**PRODUCT DEMAND PREDICTION WITH MACHINE LEARNINGS**

**Phase 4: Development Part 2**

**Dataset Link:**[**https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning**](https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning)

**FEATURE ENGINEERING:**

Feature engineering is the process of creating new features or transforming existing features in a dataset to improve the performance of machine learning models. It involves selecting, enhancing, or modifying the input variables (features) used in a model to make them more informative or suitable for the task at hand. Effective feature engineering can lead to better model accuracy and generalization. Here are some key aspects of feature engineering:

**1.Feature Selection**:

Sometimes, datasets contain many features, some of which may be irrelevant or redundant. Feature selection involves choosing the most relevant features while discarding less important ones. Common techniques for feature selection include statistical tests, correlation analysis, and domain knowledge.

**2.Feature Transformation:**

Feature transformation involves changing the representation of features to make them more suitable for a given machine learning algorithm or problem. Common transformations include.

**MODEL TRAINING:**

Model training is a crucial step in the development of machine learning models. It involves using a dataset to teach a model to make predictions or classify data based on patterns and relationships it learns from the data. The primary goal of model training is to optimize the model's parameters or weights to minimize its prediction errors and improve its performance. Here are the key steps and concepts involved in model training.

**1.Data Preparation**:

Before training a model, you need to prepare the data. This includes data preprocessing steps such as cleaning, transformation, and feature engineering. The dataset is typically split into three parts: a training set, a validation set, and a test set.

**2.Selecting a Model:**

You need to choose an appropriate machine learning algorithm or model architecture based on the nature of your problem (classification, regression, clustering, etc.) and the characteristics of your data. Different algorithms have different strengths and weaknesses.

**3.Initialization:**

Initialize the model's parameters or weights with some initial values. The choice of initialization can affect the convergence of the training process.

**4.Loss Function:**

Define a loss function (also known as a cost function or objective function) that quantifies how well the model's predictions match the actual target values in the training data. Common loss functions include mean squared error for regression problems and cross-entropy for classification problems.

**5.Optimization Algorithm:**

Choose an optimization algorithm, such as gradient descent, stochastic gradient descent (SGD), or variants like Adam or RMSprop. The optimization algorithm iteratively updates the model's parameters to minimize the loss function.

**6.Training Loop:**

The model is trained through an iterative process in which it makes predictions on the training data, calculates the loss, and updates its parameters using the optimization algorithm. This process is repeated for a specified number of iterations (epochs) or until convergence.

**7.Validation**:

After each training epoch or a certain number of iterations, the model's performance is evaluated on the validation set using appropriate evaluation metrics. This helps monitor the model's progress and detect overfitting.

**8.Hyperparameter Tuning:** Fine-tune the hyperparameters of the model, such as learning rate, batch size After each training epoch or a certain number of iterations, the model's performance is evaluated on the validation set using appropriate evaluation metrics. This helps monitor the model's progress and detect overfitting.

, and regularization strength, to optimize its performance on the validation set. This is typically done through grid search, random search, or more advanced techniques like Bayesian optimization.

**9.Early Stopping:**

Implement early stopping to prevent overfitting. If the model's performance on the validation set starts to degrade, training can be stopped to avoid training the model too long and fitting noise in the data.

**10.Test Evaluation:**

Once training is complete, the final model is evaluated on the test set to assess its generalization performance. This step provides an estimate of how well the model will perform on new, unseen data.

**11.Model Deployment:**

If the model meets the desired performance criteria, it can be deployed for real-world use. Deployment involves integrating the model into an application or system where it can make predictions on new data.

**12.Monitoring and Maintenance:**

After deployment, it's essential to monitor the model's performance in production and periodically retrain it with fresh data to ensure it remains accurate and up-to-date.

Model training is an iterative and resource-intensive process that requires careful attention to data quality, algorithm selection, and hyperparameter tuning. It is a fundamental step in creating machine learning solutions that can make accurate predictions or decisions based on data.

**PYTHON PROGRAM:**

import pandas as pd

import numpy as np

import plotly.express as px

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.tree import DecisionTreeRegressor

file = pd.read\_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/demand.csv")

data = file

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 |

data.isnull()

data.dropna()

#

data.isna().sum()

# ID 0

Store ID 0

Total Price 1

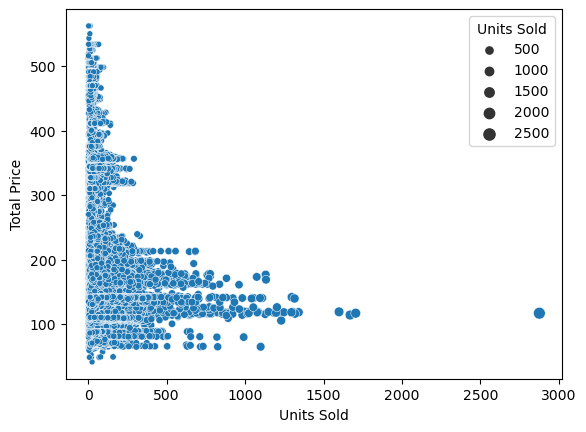
Base Price 0

Units Sold 0

dtype: int64

sns.scatterplot(data, x='Units Sold', y='Total Price',size= 'Units Sold')

# <Axes: xlabel='Units Sold', ylabel='Total Price'>

Out put:

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 |

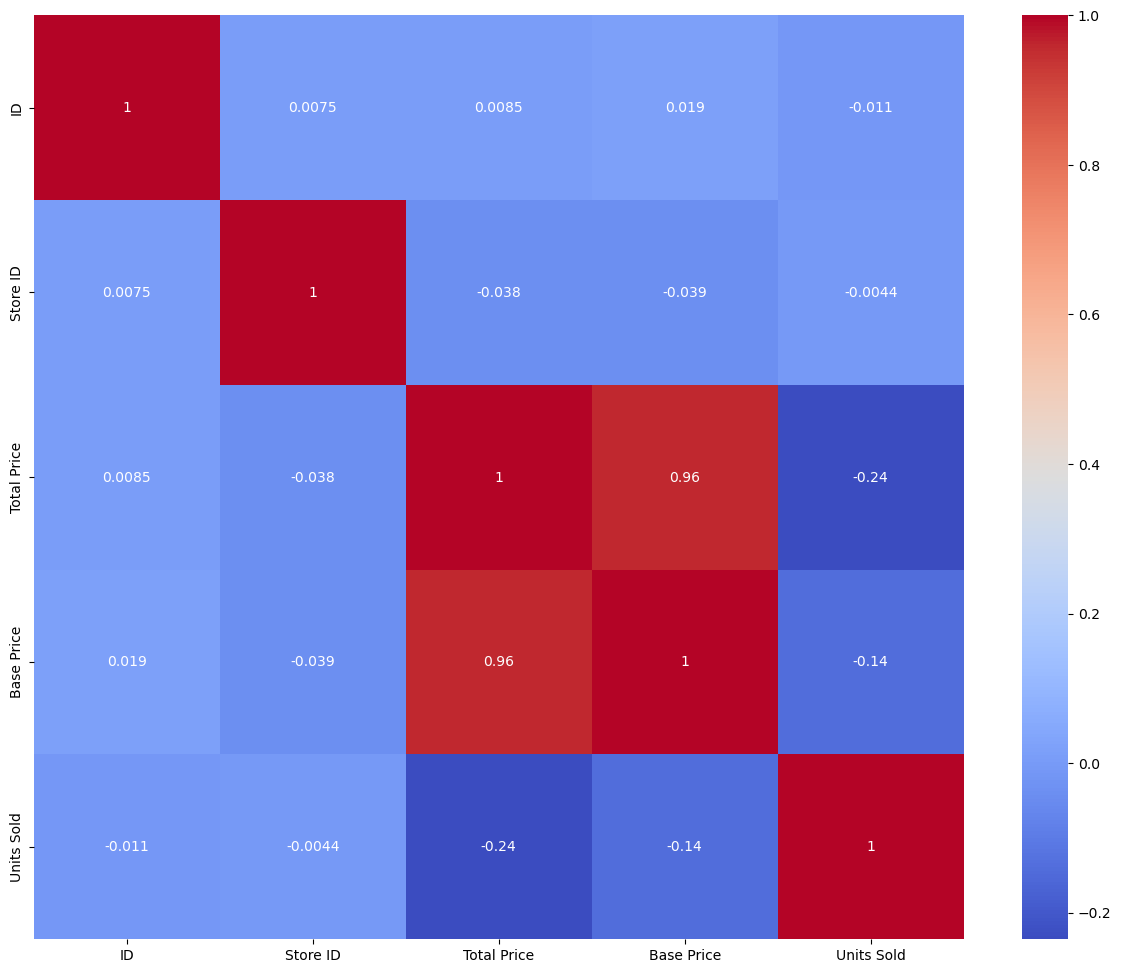
correlations = data.corr(method='pearson')

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

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

plt.show()

out put:



**EVALUATION:**

Evaluation in the context of machine learning and data analysis refers to the process of assessing the performance and quality of a model or a system based on specific criteria and metrics. It is a critical step to determine how well a model or system is performing and whether it meets the intended goals. Here are key aspects of evaluation.