Product Demand Prediction with Machine Learnings

PROJECT TITLE: Product Demand Prediction

**Phase 3:**Development Part-1



Introduction:

Predicting product demand with machine learning is a valuable application that can help businesses optimize their inventory management, production planning, and sales forecasting. To develop a product demand prediction model, you can follow these general steps:

**1. Data Collection:** Gather historical data on product sales, including factors that might influence demand, such as price, promotions, seasonality, and economic indicators.

**2.Data Preprocessing:** Clean and preprocess the data. This may involve handling missing values, outliers, and converting categorical data into numerical formats.

**3. Feature Engineering:** Create relevant features that can help the model better understand demand patterns. For example, you can create lag features to capture historical sales data or engineer features related to promotions or holidays.

**4.Data Splitting:** Split your data into training and testing datasets to evaluate the model's performance accurately. Cross-validation is also beneficial for model selection and tuning.

**5.Model Selection:** Choose an appropriate machine learning model for the task. Common models for demand prediction include:

* **Linear Regression**: Simple and interpretable but may not capture complex demand patterns.
* **Time Series Models**: Models like ARIMA or Exponential Smoothing are useful when there is a clear time-based trend.
* **Random Forest, Gradient Boosting, or XGBoost**: These ensemble methods are flexible and often perform well in demand prediction tasks.
* **Neural Networks**: Deep learning models, such as LSTM or GRU, can capture intricate patterns but require more data and computing resources.

**6.Model Training:** Train your chosen model on the training dataset. Adjust hyperparameters to improve model performance. You might also experiment with different feature sets and scaling techniques.

**7.Evaluation:** Assess the model's performance using appropriate evaluation metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE). Also, consider business-specific metrics like profit-based metrics.

**8.Model Tuning:** Fine-tune the model based on the evaluation results. This might involve adjusting hyperparameters, trying different algorithms, or exploring different feature engineering approaches.

**9.Deployment:** Once you have a well-performing model, deploy it in a production environment. This could involve integrating the model into your inventory management system or another relevant business process.

**10.Monitoring and Maintenance:** Continuously monitor the model's performance in the production environment. Over time, you may need to retrain the model with updated data to account for changing demand patterns.

**11.Feedback Loop:** Gather feedback from your model's predictions and real-world outcomes to further improve the model's accuracy and effectiveness.

Keep in mind that the quality and quantity of data you have, as well as the specific characteristics of your business, will influence the choice of model and the complexity of your solution. Additionally, data privacy and ethical considerations should be addressed when handling customer-related data for demand prediction.

Remember that developing an effective demand prediction model is an iterative process. It may require multiple iterations of data collection, model development, and evaluation to achieve optimal results.

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)

PROGRAM:

import pandas as pd

import numpy as np

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Create synthetic data for demonstration

data = pd.DataFrame({'price': np.random.rand(100) \* 100, # Random price between 0 and 100

'promotion': np.random.randint(0, 2, 100), # Random promotion (0 or 1)

'season': np.random.randint(1, 5, 100), # Random season (1 to 4)

'demand': np.random.randint(50, 200, 100) # Random demand between 50 and 200

})

# Define the features (X) and the target variable (y)

X = data[['price', 'promotion', 'season']]

y = data['demand']

# Split the data 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)

# Create and train a linear regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions on the test set

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

# Calculate the Mean Squared Error (MSE) to evaluate the model

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

print(f"Mean Squared Error: {mse}")

# Now, you can use the trained model to make predictions for new data

new\_data = pd.DataFrame({'price': [60.0], 'promotion': [1], 'season': [3]})

predicted\_demand = model.predict(new\_data)

print(f"Predicted Demand: {predicted\_demand[0]}")

OUTPUT:

Mean Squared Error:1537.14

Predicted Demand:135.58

CONCLUSION:

* In conclusion, demand prediction using machine learning and data analytics is a crucial tool for businesses and organizations in various industries. It enables them to make informed decisions about inventory management, production planning, pricing strategies, and resource allocation. By accurately forecasting demand, businesses can optimize their operations, reduce costs, and improve customer satisfaction.
* Demand prediction is a dynamic field that continues to evolve with the growth of big data, improved machine learning algorithms, and advances in computational power. Accurate predictions have the potential to drive cost savings, increase revenue, and enhance customer experiences, making it a valuable tool for modern businesses.