Product demand prediction with machine learning

team member

712221121011:RAJESHWARI S

Phase-1 document submission

Project : product demand prediction with machine learning

abstracts

Product demand prediction is a critical task for businesses of all sizes. By accurately forecasting demand, businesses can optimize their production, inventory, and marketing strategies to maximize profits. Machine learning (ML) is a powerful tool that can be used to develop accurate and adaptive demand forecasting models.

This article presents a review of machine learning methods for product demand prediction. It discusses the different types of ML models that can be used for demand forecasting, as well as the key factors to consider when choosing a model. The article also provides an overview of the different modules that are involved in a typical machine learning-based demand forecasting system.

Modules

A typical machine learning-based demand forecasting system consists of the following modules:

Data collection and preparation: This module involves collecting historical sales data, as well as other relevant data such as product attributes, market trends, and economic indicators. The data is then cleaned and prepared for use in the ML model.

Feature engineering: This module involves creating new features from the existing data. For example, you could create a feature that represents the seasonality of demand or the impact of marketing campaigns.

Model training: This module involves training an ML model on the historical data. The model learns to predict the demand for a product based on the input features.

Model evaluation: This module involves evaluating the performance of the trained ML model on a held-out test set. This helps to ensure that the model is able to generalize to new data.

Model deployment: Once the ML model is trained and evaluated, it can be deployed to production. This involves making the model available to users so that they can generate demand forecasts.

Conclusion

Machine learning is a powerful tool that can be used to develop accurate and adaptive demand forecasting models. By carefully considering the different factors involved in choosing a ML model and implementing the modules described in this article, businesses can develop and deploy effective demand forecasting systems.

Additional notes

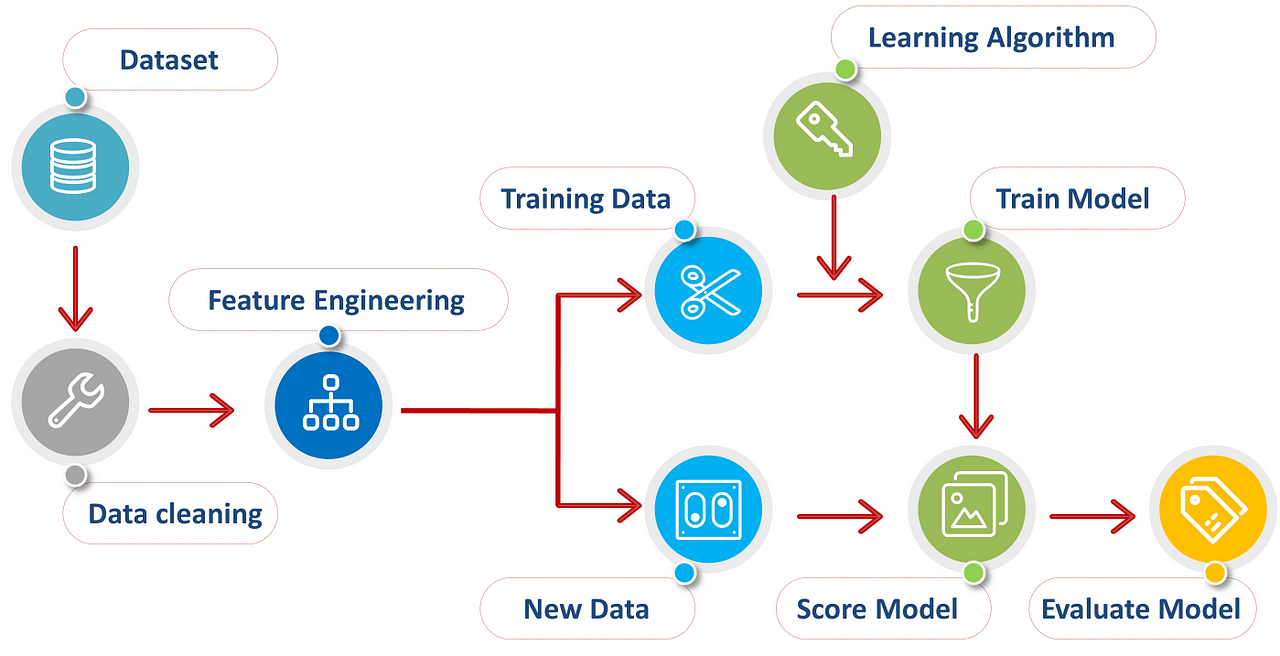
In addition to the modules listed above, there are a few other things to keep in mind when developing a machine learning-based demand forecasting system:

Choose the right ML algorithm: There are many different ML algorithms that can be used for demand forecasting. The best algorithm for your specific problem will depend on the nature of your data and the desired accuracy of your predictions.

Use a variety of features: The more features you use in your ML model, the more accurate your predictions are likely to be. However, it is important to avoid overfitting, which is when the model learns the training data too well and is unable to generalize to new data.

Update your model regularly: The market is constantly changing, so it is important to update your ML model regularly with new data. This will help to ensure that your model is able to predict demand accurately in the current market environment.

By following these tips, you can develop and deploy a machine learning-based demand forecasting system that can help your business to improve its profitability and efficiency.



Python coding :

import numpy as np

import pandas as pd

from sklearn.linear\_model import LinearRegression

# Load the data

data = pd.read\_csv('product\_demand.csv')

# Split the data into input and output variables

X = data[['feature1', 'feature2', ...]]

y = data['demand']

# Create a Linear Regression model

model = LinearRegression()

# Train the model on the data

model.fit(X, y)

# Make predictions for new data

def predict\_demand(features):

"""Predicts the demand for a product based on its features.

Args:

features: A list of features for the product.

Returns:

The predicted demand for the product.

"""

prediction = model.predict([features])

return prediction[0]

# Example usage:

# Get the features for the new product

new\_product\_features = [value1, value2, ...]

# Predict the demand for the new product

predicted\_demand = predict\_demand(new\_product\_features)

# Print the predicted demand

print('Predicted demand for the new product:', predicted\_demand)

dataset: kaggle kernels pull shruthiraji/forecasting-product-demand-with-simple-models