SALES PREDICTION USING PYTHON

Project Overview:

Sales prediction is an essential aspect of business strategy, helping organizations make informed decisions about marketing budgets, inventory management, and revenue forecasting. This project leverages a dataset containing advertising data to build a machine learning model that predicts sales based on the advertising expenditure across TV, Radio, and Newspaper channels.

The project follows a structured approach to:

1. Understand the dataset.
2. Preprocess the data.
3. Explore data visualization for insights.
4. Build a regression model for prediction.
5. Evaluate the model's performance.
6. Predict sales on new advertising data.

Introduction:

Sales prediction models aim to estimate future sales based on historical data. In this project, we use a dataset with advertising expenditure and corresponding sales data. The goal is to determine how investments in various media (TV, Radio, Newspaper) influence sales.

Dataset Overview:

- The dataset (advertising.csv) contains the following columns:
- TV: Advertising budget spent on TV (in thousands of dollars).
- Radio: Advertising budget spent on Radio (in thousands of dollars).
- Newspaper: Advertising budget spent on Newspapers (in thousands of dollars).
- Sales: Product sales generated (in thousands of units).

	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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

Import Libraries:

❖ We use essential Python libraries for data manipulation, visualization, and machine learning.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

Load and Explore Dataset:

❖ The dataset is loaded into a Pandas DataFrame for exploration.

```
# Load the dataset
data = pd.read_csv('advertising.csv')

# Display the first few rows
print(data.head())

# Check for missing values
print(data.isnull().sum())

# Statistical summary of the dataset
print(data.describe())
```

Key Insights:

- ❖ The dataset is clean, with no missing values.
- ❖ All features are numerical and suitable for regression analysis.

Data Preprocessing:

- ❖ Correlation analysis helps identify the relationship between features and the target variable (Sales).

```
# Correlation matrix
print(data.corr())

# Select Features (TV, Radio, Newspaper) and Target (Sales)
X = data[['TV', 'Radio', 'Newspaper']]  # Features
y = data['Sales']  # Target
```

Data Visualization:

- ❖ Visualizations provide insights into relationships between features and sales.

```
# Correlation Heatmap
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# Pairplot
sns.pairplot(data)
plt.show()

# Distribution of Sales
sns.histplot(data['Sales'], kde=True, bins=20)
plt.title('Distribution of Sales')
plt.show()
```

Observations:

- ❖ Sales are highly correlated with TV and Radio budgets, while the Newspaper budget has a weaker correlation.
- ❖ Sales data follows a normal-like distribution.

Model Building:

- We split the dataset into training and testing sets and train a linear regression model.

```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize and train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Model Evaluation:

- Model evaluation metrics like Mean Squared Error (MSE) and R-squared are calculated.

```
# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

Overall Program:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
data = pd.read_csv('advertising.csv')

# Display dataset info
print("First five rows of the dataset:\n", data.head())
print("\nDataset summary:\n", data.describe())
```

```
# Check for missing values
print("\nMissing values in dataset:\n", data.isnull().sum())

# Data Visualization
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

sns.pairplot(data)
plt.show()

sns.histplot(data['Sales'], kde=True, bins=20)
plt.title('Distribution of Sales')
plt.show()

# Selecting features and target variable
X = data[['TV', 'Radio', 'Newspaper']]
y = data['Sales']

# Splitting dataset into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Model Training
model = LinearRegression()
model.fit(X_train, y_train)

# Model Evaluation
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"\nModel Performance:")
print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")

# Predicting sales for a new advertising budget
new_data = pd.DataFrame({'TV': [150], 'Radio': [20], 'Newspaper': [25]})
predicted_sales = model.predict(new_data)

print(f"\nPredicted Sales for new ad campaign: {predicted_sales[0]}")
```

Results:

- ❖ MSE: Measures the average squared difference between actual and predicted values.
- ❖ R-squared: Indicates the proportion of variance in the target variable explained by the model.

Predictions on New Data

- ❖ We predict sales for a new advertising campaign.

```
# Predict sales for a new advertising campaign
new_data = pd.DataFrame({'TV': [150], 'Radio': [20], 'Newspaper': [25]})
predicted_sales = model.predict(new_data)

print(f"Predicted Sales: {predicted_sales[0]}")
```

Conclusion:

➔ **Insights from Data Analysis:**

- ❖ TV and Radio advertising budgets have a significant positive impact on sales.
- ❖ Newspaper advertising has a weaker correlation with sales.

➔ **Model Performance:**

- ❖ The linear regression model provides a reliable prediction of sales based on advertising budgets.
- ❖ Performance metrics indicate a good fit, though there may be room for improvement with more complex models.

➔ **Future Improvements:**

- ❖ Incorporate additional features such as market trends, seasonal data, or competitor analysis.
- ❖ Experiment with advanced models like Random Forest, Gradient Boosting, or Neural Networks.
- ❖ Perform hyperparameter tuning for improved model accuracy.