PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

Problem Statement

The primary problem is to develop a machine learning model that can accurately predict product demand based on historical sales data and external factors. The goal is to assist businesses in optimizinginventory management and production planning to efficiently meet customer needs.

Design Thinking:

1. **Data Collection:** Gathering historical sales data and external factors that influence product demand, suchas marketing campaigns, holidays, economic indicators, etc.

2. **Data Preprocessing:** Cleaning and preprocessing the collected data, handling missing values, andconverting categorical features into numerical representations.

3. **Feature Engineering:** Creating additional features that capture seasonal patterns, trends, and externalinfluences on product demand.

4. **Model Selection:** Choosing suitable regression algorithms (e.g., Linear Regression, Random Forest,XGBoost) for demand forecasting.

5. **Model Training:** Training the selected model using the preprocessed data.

6. **Evaluation:** Evaluating the model's performance using appropriate regression metrics (e.g., MeanAbsolute Error, Root Mean Squared Error).

Dataset:

The dataset required for this project is available on Kaggle at the following link: https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning

✚ Data Collection :

Data collection is an essential part of the design thinking process. It allows designers to gain a deep understanding of the problem they   are trying to solve and the users they are designing for.
1. **Internal data:** This could include sales data, customer surveys, website analytics, and social media data.
2. **External data:** This could include market research reports, industry trends, and economic data.

Problem: A company that sells shoes is experiencing a decline in sales.

Data collection: The designers collect historical sales data and external factors that influence demand, such as marketing campaigns, holidays, and economic indicators.

Analysis: The designers analyze the data and identify the following trends:

- Sales have been declining steadily over the past year.
- Sales tend to increase during the summer months and decrease during the winter months.
- Sales are higher during the holiday season.
- Sales are lower during periods of high unemployment.

Insights: The designers conclude that the decline in sales is likely due to a combination of factors, including the seasonal nature of shoe sales, the recent economic downturn, and competition from other shoe retailers.

Design: Based on their insights, the designers develop new strategies to increase sales, such as offering seasonal discounts, launching new marketing campaigns during the holiday season, and targeting customers who are less likely to be affected by economic downturns.

Data Preprocessing :

Cleaning the data: This involves identifying and correcting errors in the data, such as typos, inconsistencies, and missing values.

Preprocessing the data: This involves transforming the data into a format that is compatible with your analysis tools. This may involve converting categorical features into numerical representations, scaling the data, or removing outliers.

Handling missing values: Missing values are a common problem in real-world datasets. There are a variety of ways to handle missing values, such as deleting them, imputing them with a default value, or using a machine learning algorithm to predict them.

Removing duplicate rows: If your dataset contains duplicate rows, you may want to remove them to reduce the size of the dataset and improve the accuracy of your analysis.

Converting categorical features into numerical representations: Categorical features are features that can take on a discrete set of values, such as gender, product category, or state of residence. In order to use categorical features in machine learning models, you need to convert them into numerical representations. This can be done using a variety of methods, such as one-hot encoding or label encoding.

Scaling the data: Scaling the data involves transforming the data so that all of the features have the same scale. This is important for some machine learning algorithms, such as linear regression.

Removing outliers: Outliers are data points that are significantly different from the rest of the data. They can be caused by errors in the data or by natural variation in the data. Outliers can skew the results of your analysis, so it is often important to remove them before you start your analysis.

Data preprocessing is an important step in the design thinking process. By cleaning and preprocessing your data, you can ensure that your data is accurate, complete, and in a format that is compatible with your analysis tools.

🪔 Feature Engineering:

Seasonal Features: Create binary or numerical indicators for seasons or months to account for regular cyclical patterns.

Trend Features: Calculate rolling averages or moving sums to identify upward or downward trends in demand.

Holiday Flags: Introduce binary flags for holidays to acknowledge their impact on demand fluctuations.

Marketing Features: Incorporate marketing campaign indicators or budgets to quantify their influence.

Economic Indicators: Include relevant economic data (e.g., GDP growth, inflation) as numerical features to consider macroeconomic effects.

Weather Data: If applicable, add weather-related features like temperature or precipitation that can affect demand in certain industries.

Lagged Features: Create lagged versions of demand data to capture temporal dependencies and autocorrelation.

Interaction Features: Multiply factors like marketing spend with product price to assess their combined influence.

Time Since Last Event: Calculate time elapsed since the last significant event to gauge lasting effects.

Custom Features: Design domain-specific features based on business insights or historical data patterns to enhance forecasting accuracy.

🔱 Model Selection:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression

# Load the dataset
data = pd.read_csv('demand_forecasting_data.csv')

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(data[['sales_history',
'external_factors']], data['demand'], test_size=0.25, random_state=42)

# Create the linear regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the mean absolute error (MAE)
MAE = np.mean(abs(y_pred - y_test))

# Print the MAE
print('MAE:', MAE)
```

🔱 Model Training:

Model Training for Product Demand Prediction

Once you have preprocessed your data, you can train your machine learning model. This involves feeding the model the preprocessed data and allowing it to learn the relationships between the input features and the target variable (product demand).

There are a variety of different machine learning algorithms that can be used for product demand prediction. Some popular choices include linear regression, decision trees, and random forests.

The specific machine learning algorithm that you choose will depend on a number of factors, such as the size and complexity of your dataset, the desired accuracy of your model, and the computational resources available to you.

Once you have selected a machine learning algorithm, you need to train the model on your training data. This involves feeding the model the training data and allowing it to learn the relationships between the input features and the target variable.

The model training process can take some time, depending on the size and complexity of your dataset and the machine learning algorithm that you are using.

Once the model is trained, you can evaluate its performance on a holdout dataset that was not used to train the model. This will help you to assess the accuracy and reliability of the model.

If the model's performance is satisfactory, you can deploy the model to production. This involves making the model available to businesses so that they can use it to predict product demand.

Here is a summary of the model training process in fewer words:

1. Select a machine learning algorithm.
2. Train the model on your training data.
3. Evaluate the model's performance on a holdout dataset.
4. Deploy the model to production.

Evaluation:
Evaluation of a Machine Learning Model for Product Demand Prediction

Once you have trained a machine learning model for product demand prediction, you need to evaluate its performance to assess its accuracy and reliability. This can be done by using a holdout dataset that was not used to train the model.

Here are some appropriate regression metrics that you can use to evaluate the model's performance:

Mean Absolute Error (MAE): The MAE is the average of the absolute differences between the predicted and actual values.
Root Mean Squared Error (RMSE): The RMSE is the square root of the average of the squared differences between the predicted and actual values.
Mean Squared Error (MSE): The MSE is the average of the squared differences between the predicted and actual values.

These metrics can be used to compare the performance of different models and to identify the model that is most accurate for your dataset.

The interpretability of the model: Some machine learning models are more interpretable than others. This means that it is easier to understand how the model makes its predictions. Interpretability is important if you want to understand the factors that are driving demand for your product.

The computational cost of the model: Some machine learning models are more computationally expensive than others. This means that they require more computing power to train and deploy. It is important to choose a model that is computationally feasible for your business needs.

Once you have evaluated the model's performance, you can use the model to predict product demand and make informed decisions about inventory management and production planning.

Here is an example of how you could evaluate a machine learning model for product demand prediction using the MAE metric:

1. Split your dataset into a training set and a holdout set.
2. Train the model on the training set.
3. Use the model to predict product demand for the holdout set.
4. Calculate the MAE between the predicted and actual values for the holdout set.

A lower MAE indicates a more accurate model.

You can repeat this process for different machine learning models to compare their performance and identify the model that is most accurate for your dataset.