Product Demand Prediction with Machine learning

(Phase-4)

Feature Engineering:

1. Data Selection: Choose relevant features from your dataset. Consider factors like historical sales, product attributes, pricing, promotions, seasonality, and external variables like holidays or economic indicators.

2. Handling Missing Data: Address missing values in your dataset. You can either impute missing values or consider removing data points with significant missing information, depending on the dataset size and the nature of the data.

3. Categorical Encoding: Encode categorical variables into numerical format using techniques like one-hot encoding or label encoding.

4. Feature Scaling: Scale your numerical features, such as sales, price, and quantity, to ensure they have a similar range. Common methods include Min-Max scaling or Z-score standardization.

5. Feature Generation: Create new features if they might be relevant. For example, you could derive features like moving averages, seasonal indicators, or historical trends.
Feature Engineering:

6. Data Preprocessing: Start by cleaning the dataset. Handle missing values, outliers, and any inconsistencies in the data.

7. Feature Selection:Identify which features are relevant for demand prediction. You can use techniques like correlation analysis or feature importance from tree-based models to make informed choices.

8. Feature Creation: Create new features that might be useful for prediction. For example, you can derive features like seasonality, trends, or lag variables.

9. Normalization/Scaling:Scale numerical features, such as using Min-Max scaling or Standardization, to ensure they have similar ranges.

10. Categorical Encoding: Encode categorical variables using techniques like one-hot encoding or label encoding to make them suitable for modeling.

Model Training:

1. Split the Data: Divide your dataset into training, validation, and test sets to assess the model's performance properly. A common split is 70% training, 15% validation, and 15% test data.

2. Choose a Model: Select an appropriate model for your prediction task. Common choices include linear regression, decision trees, random forests, or more advanced models like XGBoost or neural networks.

3. Training the Model: Fit your chosen model to the training data. Ensure you tune hyperparameters for optimal performance.

4. Validation: Assess the model's performance using the validation dataset. Metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) are commonly used for regression tasks.

5. Hyperparameter Tuning: Fine-tune the model by adjusting hyperparameters based on validation results. Techniques like grid search or random search can help in finding the best parameters.

6. Select Algorithms:Choose the machine learning or deep learning algorithms you want to use for prediction. Common choices include linear regression, decision trees, random forests, gradient boosting, or neural networks.

7. Split Data:Divide the dataset into training, validation, and test sets. This helps in assessing the model's performance.

8. Train the Model: Use the training data to train your selected model. Fine-tune hyperparameters as needed.

9. Cross-Validation:Perform cross-validation to assess the model's robustness and avoid overfitting.

10. Regularization: Apply regularization techniques, such as L1 or L2 regularization, to prevent overfitting.

Evaluation:

1. Test Set Evaluation: Once you are satisfied with the model's performance on the validation data, evaluate it on the test dataset to get an unbiased estimate of its predictive power.

2. Performance Metrics: Calculate and interpret relevant performance metrics such as MAE, MSE, RMSE, or R-squared to understand how well the model predicts product demand.

3. Visualization: Visualize the model's predictions against the actual demand using graphs or plots to gain insights into where the model performs well or poorly.

4. Business Impact Assessment: Consider how the model's predictions can be utilized in your specific business context. Evaluate if it meets the goals and expectations of the project.

5. Iterate: If the model's performance is not satisfactory, consider revisiting feature engineering, model selection, or hyperparameter tuning and iterate through the process to improve the prediction accuracy.

6. Metrics:Select appropriate evaluation metrics for demand prediction. Common metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

7. Model Evaluation:Evaluate the model's performance using the chosen metrics on the validation set. This helps you understand how well your model is performing.

8. Fine-Tuning: If the model is underperforming, consider adjusting hyperparameters, changing algorithms, or improving feature engineering.

9. Testing:Once you're satisfied with the model's performance on the validation set, test it on the separate test set to assess its real-world predictive capability.

10. Interpretation: Interpret the model's results. Understand the significance of each feature in making predictions, which can provide valuable insights.

11. Deployment: If the model meets your requirements, deploy it for demand prediction in your real-world scenario.