PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

Team Member

712221121004: BHARANI.R

Phase 5 Submission Document

TITLE: Product Demand Prediction With Machine Learning

Phase 5: Development part 2

TOPIC : Document the product demand prediction project

INTRODUCTION:

Product demand prediction is a crucial task for businesses to optimize their supply chain management and maximize profits. Machine learning is a powerful tool that can help businesses predict demand for their products with high accuracy. 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.

Machine learning:

It is a subsection of artificial intelligence that analyzes data to learn from it and make predictions or decisions . Machine learning algorithms can be used to understand and improve from experience automatically. That makes them very effective in demand forecasting, which can be used to predict future demand trends based on past data.

Demand Forecasting:

Demand forecasting predicts how much of a particular product or service customers will demand over a specific period. Demand forecasting is an important part of business supply planning. It helps businesses make informed decisions about production, inventory levels, marketing strategies, financial planning, and other aspects that depend on customer demand

CLASSIFICATIONS

- Very short-term: typically from seconds or minutes to several hours.
- Short-Term: typically from hours to weeks.
- Medium-Term: typically from a week to a year.
- Long-Term: typically more than a year.

Applications of machine learning in demand forecasting:

Some real-world applications of machine learning in demand forecasting include:

- ❖ Predicting consumer demand for new products
- ❖ Forecasting sales for retail businesses
- ❖ Forecasting energy demand
- ❖ Predicting demand for transportation services

Benefits :

There are many benefits to using machine learning for demand forecasting, including the following:

- Speed processing of data – Machine learning collects data automatically. Hence you will get all data very quickly.
- Increased accuracy – Machine learning can accurately predict demand patterns based on past data, which can help businesses more accurately forecast future needs and better allocate resources.
- Reduced costs – Machine learning can help reduce the number of wasted resources due to over or under-stocking, which leads to significant savings for businesses.

- Improved decision-making – By providing accurate forecasts, machine learning can help businesses make better inventory levels, pricing, production planning, and more decisions. That helps companies stay competitive and improve their bottom line.

Best Machine learning software for Demand forecasting

- Minitab
- Statgraphics Centurion
- Google Cloud Bigquery
- Prime AI
- Valohai
- Intersect Labs
- Neuton AutoML
- TADA
- SAS Customer Intelligence 360
- OpenText Magellan

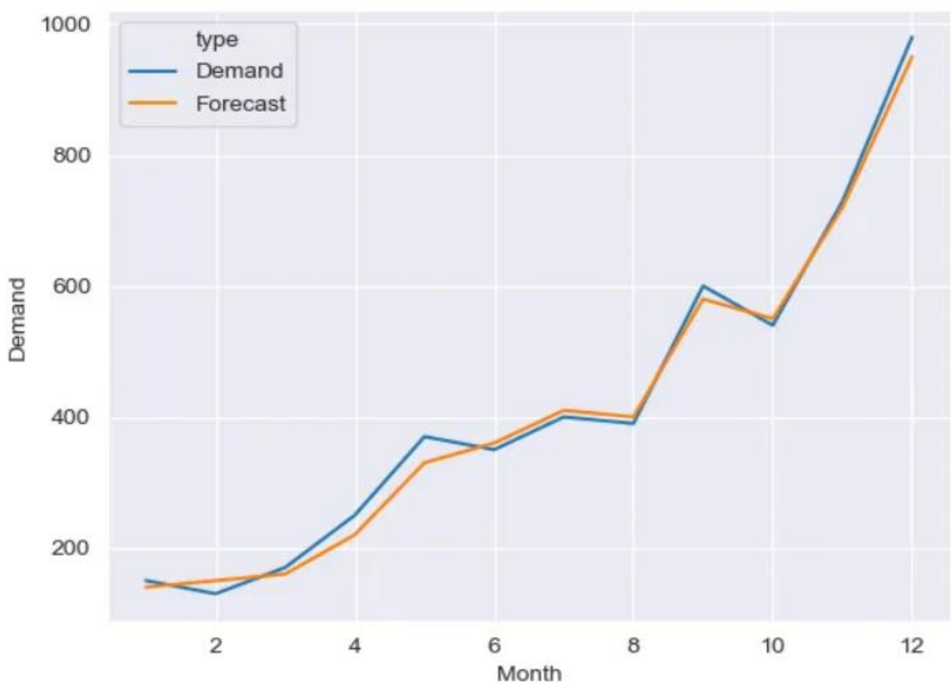
Design thinking process

Design thinking is a problem-solving approach that involves empathizing with the user, defining the problem, ideating solutions, prototyping, and testing. It is a human-centered approach that aims to create innovative solutions that meet the needs of the user.Demand forecasting is the process of estimating future demand for a product or service.

It is an essential component of supply chain management and helps businesses make informed decisions about production, inventory, and pricing. Machine learning is a subset of artificial intelligence that involves training algorithms to learn from data and make predictions.In demand forecasting, machine learning can be used to analyze historical sales data and identify patterns and trends. This information can then be used to make accurate predictions about future demand .

Design thinking can be used to identify user needs and preferences and develop solutions that meet those needs. By combining machine learning with design thinking, businesses can create demand forecasting models that are more accurate and effective in meeting user needs. However, it’s important to note that demand forecasting is not an exact science and there are many factors that can influence demand such as weather, seasonality, economic conditions, etc.

DATASET:



DATA PREPERATION

```
import pandas as pd

import numpy as np

df = pd.read_csv('train.csv')

df = df[df['item'] == 1]

df['date'] = pd.to_datetime(df['date']) - pd.to_timedelta(7, unit='d'

df = df.filter(['date', 'sales']).groupby([pd.Grouper(key='date', freq='W-
MON')]).sum().  reset_index()
```

First, we prepare our data, after importing our needed modules we load the data into a pandas dataframe. In the supplied train.csv there are 50 items in this example we'll do predictions of sales for item 1 on a weekly basis. The data is separated by days, and stores that sold that particular item in no particular order.

```
print(df.head())

0   2012-12-31    894

1   2013-01-07    863

2   2013-01-14    867

3   2013-01-21    816

4   2013-01-28    969
```

let's make another column in our dataframe with last week's sales.

```
df['shift_sale'] = df['sales'].shift(1)

df = df.iloc[1:]
```

Because we don't have the week before our first entry we just drop that row. Another column we want to make is a four-week average. For that, we'll write a quick method.

```
def four_week_avg(sales):
 sum = 0
 week_avg = []
for i in range(3, -1, -1):
for j in range(i):
sum += sales[j]
if(i!=0):
week_avg.append(sum/i)
sum = 0
week_avg.append(sales[0])
week_avg.reverse()
for row in range(len(sales) - 4):
for row in range(row, row + 4):
sum += sales[row]
week_avg.append(sum / 4)
sum = 0
return week_avg df['week_avg'] = four_week_avg(df['sales'].tolist())
```

```
print(df.head())

sales  shift_sale    week_avg

0       863       894.0  863.000000

1       867       863.0  863.000000

2       816       867.0  865.000000

3       969       816.0  848.666667

4       920       969.0  878.750000
```

Model and Evaluation

For our metrics and evaluation, we first need to import some modules.

```
from matplotlib import pyplot as plt

from sklearn import svm

from sklearn.model_selection import train_test_split

from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error,
median_absolute_error, explained_variance_score, max_error
```

Then we split our data into a training set and a test set for evaluation later. We'll predict sales for one year.

Then we split our data into a training set and a test set for evaluation later. predict sales for one year.

```
test = df.iloc[-52:]

df = df.iloc[:-52]
```

Using scikits train_test_split we are going to split the data for training and validation.

```
X = df.drop('sales', axis=1)

y = df['sales']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

clf = svm.SVR(C=1, kernel='linear', degree=8, gamma='scale', coef0=10)

clf.fit(X_train, y_train)

predictions = clf.predict(X_test)

print(f'Model fit results:\n'

f'r2_score {r2_score(y_test, predictions)} \t MSE {mean_squared_error(y_test, predictions)}'

f'\tEVS {explained_variance_score(y_test, predictions)} \n MAE {mean_absolute_error(y_test,
```

predictions)}'

f'\tMAD {median_absolute_error(y_test, predictions)}\t ME {max_error(y_test, predictions)}')

###OUTPUT###

Model fit results:

r2_score 0.9071953443448584 MSE 6553.674543344077

EVS 0.9175800366290838 MAE 58.80295451823111

MAD 37.648574124556035 ME 304.51308147895793

Now let’s test it with the data we dropped out at the beginning, which would show how our model would perform for one year.

predictions = clf.predict(test.drop('week_sale', axis=1))

print(f'Model test results:\n'

f'r2_score {r2_score(test["week_sale"], predictions)} \t MSE {mean_squared_error(test["week_sale"], predictions)}'

f'\tEVS {explained_variance_score(test["week_sale"], predictions)} \n MAE {mean_absolute_error(test["week_sale"], predictions)}'

f'\tMAD {median_absolute_error(test["week_sale"], predictions)}\t ME {max_error(test["week_sale"], predictions)}')

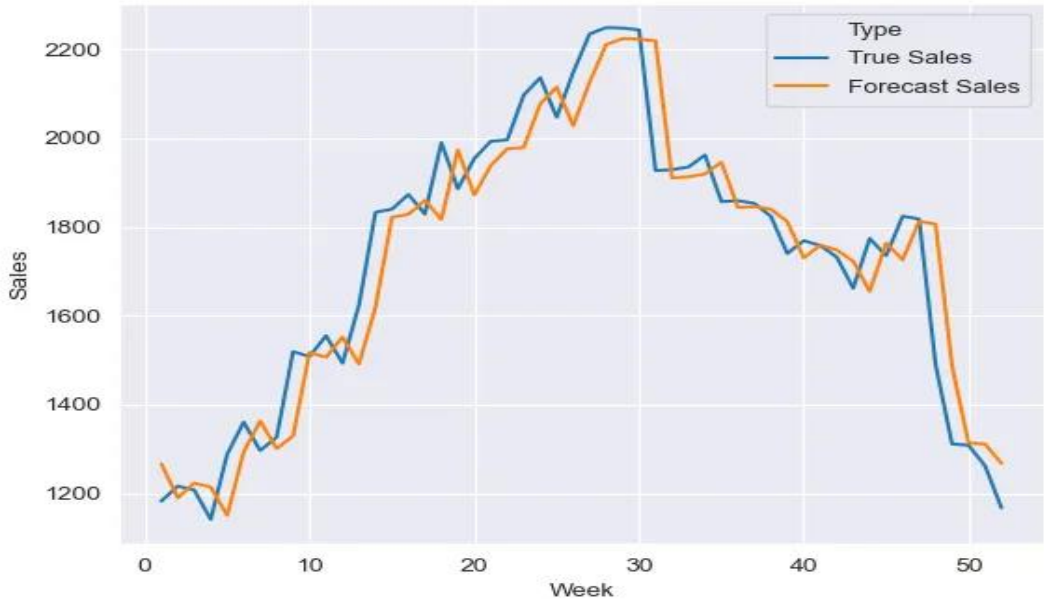
###OUTPUT###

Model test results:

r2_score 0.8996996866756606 MSE 10089.10921850138

EVS 0.900221498465065 MAE 73.23050276764262

MAD 56.762729750946164 ME 316.3516922406136



__DATA PREPROCESSING:__

Demand forecasting is the practice of optimizing business operations and resources through predicting future demand. The process uses data analysis to predict the demand for your product or service in the coming weeks, months or even years. It doesn't rely on guesswork; it's about using historical sales data, market trends and other factors to make educated estimates.

Steps in Data Preprocessing

There are six steps of data preprocessing in machine learning

Step 1: Import the Libraries

The foremost step of data preprocessing in machine learning includes importing some libraries. A library is basically a set of functions that can be called and used in the algorithm. There are many libraries available in different programming languages.

Step 2: Import the Loaded Data

The next important step is to load the data which has to be used in the machine learning algorithm. This is the most important machine learning preprocessing step. Collected data is to be imported for further assessment.Once the data is loaded, checking for noisy or missing content is important.

Step 3: Check for Missing Values

Assess the loaded data and check for missing values. If missing values have been found, there are particularly two ways to resolve this issue: Either Remove the entire row that contains a missing value. However, removing the entire row can generate a possibility of losing some important data. This approach is useful if the dataset is very largeOr Estimate the value by taking the mean, median or mode.

Step 4: Arrange the Data

Machine learning modules cannot understand non-numeric data. It is important to arrange the data in a numerical form in order to prevent any problems at later stages. Converting all text values into numerical form is the solution to this problem. You can use the LabelEncoder() function to do this.

Step 5: Do Scaling

Scaling is a technique that can convert data values into shorter ranges. Rescaling and Standardization can be used for scaling the data.

Step 6: Distribute Data into Training, Evaluation and Validation Sets

The final step is to distribute data in three different sets, namely

- Training
- Validation
- Evaluation

Demand forecasting is the process of predicting the quantity of goods and services that will be demanded by consumers at a future point in time 1. There are several methods to forecast demand, including the Delphi technique, conjoint analysis, intent survey, trend projection method, and econometric forecasting Here are some of the most popular and crucial methods in demand forecasting:

Delphi technique: This method involves collecting opinions from experts in the field and using them to make predictions .

Conjoint analysis: This method involves analyzing customer preferences and behavior to predict future demand .

Intent survey: This method involves asking customers about their future purchase intentions.

Trend projection method: This method involves analyzing historical data to identify trends and using them to predict future demand.

Econometric forecasting: This method involves using statistical models to analyze the relationship between demand and other factors such as price, income, and advertising.

CONCLUSION:

We went through how with little data preparation and some knowledge of machine learning we can make a forecast for sales of a product for an entire year. This model is far from perfect and we can see a specific time delay where the model catches up. With some tinkering with the parameters and better data preparation, the results can get better. I have left the technical parts out as this is more of a walkthrough on how to use SVR. If you are interested in the details I have put some links in the post where you can read into it more.