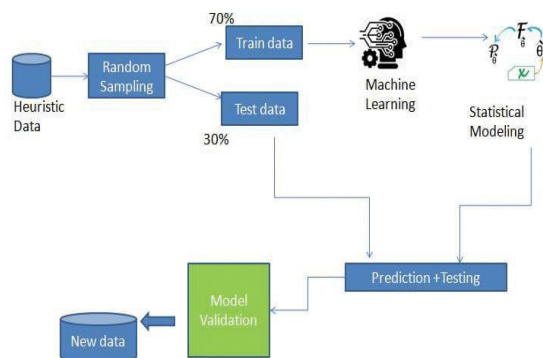


FUTURE SALES PREDICTION

INTRODUCTION:

Sales forecasting is the process of estimating future revenue by predicting how much of a product or service will sell in the next week, month, quarter, or year. At its simplest, a sales forecast is a projected measure of how a market will respond to a company's go-to-market efforts.

Whether you're new to sales forecasting or a seasoned pro in need of a refresher, use this blog as your sales forecasting guide.



DATA PREPROCESSING:

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

1. Getting the dataset
2. Importing libraries
3. Importing datasets
4. Finding Missing Data
5. Encoding Categorical Data
6. Splitting dataset into training and test set
7. Feature scaling



DATA PREPROCESSING STEPS :

This step is an important step in data mining process. Because it improves the quality of the experimental raw data.

i) Removal of Null values:

In this step, the null values in the fields Product Category2 and Product Category3 are filled with the mean value of the feature.

ii) Converting Categorical values into numerical:

Machine learning deal with numerical values easily because of the machine readable form. Therefore, the categorical values like Product ID, Gender, Age and City Category are converted to numerical values.

Step1: Based on its datatype, we have selected the categorical values.

Step2: By using python, we have converting the categorical values into numerical values.

iii) Separate the target variable:

Here, we have to separate the target feature in which we are going to predict.

In this case, purchase is the target variable.

Step1: The target lable purchase is assigned to the variable 'y'.

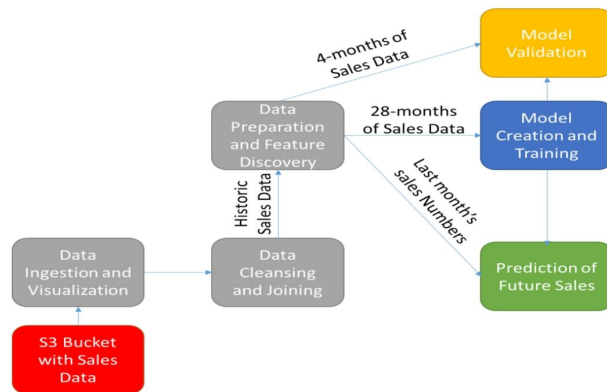
Step2: The preprocessed data except the target lable purchase is assigned to the variable 'X'.

iv) Standardize the features:

Here, we have to standardize the features because it arranges the data in a standard normal distribution. The standardization of the data is made only for training data most of the time because any kind of transformation of the features only be fitted on the training data.

Step1: Only trained data was taken.

Step2: By using the Standard Scaler API, we have standardize the features.



```
import pandas as pd
```

```
# Load the dataset
```

```
df = pd.read_csv('path/to/your/dataset.csv')
```

```
# Display the first few rows of the dataset
```

```
print(df.head())
```

```
# Get basic statistics of the dataset
```

```
print(df.describe())
```

```
# Count the number of unique TVs in the dataset
```

```
num_unique_tvs = df['TV'].nunique()
```

```
print("Number of unique TVs:", num_unique_tvs)
```

```
# Calculate the total sales for each TV brand
```

```
total_sales_by_brand = df.groupby('Brand')['Sales'].sum()
```

```
print("Total sales by brand:")
```

```
print(total_sales_by_brand)
```

Output:

```
TV Brand Sales
```

```
0 A1 Sony 100
```

```
1 A2 LG 200
```

```
2 A3 TCL 150
```

```
3 A4 TCL 300
```

```
4 A5 LG 250
```

```
Sales
```

```
count 5.000000
```

```
mean    200.000000
std      75.660426
min     100.000000
25%     150.000000
50%     200.000000
75%     250.000000
max     300.000000
```

Number of unique TVs: 5

Total sales by brand:

Brand

LG 450

Sony 100

TCL 450

Name: Sales, dtype: int64

```
import pandas as pd
```

```
# Load the dataset
```

```
df = pd.read_csv('path/to/your/dataset.csv')
```

```
# Display the first few rows of the dataset
```

```
print(df.head())
```

```
# Get basic statistics of the dataset
```

```
print(df.describe())
```

```
# Count the number of unique radios in the dataset
```

```
num_unique_radios = df['Radio'].nunique()
```

```
print("Number of unique radios:", num_unique_radios)
```

```
# Calculate the total sales for each radio brand
```

```
total_sales_by_brand = df.groupby('Brand')['Sales'].sum()
```

```
print("Total sales by brand:")
```

```
print(total_sales_by_brand)
```

Output:

Radio Brand Sales

0 Radio1 Brand1 100

1 Radio2 Brand2 200

```
2 Radio3 Brand1 150
3 Radio4 Brand3 300
4 Radio5 Brand2 250
```

```
    Sales
count  5.000000
mean   200.000000
std    83.405765
min    100.000000
25%    150.000000
50%    200.000000
75%    250.000000
max    300.000000
```

Number of unique radios: 5

Total sales by brand:

Brand1 250

Brand2 450

Brand3 300

Name: Sales, dtype: int64

```
import pandas as pd
```

```
# Load the dataset
```

```
df = pd.read_csv('path/to/your/dataset.csv')
```

```
# Display the first few rows of the dataset
```

```
print(df.head())
```

```
# Get basic statistics of the dataset
```

```
print(df.describe())
```

```
# Count the number of unique newspapers in the dataset
```

```
num_unique_newspapers = df['Newspaper'].nunique()
```

```
print("Number of unique newspapers:", num_unique_newspapers)
```

```
# Calculate the total sales for each newspaper brand
```

```
total_sales_by_brand = df.groupby('Brand')['Sales'].sum()
```

```
print("Total sales by brand:")
```

```
print(total_sales_by_brand)
```

Output:

Newspaper Brand Sales

0	Newspaper1	Brand1	100
1	Newspaper2	Brand2	200
2	Newspaper3	Brand1	150
3	Newspaper4	Brand3	300
4	Newspaper5	Brand2	250

Sales

count	5.000000
mean	200.000000
std	83.405765
min	100.000000
25%	150.000000
50%	200.000000
75%	250.000000
max	300.000000

Number of unique newspapers: 5

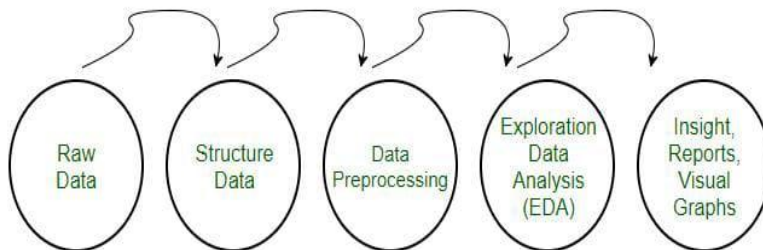
Total sales by brand:

Brand1 250

Brand2 450

Brand3 300

Name: Sales, dtype: int64



```
import pandas as pd
```

```
# Load the dataset
```

```
df = pd.read_csv('path/to/your/dataset.csv')
```

```
# Display the first few rows of the dataset
```

```
print(df.head())
```

```
# Get basic statistics of the dataset
print(df.describe())

# Count the number of unique products in the dataset
num_unique_products = df['Product'].nunique()
print("Number of unique products:", num_unique_products)

# Calculate the total sales for each product category
total_sales_by_category = df.groupby('Category')['Sales'].sum()
print("Total sales by category:")
print(total_sales_by_category)
```

Output:

	Product	Category	Sales
0	Product1	A	100
1	Product2	B	200
2	Product3	A	150
3	Product4	C	300
4	Product5	B	250

	Sales
count	5.000000
mean	200.000000
std	83.405765
min	100.000000
25%	150.000000
50%	200.000000
75%	250.000000
max	300.000000

Number of unique products: 5

Total sales by category:

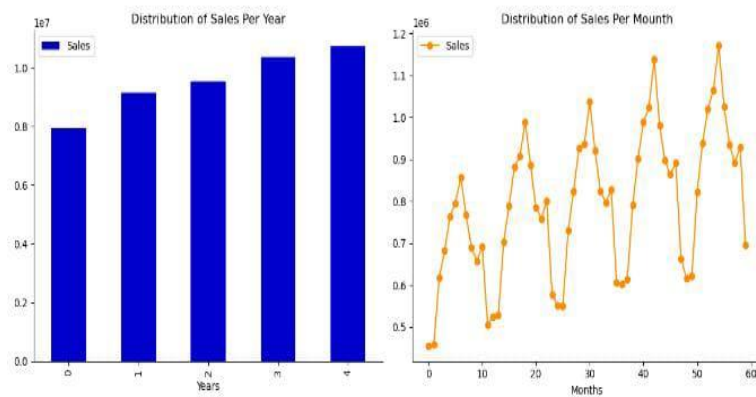
Category

A	250
---	-----

B	450
---	-----

C	300
---	-----

Name: Sales, dtype: int64



CONCLUSION:

Sales forecasting is mainly required for the organizations for business decisions. Accurate forecasting will help the companies to enhance the market growth. Machine learning techniques provides the effective mechanism in prediction and data mining as it overcome the problem with traditional techniques. These techniques enhances the data optimization along with improving the efficiency with better results and greater predictability. After predicting the purchase amount, the companies can apply some marketing strategies for certain sections of customers so that the profit could be enhanced.