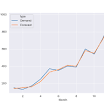
FEATURES ENGINEERING MODEL

Demand forecasting is a technique for the estimation of probable demand for a product or service in the future. Demand means outside requirements of a product or service. In general, forecasting means making an estimation in the present for a future occurring event.

Regression Modelling  
  
Regression models like Random Forest & XGBoost can also be used to forecast demand for the future. In our case, XGBoost has outperformed Random Forest. 24 data points are used as input and the 25th data points as prediction; the entire data set must be broken down into sets of 24+1

method is used for prodect demand

The five most popular demand forecasting methods are: trend projection, market research, sales force composite, Delphi method, and the econometric method.

Predicting product demand with machine learning typically involves several steps, including data preprocessing, feature engineering, and model selection. Here's a high-level overview of the process:

1. Data Collection:

Gather historical data related to the product, such as sales, pricing, promotional activities, and external factors (e.g., weather, holidays).

2. Data Preprocessing:

Clean and preprocess the data. This may involve handling missing values, dealing with outliers, and converting categorical data into a suitable format for machine learning models (e.g., one-hot encoding).

3. Feature Engineering:

Feature engineering is crucial for improving prediction accuracy. Some techniques include:

Lag Features\*\*: Create features representing past sales or demand patterns.

Seasonal Decomposition\*\*: Extract seasonality trends from the data.

-Text Mining\*\*: Analyze product descriptions or reviews for sentiment analysis.

External Data Integration\*\*: Incorporate external data like economic indicators or social media trends.

Time-based Features\*\*: Extract features such as day of the week, month, or year.

Aggregations\*\*: Compute statistics like mean, median, or standard deviation for specific time periods.

4. Data Splitting:

Split the data into training and testing sets to evaluate the model's performance. Time-based splitting is often used to mimic real-world scenarios.

5. Model Selection:

Choose an appropriate machine learning model for demand prediction. Common choices include:

Time Series Models\*\*: ARIMA, Exponential Smoothing, Prophet.

Regression Models\*\*: Linear Regression, Random Forest, XGBoost.

Neural Networks\*\*: LSTM or GRU networks for sequential data.

Hybrid Models\*\*: Combining multiple models for better accuracy.

6. Model Training:

Train the selected model on the training data.

7. Model Evaluation:

Evaluate the model's performance on the testing data using appropriate metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

8. Hyperparameter Tuning:

Optimize the model's hyperparameters to improve its performance.

9. Deployment:

Deploy the model in a production environment where it can make real-time or batch predictions.

10. Monitoring and Maintenance:

Continuously monitor the model's performance and retrain it periodically to adapt to changing demand patterns.

Remember that the success of your demand prediction model heavily depends on the quality of your data, the relevance of the features you engineer, and the choice of an appropriate model. Additionally, always keep an eye on new data and adapt your model accordingly to ensure it remains accurate over time.

Certainly! Let's delve into the specific steps involved in training and evaluating a machine learning model for product demand predictiotime.

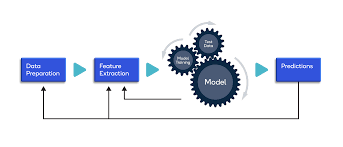
TRAINING AND EVALUATION:

1. Data Preprocessing:

Handle missing data: Impute missing values or remove incomplete data points.

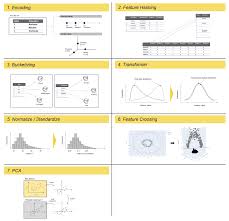
Data transformation: Normalize or scale numerical features if needed.

Encoding: Encode categorical variables using techniques like one-hot encoding. . Split the data into training and testing sets. For time series data, ensure chronological splitting.



2.Feature Engineering:

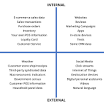
Create relevant features, as discussed in the previous response.

 Ensure that engineered features are consistent between the training and testing sets to avoid data leakage.

3. Model Selection:

Choose an appropriate machine learning algorithm based on the problem's nature (e.g., regression for continuous demand prediction).

For time series forecasting, consider models like ARIMA, Exponential Smoothing, or LSTM networks.

4. Model Training:

- Train the chosen model on the training dataset using the engineered features.

- Tune hyperparameters if necessary to optimize the model's performance.

- For time series models, fit the model to historical data, considering seasonality and trends.

5. Model Evaluation:

Evaluate the model's performance using relevant metrics, such as:

Mean Absolute Error (MAE)

Mean Squared Error (MSE)

Root Mean Squared Error (RMSE)

Mean Absolute Percentage Error (MAPE)

Visualize predictions against actual demand to identify discrepancies.

6. Cross-Validation (optional):

Implement k-fold cross-validation to assess model robustness and generalization to new data.

7. Hyperparameter Tuning (optional):

Use techniques like grid search or random search to fine-tune hyperparameters and optimize the model's performance.

8. Model Deployment (if applicable):

Deploy the trained model in a production environment, ensuring it can handle real-time or batch predictions as needed.

9. Monitoring and Maintenance:

Continuously monitor the model's performance in the production environment.

Re-train the model periodically with updated data to adapt to changing demand patterns.

10. Feedback Loop (optional):

Incorporate user feedback and model performance feedback into the model improvement process.

The key to a successful product demand prediction system is an iterative approach, where you continuously improve the model and adapt to changing conditions. Regularly retraining the model and monitoring its performance is essential to ensure it remains accurate over time.