PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

CONTENTS

PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

The demand for a product change as the price of the product changes. To give a real-life example, if a product is not needed, demand decreases when price increases, and demand increases when price decreases. If you want to know how to use machine learning to predict product demand, this article is for you. This article describes the task of predicting product demand with machine learning using Python

PRODUCT DEMAND PREDICTION CASE STUDY

- A product company plans to offer discounts on its product during the upcoming holiday season. The company wants to find the price at which its product can be a better deal compared to its competitors. For this task, the company provided a dataset of past changes in sales based on price changes. You need to train a model that can predict the demand for the product in the market with different price segments.
- The **dataset** that we have for this task contains data about:the product id
- ;store id;
- total price at which product was sold;
- base price at which product was sold;
- Units sold (quantity demanded

PRODUCT DEMAND PREDICTION USING PYTHON

The necessary Python libraries and the dataset we need for the task of product demand prediction:

```
import pandas as pd
import numpy as np
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
Data.head()
```

PRODUCT DEMAND PREDICTION MODEL

- Training a machine learning model to predict the demand for the product at different prices. I will choose the Total Price and the Base Price column as the features to train the model, and the Units Sold column as labels for the model:
- x = data[["Total Price", "Base Price"]]
- y = data["Units Sold"]
- the data into training and test sets and use the decision tree regression algorithm to train our model:
- xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
- from sklearn.tree import DecisionTreeRegressormodel = DecisionTreeRegressor()
- model.fit(xtrain, ytrain)
- input the features (Total Price, Base Price) into the model and predict how much quantity can be demanded based on those values:
- 1#features = [["Total Price", "Base Price"]]
- 2features = np.array([[133.00, 140.00]])
- 3model.predict(features)
- array([27.])

BUILD A PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

- Creating a product demand prediction model with machine learning involves several steps:
- **Data Collection**: Gather historical data on product sales, including variables like time, price, promotions, and external factors (e.g., holidays, economic indicators).
- **Data Preprocessing**: Clean the data by handling missing values, outliers, and encoding categorical variables. Perform feature engineering to create relevant features.
- **Data Splitting**: Divide the data into training and testing sets. You may also consider using techniques like time series cross-validation if your data is time-dependent.
- **Model Selection**: Choose an appropriate machine learning model for regression or time series forecasting. Common choices include Linear Regression, Decision Trees, Random Forests, XGBoost, ARIMA, or LSTM for time series data

SUMMARY

This is how we can use Python to train a machine learning model for the task of predicting product demand. Price is one of the most important factors influencing product demand. If there is no need, few people will buy the product even if the price goes up. I hope you enjoyed this article on predicting product demand with machine learning using Python.