

DOMAIN : APPLIED DATA SCIENCE

PROJECT : PRODUCT DEMAND PREDICTION WITH
MACHINE LEARNINGS

**Demand Forecasting to predict
future demand of Products**



TEAM MEMBERS:

- 1.ASHIKA.A (821021104011)
- 2.BANUPRIYA.S (821021104014)
- 3.JAYAPRIYA.J (821021104023)
- 4.PRIYA NANDHINI.L(821021104036)
- 5.VARSHA.N (821021104053)

PHASE 3 : DEVELOPMENT PART 1

BEGIN BUILDING THE PRODUCT DEMAND PREDICTION MODEL BY LOADING AND PREPROCESSING THE DATASET.

TABLE OF CONTENTS:

- INTRODUCTION
- DATASET LOADING
 1. IMPORTING THE REQUIRED LIBRARIES
 2. LOADING THE DATASET
- DATA PREPROCESSING
 1. HANDLING THE MISSING VALUES
 2. REMOVING MISSING VALUES
 3. TRAIN, TEST AND SPLIT
 4. FEATURE SCALING
- CONCLUSION

INTRODUCTION:

This project involves 6 steps. They are data collection, data preprocessing, feature engineering, model selection, training, and evaluation.

Here , we are going to work with dataset loading and data preprocessing.

DATASET LOADING:

In dataset loading, here is the dataset we are going to use for product demand that is already provided from Kaggle website.

On before loading the datasets, we have to import the necessary libraries and packages.

<https://www.kaggle.com/datasets/chakradharmattapalli/productdemandprediction-with-machine-learning>

1.Importing the required libraries :

In first, we have to import the necessary libraries and packages. Here , we imported the NumPy and pandas.

NUMPY- It is the fundamental package for **scientific computing** with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

PANDAS - Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series

```
# importing required libraries
import pandas as pd
import numpy as np
```

2.Loading the Dataset:

For loading the dataset,

```
# importing required libraries
import pandas as pd
import numpy as np
```

```
#loading the dataset
data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")
print(data)
```

The output of the above code for loading the dataset will be,

```
C:\Users\CT\PycharmProjects\pythonProject5\venv\Scripts\python.exe C:\Users\CT\PycharmProjects\pythonProject5\phase3.py
```

| ID | Stone ID | Total Price | Base Price | Units Sold |
|--------|----------|-------------|------------|------------|
| 0 | 1 | 8091 | 99.0375 | 111.8625 |
| 1 | 2 | 8091 | 99.0375 | 99.0375 |
| 2 | 3 | 8091 | 133.9500 | 133.9500 |
| 3 | 4 | 8091 | 133.9500 | 133.9500 |
| 4 | 5 | 8091 | 141.0750 | 141.0750 |
| ... | ... | ... | ... | ... |
| 150145 | 212638 | 9984 | 235.8375 | 235.8375 |
| 150146 | 212639 | 9984 | 235.8375 | 235.8375 |
| 150147 | 212642 | 9984 | 357.6750 | 483.7875 |
| 150148 | 212643 | 9984 | 141.7875 | 191.6625 |
| 150149 | 212644 | 9984 | 234.4125 | 234.4125 |

```
[150150 rows x 5 columns]
Process finished with exit code 0
```

DATA PREPROCESSING :

Data Cleaning uses methods to handle **incorrect, incomplete, inconsistent, or missing values**.

STEPS INVOLVED IN DATA PREPROCESSING:

1.Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

(a). Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways.

1. Ignore the tuples:

This approach is suitable only when the dataset we have is quite large

and multiple values are missing within a tuple.

2. Fill the Missing values:

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

(b). Noisy Data:

Noisy data is a meaningless data that can't be interpreted by machines. It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

1. Binning Method:

This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

2. Regression:

Here data can be made smooth by fitting it to a regression function. The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

3. Clustering:

This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. Normalization:

It is done in order to scale the data values in a specified range (-1.0 to

1.0 or 0.0 to 1.0)

2. Attribute Selection:

In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

3. Discretization:

This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

4. Concept Hierarchy Generation:

Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

3. Data Reduction:

Data reduction is a crucial step in the data mining process that involves reducing the size of the dataset while preserving the important information. This is done to improve the efficiency of data analysis and to avoid overfitting of the model. Some common steps involved in data reduction are:

Feature Selection:

This involves selecting a subset of relevant features from the dataset. Feature selection is often performed to remove irrelevant or redundant features from the dataset. It can be done using various techniques such as correlation analysis, mutual information, and principal component analysis (PCA).

Feature Extraction:

This involves transforming the data into a lower-dimensional space while preserving the important information. Feature extraction is often used when the original features are high-dimensional and complex. It can be done using techniques such as PCA, linear discriminant analysis (LDA), and non-negative matrix factorization (NMF).

Sampling:

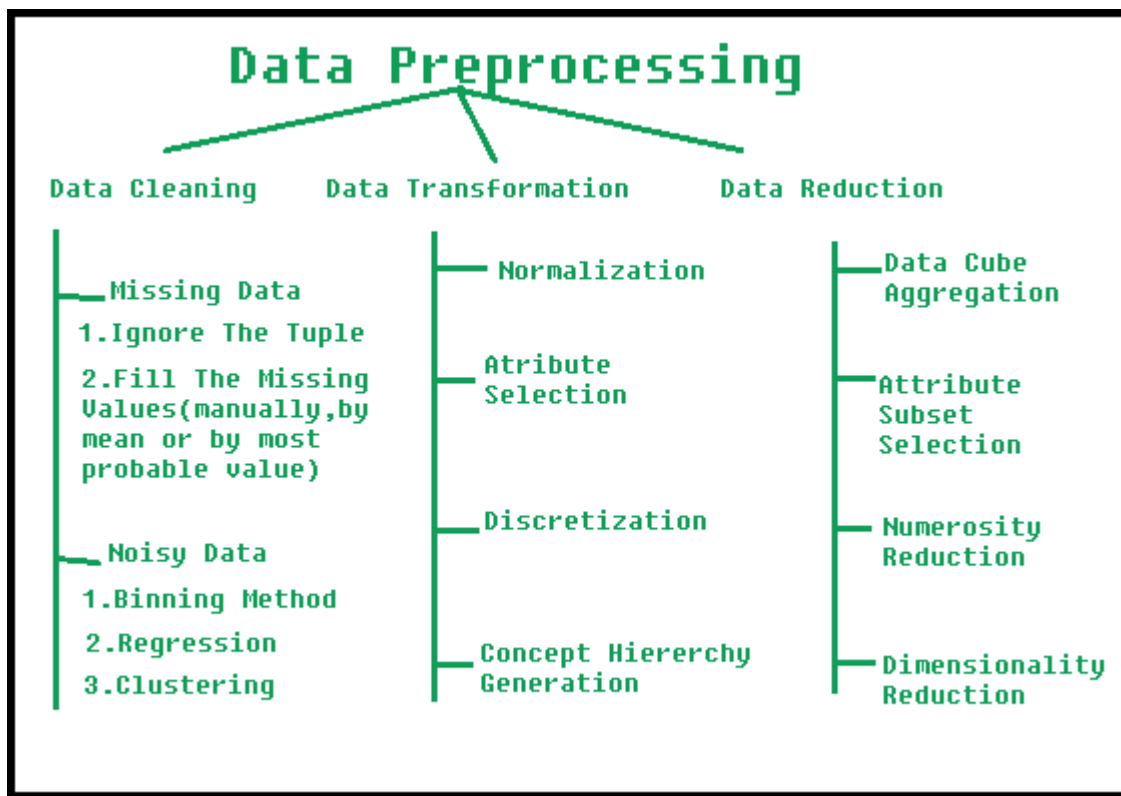
This involves selecting a subset of data points from the dataset. Sampling is often used to reduce the size of the dataset while preserving the important information. It can be done using techniques such as random sampling, stratified sampling, and systematic sampling.

Clustering:

This involves grouping similar data points together into clusters. Clustering is often used to reduce the size of the dataset by replacing similar data points with a representative centroid. It can be done using techniques such as k-means, hierarchical clustering, and density-based clustering.

Compression:

This involves compressing the dataset while preserving the important information. Compression is often used to reduce the size of the dataset for storage and transmission purposes. It can be done using techniques such as wavelet compression, JPEG compression, and gzip compression.



1.HANDLING THE MISSING DATA:

Count NaN values using isnull():

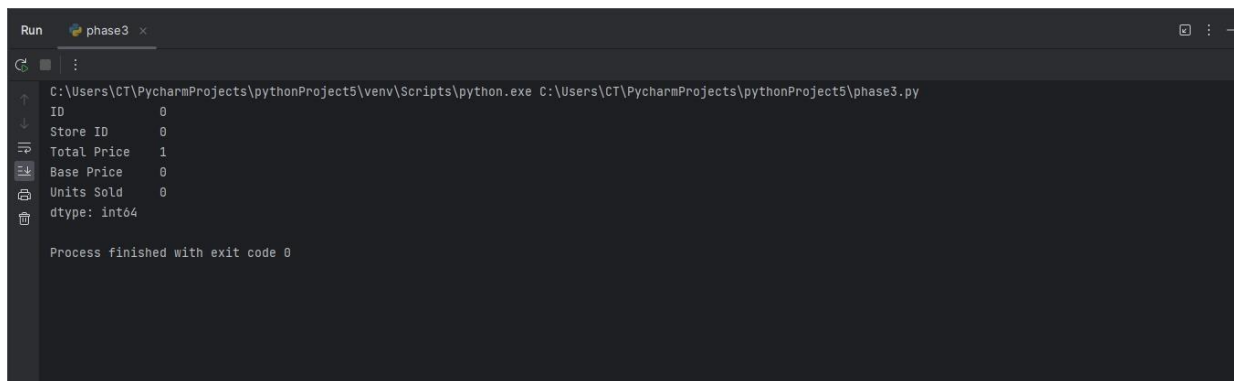
It returns a boolean same-sized object indicating if the values are NA. Missing values get mapped to True and non-missing value gets mapped to False. Calling the sum() method on the isnull() series returns the count of True values which actually corresponds to the number of NaN values.

```
# importing required libraries
import pandas as pd
import numpy as np

#loading the dataset
data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")
print(data)
#data preprocessing

#handling missing values
print(data.isnull().sum())
```

The output of the above code will be,



```
Run phase3 x
C:\Users\CT\PycharmProjects\pythonProject5\venv\Scripts\python.exe C:\Users\CT\PycharmProjects\pythonProject5\phase3.py
ID      0
Store ID 0
Total Price 1
Base Price 0
Units Sold 0
dtype: int64

Process finished with exit code 0
```

2.REMOVING MISSING VALUES:

The first method is to remove all rows that contain missing values or, in extreme cases, entire columns that contain missing values. This can be performed by using df. dropna() function. axis=0 or axis=1 is used to delete rows/columns with NaN values.

```
# importing required libraries
import pandas as pd
```



```

import numpy as np

#loading the dataset
data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")
print(data)

#data preprocessing

#handling missing values
print(data.isnull().sum())

#removing missing values

data.dropna(inplace=True)
print(data.isnull().sum())
print(data)

```

The output of the above code will be ,

The screenshot shows the output of the code in a Jupyter Notebook. It displays the first few rows of the dataset and the total number of rows and columns.

| | ID | Store ID | Total Price | Base Price | Units Sold |
|--------|--------|----------|-------------|------------|------------|
| 0 | 1 | 8091 | 99.0375 | 111.8625 | 20 |
| 1 | 2 | 8091 | 99.0375 | 99.0375 | 28 |
| 2 | 3 | 8091 | 133.9500 | 133.9500 | 19 |
| 3 | 4 | 8091 | 133.9500 | 133.9500 | 44 |
| 4 | 5 | 8091 | 141.0750 | 141.0750 | 52 |
| ... | ... | ... | ... | ... | ... |
| 150145 | 212638 | 9984 | 235.8375 | 235.8375 | 38 |
| 150146 | 212639 | 9984 | 235.8375 | 235.8375 | 30 |
| 150147 | 212642 | 9984 | 357.6750 | 483.7875 | 31 |
| 150148 | 212643 | 9984 | 141.7875 | 191.6625 | 12 |
| 150149 | 212644 | 9984 | 234.4125 | 234.4125 | 15 |

[150149 rows x 5 columns]

Process finished with exit code 0

3.TRAIN ,TEST AND SPLIT:

- Train/Test is a method to measure the accuracy of your model.
- It is called Train/Test because you split the data set into two sets: a training set and a testing set.
- 80% for training, and 20% for testing.
- You train the model using the training set.

- You test the model using the testing set.
- Train the model means create the model.
- Test the model means test the accuracy of the model.

```
# importing required libraries
import pandas as pd
import numpy as np

#loading the dataset
data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")
print(data)
#data preprocessing

#handling missing values
print(data.isnull().sum())

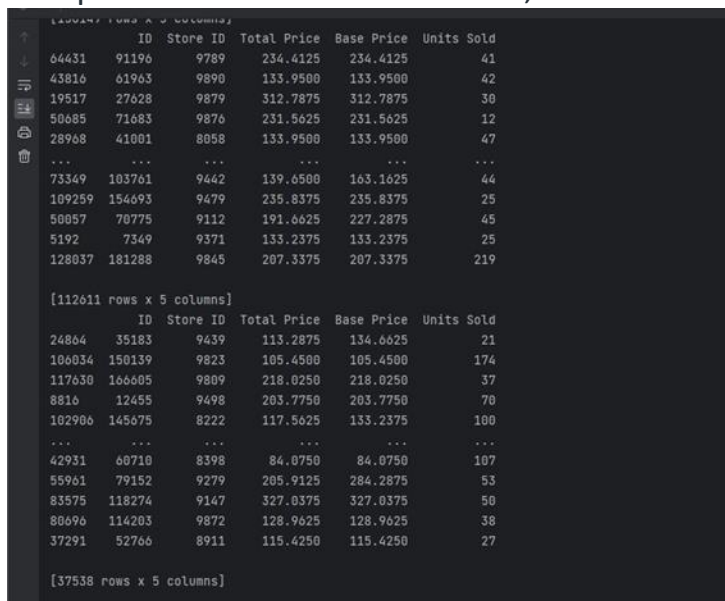
#removing missing values

data.dropna(inplace=True)
print(data.isnull().sum())
print(data)

#train test

from sklearn.model_selection import train_test_split
x_train,x_test=train_test_split(data,test_size=0.25,random_state=1)
print(x_train)
print(x_test)
```

The output of the above code will be,



| ID | Store ID | Total Price | Base Price | Units Sold |
|--------|----------|-------------|------------|------------|
| 644631 | 91196 | 9789 | 234.4125 | 41 |
| 43816 | 61963 | 9890 | 133.9500 | 42 |
| 19517 | 27628 | 9879 | 312.7875 | 30 |
| 50485 | 71683 | 9876 | 231.5625 | 12 |
| 28968 | 41001 | 8058 | 133.9500 | 47 |
| ... | ... | ... | ... | ... |
| 73349 | 103761 | 9442 | 139.0500 | 44 |
| 109259 | 154693 | 9479 | 235.8375 | 25 |
| 50057 | 70775 | 9112 | 191.6625 | 45 |
| 5192 | 7349 | 9371 | 133.2375 | 25 |
| 128037 | 181288 | 9845 | 207.3375 | 219 |

| ID | Store ID | Total Price | Base Price | Units Sold |
|--------|----------|-------------|------------|------------|
| 24804 | 35183 | 9439 | 113.2875 | 21 |
| 104034 | 150139 | 9823 | 105.4500 | 174 |
| 117630 | 166605 | 9809 | 218.0250 | 37 |
| 8816 | 12455 | 9498 | 203.7750 | 70 |
| 102906 | 145675 | 8222 | 117.5625 | 100 |
| ... | ... | ... | ... | ... |
| 42931 | 60710 | 8398 | 84.0750 | 107 |
| 55961 | 79152 | 9279 | 205.9125 | 53 |
| 83575 | 118274 | 9147 | 327.0375 | 50 |
| 80496 | 114203 | 9872 | 128.9625 | 38 |
| 37291 | 52766 | 8911 | 115.4250 | 27 |

4.FEATURE SCALING:

- Feature scaling is the process of normalizing the range of features in a dataset.
- Real-world datasets often contain features that are varying in degrees of magnitude, range, and units.
- Therefore, in order for machine learning models to interpret these features on the same scale, we need to perform feature scaling.

```
# importing required libraries
import pandas as pd
import numpy as np

#loading the dataset
data=pd.read_csv(r"C:\Users\Administrator\PoductDemand.csv")
print(data)
#data preprocessing

#handling missing values
print(data.isnull().sum())

#removing missing values

data.dropna(inplace=True)
print(data.isnull().sum())
print(data)

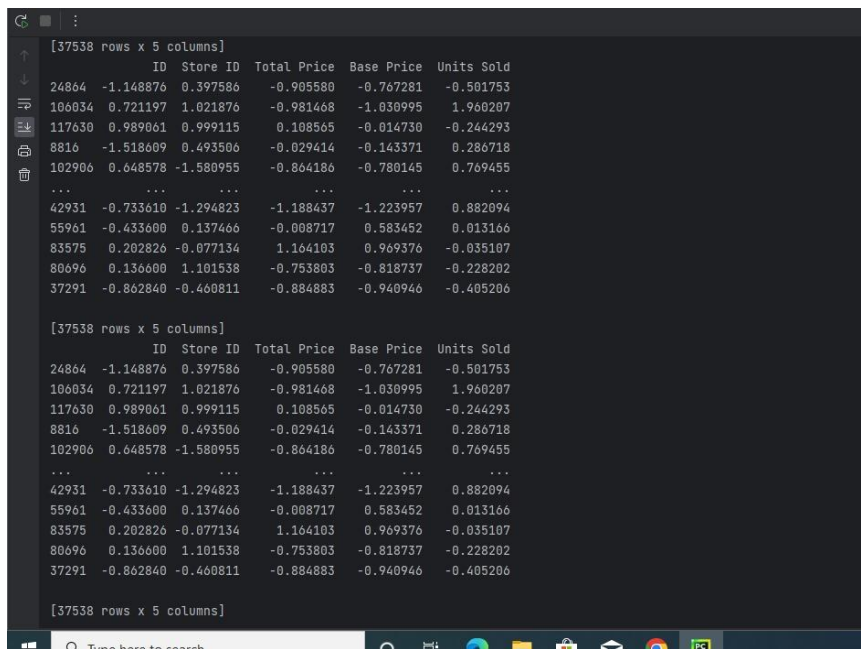
#train test

from sklearn.model_selection import train_test_split
x_train,x_test=train_test_split(data,test_size=0.25,random_state=1)
print(x_train)
print(x_test)

#feature scaling

from sklearn.preprocessing import StandardScaler
scalar=StandardScaler()
x_train[:]=scalar.fit_transform(x_train[:])
x_test[:]=scalar.fit_transform(x_test[:])
print(x_train)
print(x_test)
```

The output of the above code will be,



[37538 rows x 5 columns]

| | ID | Store ID | Total Price | Base Price | Units Sold |
|--------|-----------|-----------|-------------|------------|------------|
| 24864 | -1.148876 | 0.397586 | -0.905580 | -0.767281 | -0.501753 |
| 106034 | 0.721197 | 1.021876 | -0.981468 | -1.030995 | 1.960207 |
| 117630 | 0.989061 | 0.999115 | 0.108565 | -0.014730 | -0.244293 |
| 8816 | -1.518609 | 0.493506 | -0.029414 | -0.143371 | 0.286718 |
| 102906 | 0.648578 | -1.580955 | -0.864186 | -0.780145 | 0.769455 |
| ... | ... | ... | ... | ... | ... |
| 42931 | -0.733610 | -1.294823 | -1.188437 | -1.223957 | 0.882094 |
| 55961 | -0.433600 | 0.137466 | -0.008717 | 0.583452 | 0.013166 |
| 83575 | 0.202826 | -0.077134 | 1.164103 | 0.969376 | -0.035107 |
| 80696 | 0.136600 | 1.101538 | -0.753803 | -0.818737 | -0.228202 |
| 37291 | -0.862840 | -0.460811 | -0.884883 | -0.940946 | -0.405206 |

[37538 rows x 5 columns]

[37538 rows x 5 columns]

[37538 rows x 5 columns]

CONCLUSION:

Thus , we start our project development part 1 by loading the dataset and preprocessing it. By doing the next work by developing the model like feature engineering, model training, and evaluation we will predict the demand of proucts for the given dataset using machine learning.