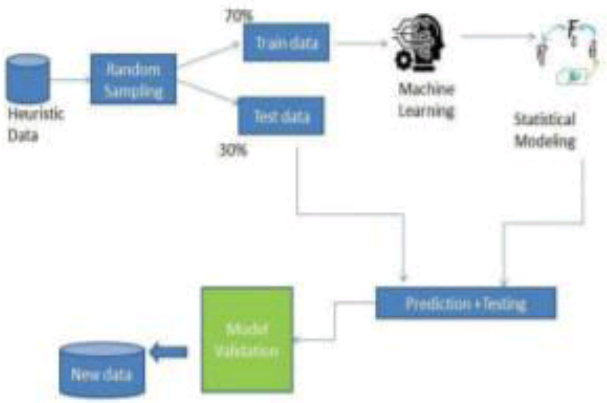
Future Sales Prediction

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'.

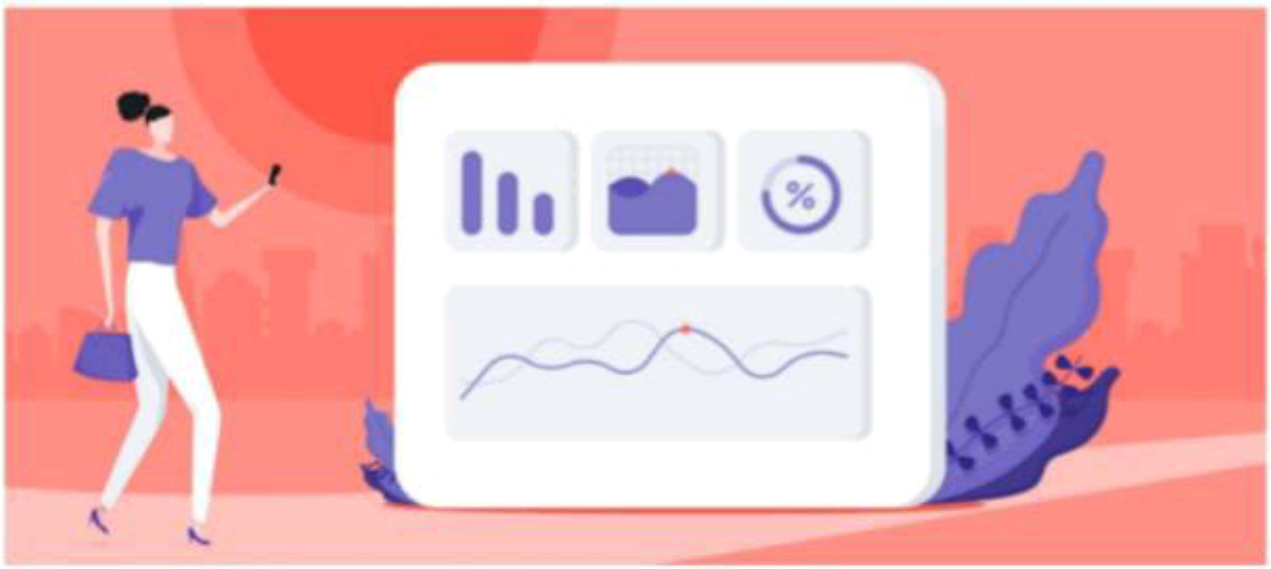
iv) Standardize the features:

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.


SALES PREDICTION




NEWS
NEWS

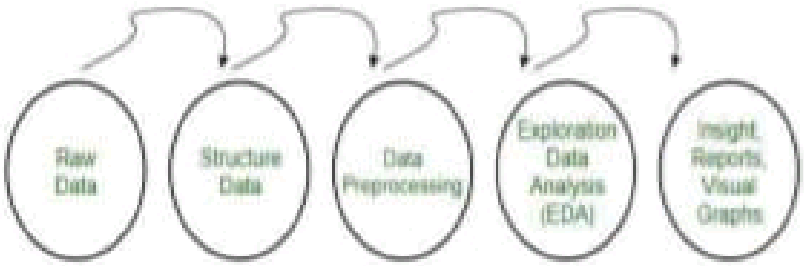
```
# Get basic statistics of the dataset
print(df.describe())

# 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
```

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())
```

```
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)
```

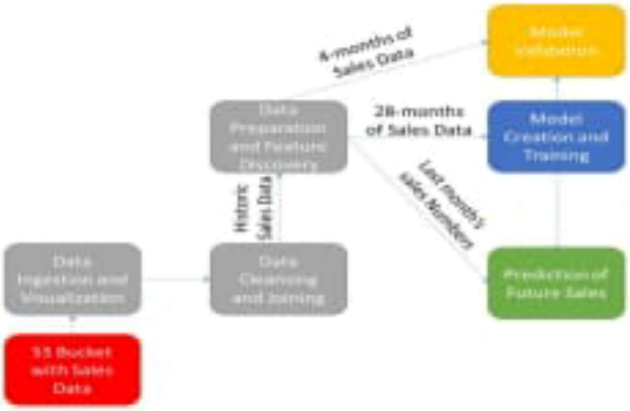
```
mean   200.000000
std     75.660426
min    100.000000
25%    150.000000
50%    200.000000
75%    250.000000
max    300.000000

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
```

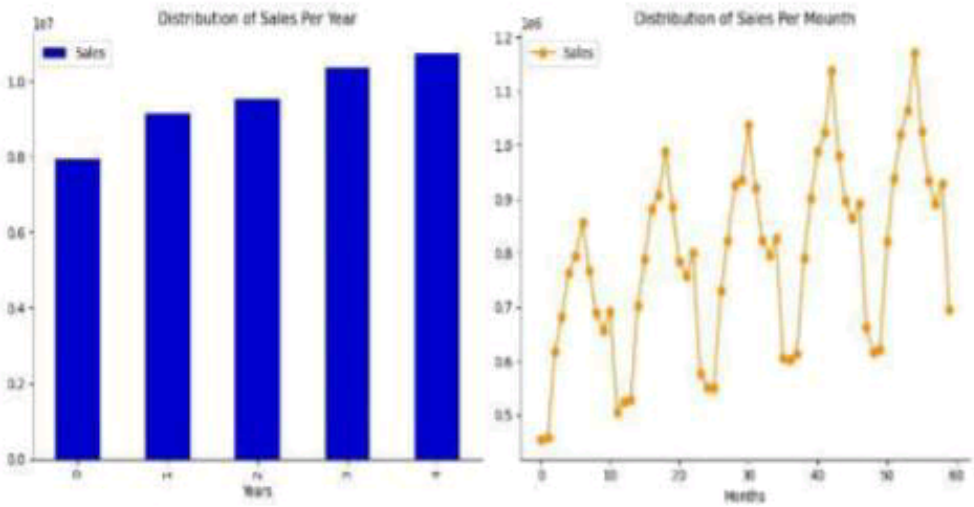

Predict sales revenue
Perceive challenges early
Motivate sales teams
Plan recruitment strategy
Aid future marketing plans




```
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
```


Distribution of Sales Per Year
1e7
Sales
Years
Distribution of Sales Per Mounth
1e6
Sales
Months

To train a model using the given dataset, you can follow these general steps:

1.Data Exploration: Start by exploring the dataset to understand its structure, features, and target variable. This will help you gain insights into the data and make informed decisions during the modeling process.

2.Data Preprocessing: Clean and preprocess the data to handle missing values, outliers, and categorical variables. This may involve techniques such as imputation, scaling, encoding, etc.

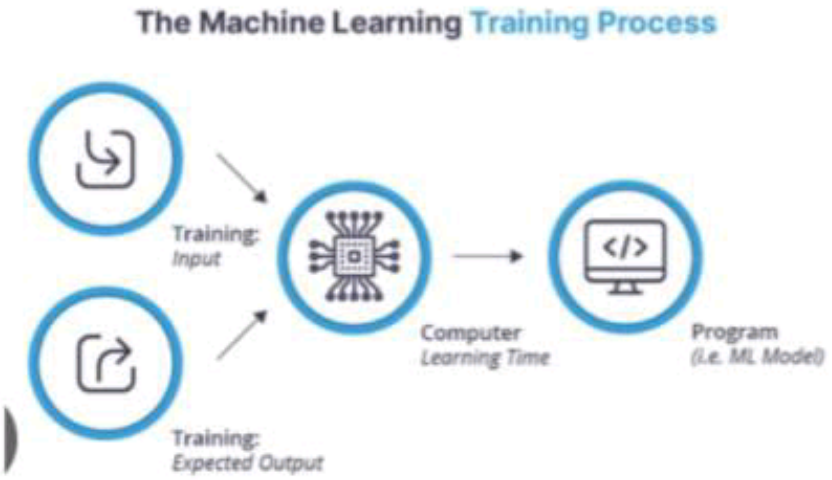
3.Feature Engineering: Create new features or transform existing ones to improve the predictive power of your model. This can include techniques like one-hot encoding, feature scaling, dimensionality reduction, etc.

4.Model Selection: Choose an appropriate machine learning algorithm based on your problem type (classification or regression) and requirements. Some popular algorithms for regression tasks include linear regression, decision trees, random forests, gradient boosting methods (e.g., XGBoost), etc.

5.Model Training: Split your dataset into training and validation sets. Use the training set to train your chosen model using appropriate hyperparameters. Evaluate the model's performance on the validation set using suitable evaluation metrics (e.g., mean squared error for regression).

6.Model Evaluation: Assess how well your trained model performs on unseen data by evaluating it on a separate test set or using cross-validation techniques. Compare different models if necessary and select the best-performing one.

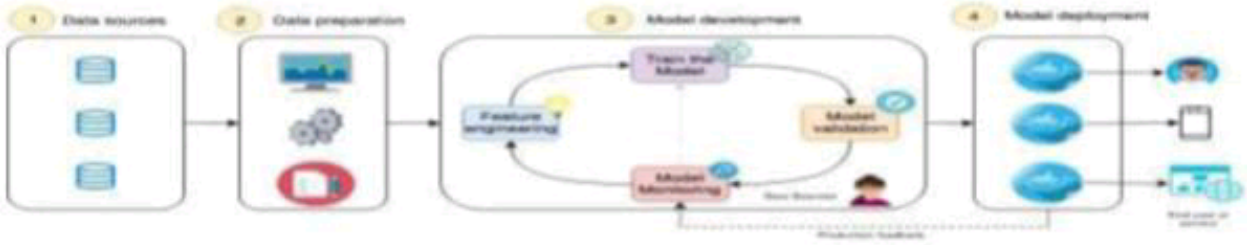
7.Hyperparameter Tuning: Fine-tune your chosen model by adjusting its hyperparameters to optimize its performance further. This can be done using techniques like grid search or random search.



Evaluation:

Once the model has been selected, it needs to be trained on the historical sales data. This involves splitting the data into training and testing sets, and using the training set to fit the model to the data. Once the model has been trained, it can be used to predict the demand for drugs in the pharmacy for a given period of time.

To evaluate the performance of the model, we can use metrics such as mean absolute percentage error (MAPE), root mean squared error (RMSE), mean absolute error (MAE), or R-squared(R^2). These metrics provide a measure of how well the model is able to predict the demand for drugs in the pharmacy. To evaluate the impact of feature engineering, you should compare the performance of models with and without engineered features using the evaluation metrics mentioned earlier. In most cases, you will likely observe that models with feature engineering outperform those without. Lower MAPE, RMSE, and MAE values and higher R^2 values are indicative of improved predictive accuracy.



To evaluate the given data set for future sales prediction, you would typically need to perform a series of steps. Here's a general outline of the evaluation process:

Predicting the future sales of a product helps a business manage the manufacturing and advertising cost of the product. There are many more benefits of predicting the future sales of a product. So if you want to learn to predict the future sales of a product with machine learning, this article is for you. In this article, I will take you through the task of future sales prediction with machine learning using Python.

Future Sales Prediction (Case Study)

The dataset given here contains the data about the sales of the product. The dataset is about the advertising cost incurred by the business on various advertising platforms. Below is the description of all the columns in the dataset:

TV: Advertising cost spent in dollars for advertising on TV;
Radio: Advertising cost spent in dollars for advertising on Radio;
Newspaper: Advertising cost spent in dollars for advertising on Newspaper;
Sales: Number of units sold;
So, in the above dataset, the sales of the product depend on the advertisement cost of the product. I hope you now have understood everything about this dataset. Now in the section below, I will take you through the task of future sales prediction with machine learning using Python.

Future Sales Prediction using Python


BUY

Let's start the task of future sales prediction with machine learning by importing the necessary Python libraries and the dataset:

```
1
import pandas as pd
2
import numpy as np
3
from sklearn.model_selection import train_test_split
4
from sklearn.linear_model import LinearRegression
5

6
data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/advertising.csv")
7
print(data.head())
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3   12.0
3  151.5   41.3       58.5   16.5
4  180.8   10.8       58.4   17.9
```

Let's have a look at whether this dataset contains any null values or not:

1

```
print(data.isnull().sum())
```

```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

So this dataset doesn’t have any null values.


thank you