |
| 10 | 8091 | 327.0375 | 327.0375 | 47 |
| 13 | 8091 | 210.0 | 210.9 | 50 |
| 14 | 8091 | 234.4125 | 234.4125 | 82 |
| 17 | 8095 | 99.0375 | 327.00375 | 99 |
| 19 | 8095 | 97.6125 | 210.9 | 120 |
| 22 | 8095 | 98.325 | 234.4125 | 40 |
| 23 | 8095 | 133.2375 | 99.0375 | 60 |
| …. | … | … | …. | … |
| 1528 | 1335 | 9823 | 183.1125 | 92 |
| 1529 | 1337 | 9832 | 142.5 | 107 |
| 1530 | 1340 | 9832 | 426.4125 | 19 |

Here's a list of tools and software commonly used in the

process:

1. **Programming Language:**

* Python is the most popular language for machine learning due to its extensive libraries and frameworks. You can use libraries like NumPy, pandas, scikit-learn, and more.

**2. Integrated Development Environment (IDE):**

* Choose an IDE for coding and running machine learning

experiments. Some popular options include Jupyter Notebook, Google Colab or traditional IDEs like PyCharm.

**3.Machine Learning Libraries:**

* You'll need various machine learning libraries, including:
* scikit-learn for building and evaluating machine learning models.
* TensorFlow or PyTorch for deep learning, if needed.
* XGBoost, LightGBM, or CatBoost for gradient boosting models.

**4. Data Visualization Tools:**

* Tools like Matplotlib, Seaborn, or Plotly are essential for data exploration and visualization.sss

**5. Data Preprocessing Tools:**

* Libraries like pandas help with data cleaning, manipulation, and preprocessing.

**6. Data Collection and Storage:**

* Depending on your data source, you might need web scraping tools (e.g., BeautifulSoup or Scrapy) or databases (e.g., SQLite, PostgreSQL) for data storage.

**7. Version Control:**

* Version control systems like Git are valuable for tracking

changes in your code and collaborating with others.

**8. Notebooks and Documentation:**

* Tools for documenting your work, such as Jupyter Notebooks or Markdown for creating README files and documentation.

**9. Hyperparameter Tuning:**

* Tools like GridSearchCV or RandomizedSearchCV from

scikit-learn can help with hyperparameter tuning.

**10. Web Development Tools (for Deployment):**

* If you plan to create a web application for model deployment ,knowledge of web development tools like Flask or Django for backend development, and HTML, CSS, and JavaScript for the front-end can be useful.

**11. Cloud Services (for Scalability):**

* For large-scale applications, cloud platforms like AWS, Google Cloud, or Azure can provide scalable computing and storage resources.

**12. External Data Sources (if applicable):**

* Depending on your project's scope, you might require tools to access external data sources, such as APIs or data scraping tools.

**1.DESIGN THINKING AND PRESENT IN FORM OF**

**DOCUMENT**

**1.Define:**

* Clearly articulate the problem statement, such as "How might we Predict product demand more accurately and transparently using machine learning?"
* Identify the key goals and success criteria for the project, such as increasing prediction accuracy, reducing bias, or improving user trust in the valuation process.

**2.Ideate:**

* Brainstorm creative solutions and data sources that can enhance the accuracy and transparency of product demand prediction.
* Encourage interdisciplinary collaboration to generate a wide range of ideas, including the use of alternative data, new algorithms, or improved visualization techniques.

**3.Prototype:**

* Create prototype machine learning models based on the ideas generated during the ideation phase.
* Test and iterate on these prototypes to determine which approaches are most promising in terms of accuracy and usability.

**4.Test:**

* Gather feedback from users and stakeholders by testing the machine learning models with real-world data and scenarios.
* Assess how well the models meet the defined goals and success criteria, and make adjustments based on user feedback.

**5.Implement:**

* Develop a production-ready machine learning solution for product demand prediction, integrating the best-performing algorithms and data sources.
* Implement transparency measures, such as model interpretability tools, to ensure users understand how predictions are generated.

**6.Evaluate:**

* Continuously monitor the performance of the machine learning model after implementation to ensure it remains accurate and relevant in a changing real estate market.

**7.Scale and Deploy:**

* Once the machine learning model has been optimized and validated deploy it at scale to serve a broader audience, such as real estate Total price, base price, and units sold.
* Ensure the model is accessible through user-friendly interfaces and integrates seamlessly into real estate workflows.

**8.Educate and Train:**

* Provide training and educational resources to help users understand how the machine learning model works, what factors it considers and its limitations.
* Foster a culture of data literacy among stakeholders to enhance trust in the technology.

**2.DESIGN INTO INNOVATION**

**1.Data Collection:**

Gather a comprehensive dataset that includes features such as id store id total prices base price and unit sold and other relevant variables.

1. **Data Preprocessing:**

Clean the data by handling missing values, outliers, and

encoding categorical variables. Standardize or normalize numerical features as necessary.

**PYTHON PROGRAM:**

Fromsklearnimportdatasets  
fromsklearn.model\_selectionimporttrain\_test\_split  
fromsklearn.metricsimportaccuracy\_score  
from sklearn .tree import Decision Tree Classifier

data=datasets.load\_wine(as\_frame=True)  
X=data.data  
y = data . target

X \_ train, X \_ test, y \_ train, y \_ test = train \_ test \_ split(X, y, test \_ size = 0.25, random \_ state = 22)

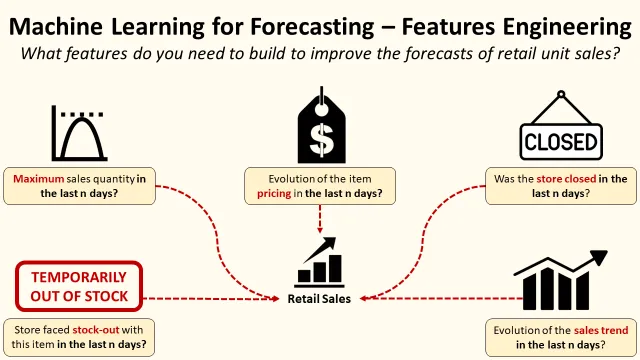
Dtree=DecisionTreeClassifier(random\_state= 22)  
d tree . fit(X \_ train , y \_ train)

**Output:**

Decision Tree Classifier(random \_ state=22)

**Feature Engineering:**

* Feature engineering is the process of transforming raw data into features that are suitable for machine learning models.
* In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.



**Model Training:**

* The process of selecting the best algorithm and model architecture for a specific job or dataset.
* It entails assessing and contrasting various models to identify the one that best fits the data & produces the best results.

**Test and Training:**

* Train/Test is a method to measure the accuracy of your model.
* It is called Train/Test because you split the data set into two sets: a training set and a testing set. 80% for training, and 20% for testing.
* You train the model using the training set.

**Monitoring and Maintenance:**

* Regularly monitor the model's performance in the real world and update it as needed.

**Innovation:**

* Consider innovative approaches such as using satellite imagery or IoT data for real-time property condition monitoring, or integrating natural language processing for textual property descriptions.

**3.BUILD LOADING AND PREPROCESSING THE**

**DATA**

**1.Data Collection:**

Obtain a dataset that contains information about product and

their corresponding prices. This dataset can be obtained from sources like real estate websites, company records, or other reliable data providers.

**2.Load the Dataset:**

* Import relevant libraries, such as pandas for data manipulation and

numpy for numerical operations.

* Load the dataset into a pandas Data Frame for easy data handling.

You can use pd.read\_csv() for CSV files or other appropriate

functions for different file formats.

**Program:**

import pandas as pd

import numpy as np

import seaborn a sns

import matplotlib .pyplot as plt

from Sklearn . model \_ selection import train \_ test \_split

from sklearn. preprocessing import Standard Scaler

from sklearn. metrics import r2\_score,

mean \_ absolute \_ error , mean \_ squared \_error

from sklearn. linear \_model import LinearRegression

from sklearn. linear \_model import Lasso

from sklearn. ensemble import RandomForestRegressor

from sklearn.svm import SVR

import xgboost as xg

%matplotlib inline

import warnings

warning . filter warning ("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/\_\_init\_\_.py:146:

User Warning: A NumPy version >=1.16.5 and <1.23.0 is required for

this version of SciPy (detected version 1.23.5

warning. warn (f"A NumPy version > = {np \_ min version} and

< {np \_ max version}"

LOADING DATASET:

dataset = pd. read \_ csv('E:/product\_demand.csv')

OUTPUT :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | STORE ID | TOTAL  PRICE | BASE  PRICE | UNITS  SOLD |
| 1 | 8091 | 99.0375 | 111.8625 | 20 |
| 2 | 8091 | 99.0375 | 99.0375 | 28 |
| 3 | 8091 | 133.95 | 133.95 | 19 |
| 4 | 8091 | 133.95 | 133.95 | 44 |
| 5 | 8091 | 141.075 | 141.075 | 52 |
| 9 | 8091 | 227.2875 | 227.2875 | 18 |
| 10 | 8091 | 327.0375 | 327.0375 | 47 |
| 13 | 8091 | 210.0 | 210.9 | 50 |
| 14 | 8091 | 234.4125 | 234.4125 | 82 |
| 17 | 8095 | 99.0375 | 327.00375 | 99 |
| 19 | 8095 | 97.6125 | 210.9 | 120 |
| 22 | 8095 | 98.325 | 234.4125 | 40 |
| 23 | 8095 | 133.2375 | 99.0375 | 60 |
| …… | ….. | ….. | …… | ….. |

**Data Exploration:**

* Explore the dataset to understand its structure and contents.
* Check for the presence of missing values, outliers, and data types of each feature.

**Data Cleaning:**

* Handle missing values by either removing rows with missing data or imputing values based on the nature of the data.

**1.Split the Data:**

* Split your dataset into training and testing sets. This helps you evaluate your model's performance later.

**Program:**

X = df. drop('price', axis=1) # Features

y = df ['price'] # Target variable

X \_train, X \_test, y \_train, y \_test = train \_test \_split (X, y, test \_size=0.2,

Random \_state=42)

**2.Feature Scaling:**

* Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and std=1) is a common choice.

**Program:**

scaler =Standard Scaler ()

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

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

**Train-Test Split:**

Split the dataset into training and testing sets to evaluate the

machine learning model's performance.

**Program:**

X = df.drop('price', axis=1) # Features

y = df['price'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size=0.2, random\_state=42)

**4. PERFORMING DIFFERENT ACTIVITIES LIKE**

**FEATURE ENGINEERING, MODEL TRAINING**

**EVALUATION etc.,**

**Feature Engineering:**

* Feature engineering is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modelling.
* Feature engineering in machine learning aims to improve the performance of models.

**Program:**

Import pandas as pd

Import numpy as np

Import plotly.express as px

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.tree import DecisionTreeRegressor

data=pd.read\_csv(“[www.Kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning](http://www.Kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning)”)

data.head()

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | STORE ID | TOTAL  PRICE | BASE  PRICE | UNITS  SOLD |
| 1 | 8091 | 99.0375 | 111.8625 | 20 |
| 2 | 8091 | 99.0375 | 99.0375 | 28 |
| 3 | 8091 | 133.95 | 133.95 | 19 |
| 4 | 8091 | 133.95 | 133.95 | 44 |
| 5 | 8091 | 141.075 | 141.075 | 52 |

**Data Preprocessing & Visualization:**

Continue data preprocessing by handling any remaining

missing values or outliers based on insights from your data exploration.

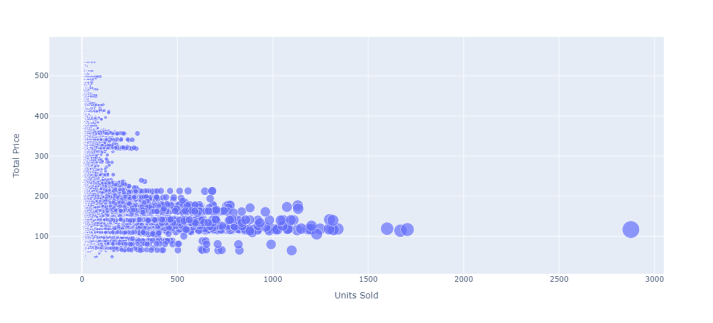
**Visualization and Pre-Processing of Data:**

IN[1]:

fig = px.scatter(data, x="Units Sold", y="Total Price", size='Units Sold’)

fig.show()

OUT[1]:



IN[2]:

print(data.corr())

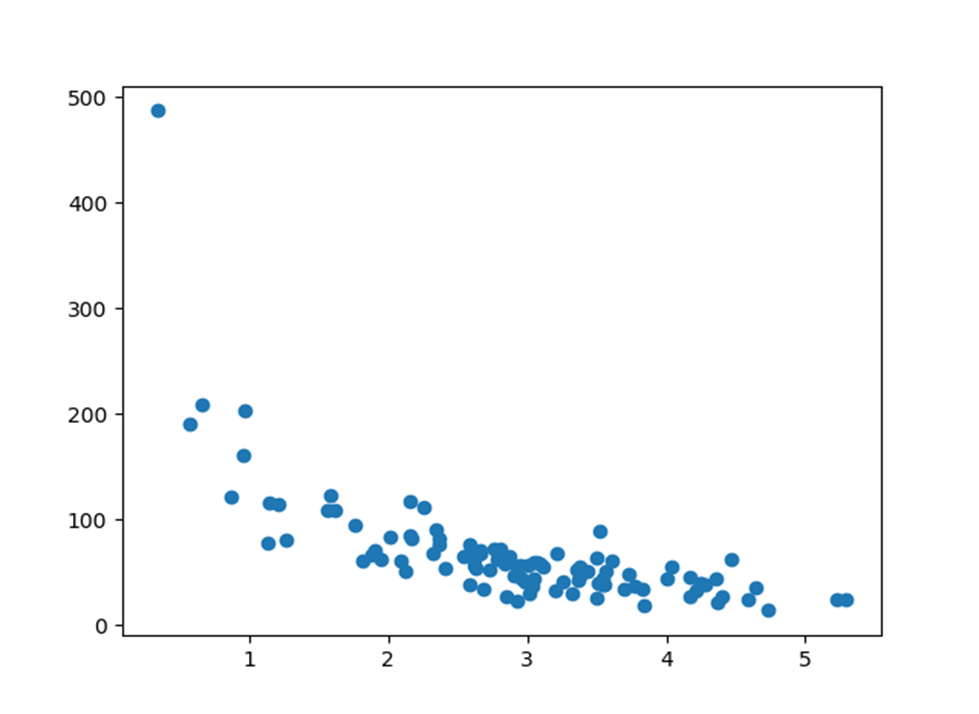
OUT[2]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ID | Store ID | Total price | Base price | Units Sold |
| ID | 1.000000 | 0.007464 | 0.008473 | 0.01892 | 0.1061 |
| Store ID | 0.007464 | 1.000000 | -0.03831 | 0.03884 | 0.0043 |
| Total price | 0.008473 | -0.03831 | 1.000000 | 0.95888 | 0.2356 |
| Base price | 0.018932 | -0.03884 | 0.958885 | 1.00000 | 0.1400 |
| Units Sold | -0.01061 | -0.00437 | -0.23562 | 0.140032 | 1.0000 |

IN[3]:

importnumpy  
importmatplotlib.pyplotasplt  
numpy.random.seed(2)  
  
x=numpy.random.normal(3,1,100)  
y=numpy.random.normal(150,40,100)/x  
  
plt.scatter(x,y)  
plt.show()

OUT[3]:



**Model Evaluation:**

* Evaluate your model's performance using appropriate regression
* metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

**Program:**

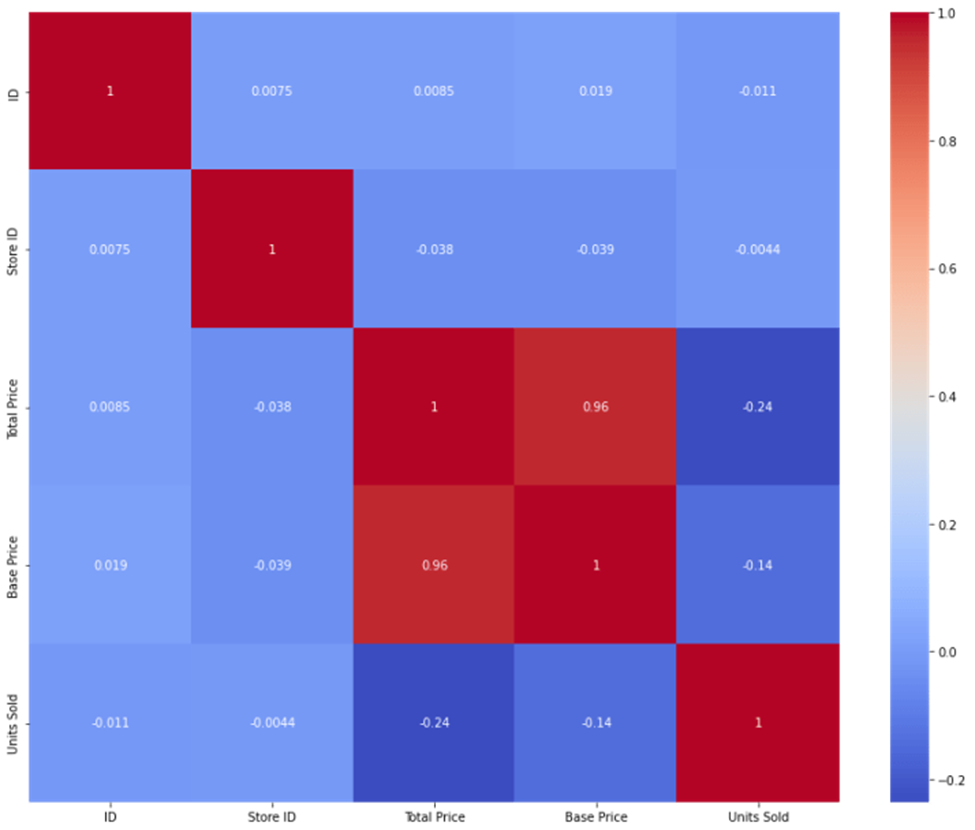
correlations = data.corr(method='pearson')

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

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

plt.show()

**output:**



**ADVANTAGE:**

Product demand prediction using machine learning offers several significant advantages:

1. **Accuracy :**

* Machine learning models can process and analyze vast amounts of data, including various property and market factors.
* This results in more accurate product demand predictions compared to traditional methods that rely on a limited set of variables

1. **Complex Data Handling:**

* They can recognize patterns and interactions

among different features, allowing for a more comprehensive assessment of a property's value.

1. **Continuous Learning:**

* Machine learning models can be continually updated with new data, enabling them to adapt to changing market conditions and trends.
* This ensures that predictions remain relevant and up-to-date.

**4.Efficiency:**

* Automated valuation models powered by machine learning can evaluate properties rapidly.
* This efficiency is beneficial for both property appraisers and individuals looking to determine the value of a property quickly.

**5. Data Integration:**

* Machine learning models can incorporate a wide range of data sources, including property characteristics, neighborhood information economic indicators, and even social trends.
* This holistic approach provides a more complete picture of the factors influencing product prices.

**DISADVANTAGES:**

* While product demand prediction using machine learning offers numerous advantages, it also comes with several disadvantages and challenges:

**1.Data Quality:**

* Machine learning models heavily rely on data quality.
* Inaccurate or incomplete data can lead to unreliable predictions.
* Ensuring the data used for training and evaluation is of high quality is essential.

**2. Overfitting:**

* Machine learning models can be prone to overfitting, where they perform exceptionally well on the training data but struggle with new, unseen data.
* This can result in overly optimistic or inaccurate predictions.

**3.Data Privacy and Security:**

* Handling sensitive property and financial data for training

models raises privacy and security concerns.

* Protecting this information from unauthorized access and breaches is critical.

**4.Model Interpretability:**

* Some machine learning models, such as deep neural networks can be challenging to interpret.
* Understanding why a model makes a specific prediction is crucial for trust and accountability.

**5.Bias and Fairness:**

* Machine learning models can inherit biases present in the

training data, potentially leading to unfair or discriminatory predictions especially in areas where historical biases exist in real estate practices.

**CONCLUSION:**

* It helps businesses make informed decisions that affect everything from inventory planning to supply chain optimization.
* With customer expectations changing faster than ever, businesses need a method to forecast demand accurately.
* In the quest to build a product demand prediction model, we have embarked on a critical journey that begins with loading and preprocessing the dataset.
* We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis.