**SALES PREDICTION PROJECT**

**1.Objective:**

The objective of this project is to predict sales based on advertising expenditures across different media channels (TV, Radio, and Newspaper).

**2.Load the Dataset and Import Necessary Packages:**

The dataset used for this project is `Advertising.csv`, which contains data on advertising budgets and corresponding sales figures.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

import warnings

warnings.filterwarnings("ignore")

The necessary packages for data manipulation, visualization, and model building are imported.

Loading the Dataset:

sales\_data = pd.read\_csv('Advertising.csv')

sales\_data.head()

sales\_data.info()

The dataset is loaded into a pandas DataFrame, and the first few rows are displayed along with the dataset's information.

**3.Checking for Null and Duplicate Values:**

sales\_data.duplicated().sum()

sales\_data.isnull().sum()

The dataset is checked for any duplicate or null values, ensuring data integrity. No duplicates or null values were found.

**4.Dropping Unnecessary Columns:**

sales\_data.drop(columns=['Unnamed: 0'], axis=1, inplace=True)

The unnecessary column `Unnamed: 0` is dropped from the dataset.

**5.Correlation Analysis:**

plt.figure(figsize=(10,8))

sns.heatmap(data=sales\_data.corr(), annot=True, vmin=-1, cmap='seismic')

plt.title('Correlation between Attributes')

plt.show()

A heatmap is plotted to visualize the correlation between different attributes in the dataset. This helps in understanding the relationship between the features and the target variable (Sales).

**6.Data Visualization:**

sns.scatterplot(data=sales\_data, x='Newspaper', y='Sales', palette='Set2')

sns.scatterplot(data=sales\_data, x='Radio', y='Newspaper', hue='Sales', label=0)

sns.pairplot(data=sales\_data)

fig, axs = plt.subplots(3, figsize=(5, 5))

plt1 = sns.boxplot(sales\_data['TV'], ax=axs[0])

plt2 = sns.boxplot(sales\_data['Newspaper'], ax=axs[1])

plt3 = sns.boxplot(sales\_data['Radio'], ax=axs[2])

plt.tight\_layout()

Various visualizations are created to understand the distribution and relationship of features:

- Scatter plots to see the relationship between individual features and sales.

- Pair plots to visualize pairwise relationships in the dataset.

- Box plots to identify outliers in the dataset.

**7.Feature Selection:**

from sklearn.model\_selection import train\_test\_split

X = sales\_data.drop(columns=['Sales'])

y = sales\_data['Sales']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=28)

The dataset is split into features (`X`) and the target variable (`y`). Then, the data is further split into training and testing sets using a 70-30 split.

**8.Model Training:**

from sklearn.linear\_model import LinearRegression

linear\_model = LinearRegression()

linear\_model.fit(X\_train, y\_train)

A linear regression model is created and trained on the training data.

**9.Model Evaluation:**

linear\_prediction = linear\_model.predict(X\_test)

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

mean\_abs\_error = mean\_absolute\_error(y\_test, linear\_prediction)

print("Mean Absolute Error: ", mean\_abs\_error)

mean\_sqr\_error = mean\_squared\_error(y\_test, linear\_prediction)

print("Mean Squared Error: ", mean\_sqr\_error)

The model's performance is evaluated using Mean Absolute Error (MAE) and Mean Squared Error (MSE), which measure the average errors in the predictions.

**10.Predicting Sales for User-Defined Values:**

def predict\_selling\_price(model, input\_data):

input\_df = pd.DataFrame([input\_data])

input\_df = pd.get\_dummies(input\_df, drop\_first=True).reindex(columns=X.columns, fill\_value=0)

prediction = model.predict(input\_df)

return prediction[0]

user\_input = {

'TV': float(input("Enter the Price of TV: ")),

'Radio': float(input("Enter the Price of Radio: ")),

'Newspaper': float(input("Enter the Price of Newspaper: "))

}

predicted\_price = predict\_selling\_price(linear\_model, user\_input)

print(f"Predicted Sales Price: {predicted\_price}")

A function is created to predict sales based on user-defined advertising expenditures. The user inputs the budget for TV, Radio, and Newspaper advertising, and the function returns the predicted sales.

**11.Conclusion:**

This project demonstrates the process of building a sales prediction model using linear regression. The steps include data preprocessing, visualization, feature selection, model training, and evaluation. The final model can predict sales based on advertising budgets in different media channels.