PROJECT 2

Optimizing Sales Forecasting: Predictive Modeling with Linear Regression

Artificial Intelligence & Machine Learning

Table Of Content

Submitted By

PRAKASH N

E-mail ID:*prakashpoint2005@gmail.com*

1. Aim

The aim of this project is to develop a linear regression model to predict sales based on advertising expenditure across three mediums: TV, Radio, and Newspaper.

2. Introduction

In the competitive world of advertising, companies need to allocate their budgets efficiently to maximize sales. By analyzing historical data, we can create predictive models to estimate the impact of advertising spends on sales. This project uses a dataset containing advertising expenses and corresponding sales to build and evaluate a linear regression model.

3. Technical Requirements

Hardware Specifications

- **Processor**: Intel Core i5 or higher
- **Memory**: 8 GB RAM or higher
- **Storage**: 256 GB SSD or higher
- **Graphics**: Integrated graphics card sufficient for data visualization tasks

Software Requirements and Environment Setup

- **Operating System**: Windows 10, macOS, or Linux
- **Programming Language**: Python 3.x
- **Development Environment**:
    - **Primary IDE**: Visual Studio Code (VS Code) with Jupyter Notebook integration
    - **Alternative**: Jupyter Notebook standalone, Google Colab

Libraries

- **Data Handling and Analysis**: pandas, numpy
- **Data Visualization**: matplotlib, seaborn
- **Machine Learning**: scikit-learn
- **Training Module**: LinearRegression from the scikit-learn library.

VS Code Extensions

- **Python Extension for Visual Studio Code**: Microsoft's official extension for Python development
- **Jupyter Extension for Visual Studio Code**: Provides support for Jupyter Notebooks inside VS Code (Microsoft's official extension for Jupyter Notebooks like Jupyter, Jupyter Keymap, Jupyter Power Toys)

Suggested Tools / Tech Stacks

1. **Python Programming Language**
2. **Jupyter Notebook** for interactive data analysis

3. **Visual Studio Code (VS Code)** as a versatile IDE with Jupyter support
4. **Essential libraries** for data handling, visualization, and machine learning
5. **Git** for version control and collaboration

4. Modeling Approach

The project employs a simple linear regression model to predict sales based on advertising spends. Linear regression is chosen for its simplicity and interpretability. The approach involves the following steps:

1. Exploratory Data Analysis (EDA) to understand the data.
2. Splitting the data into training and testing sets.
3. Training the linear regression model on the training set.
4. Making predictions on the test set.
5. Evaluating the model's performance using appropriate metrics.

5. Implementation Steps

Step 1: Load the Dataset

First, we load the dataset using pandas, a powerful library for data manipulation and analysis. The dataset contains columns for TV, Radio, Newspaper, and Sales, which we will use for our analysis and model building.

```
import pandas as pd

data = pd.read_csv('/path/to/Advertising Dataset.csv', index_col=0)
print(data.head())
print(data.tail())
print(data.shape)
```

Step 2: Data Visualization

Next, we use seaborn and matplotlib to visualize the relationships between the advertising expenditures (TV, Radio, Newspaper) and sales. Pair plots provide a convenient way to see if linear relationships exist between our predictors and the response variable.

```
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
sns.pairplot(data, x_vars=["TV", "Radio", "Newspaper"], y_vars='Sales', height=7, aspect=0.7, kind="reg")
plt.show()
```

Step 3: Define Features and Target Variable

We then define the features (independent variables) and the target variable (dependent variable). This prepares the data for model training.

```
features = ["TV", "Radio", "Newspaper"]
x = data[features]
y = data["Sales"]
```

Step 4: Split the Data into Training and Testing Sets

We split the dataset into training and testing sets to evaluate our model's performance. The training set will be used to train the model, while the test set will be used to assess how well the model generalizes to unseen data.

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=1)
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

Step 5: Train the Linear Regression Model

Using the training set, we train a linear regression model. We then print the intercept and coefficients of the model, which tell us the baseline sales and the impact of each advertising medium on sales, respectively.

```
from sklearn.linear_model import LinearRegression

linreg = LinearRegression()
linreg.fit(x_train, y_train)
print("Intercept:", linreg.intercept_)
print("Coefficients:", linreg.coef_)
print(list(zip(features, linreg.coef_)))
```

Step 6: Make Predictions on the Test Set

We use the trained model to make predictions on the test set. By comparing the predicted values with the actual values, we can assess the model's accuracy.

```
y_pred = linreg.predict(x_test)
compare_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
compare_df.reset_index(drop=True, inplace=True)
compare_df['Difference'] = compare_df['Actual'] - compare_df['Predicted']
print(compare_df)
```

Step 7: Evaluate the Model

```
from sklearn import metrics

mae = metrics.mean_absolute_error(y_test, y_pred)
mse = metrics.mean_squared_error(y_test, y_pred)
r2 = metrics.r2_score(y_test, y_pred)

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

The evaluation metrics give insights into the model's accuracy:

- **Mean Absolute Error (MAE)**: Measures the average magnitude of errors in a set of predictions, without considering their direction.
- **Mean Squared Error (MSE)**: Measures the average of the squares of the errors. It's more sensitive to outliers than MAE.
- **R-squared (R^2)**: Represents the proportion of variance in the dependent variable that can be explained by the independent variables. An R^2 of 1 indicates perfect prediction.

6. Expected Output

- A trained linear regression model capable of predicting sales based on TV, Radio, and Newspaper advertising spends.
- Visualization of the data and regression lines.
- Evaluation metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) value.

7. Conclusion

This project demonstrates the use of linear regression to model the relationship between advertising expenditures and sales. The model provides insights into how different advertising channels impact sales by analyzing historical data. The evaluation metrics indicate the model's accuracy and can help in making informed budgeting decisions for future advertising campaigns.

8. Reference

- **Advertisement Observations Dataset**
- **Link:** https://drive.google.com/file/d/1q7WhzzhfDeryMDNvVXpVyUGuSlJRi8i5/view?usp=sharing
- **My Project Location**
- **Link:** https://drive.google.com/file/d/1gmjHu-YlpXblI-IAVhe42mykeIbN_AQe/view?usp=sharing
- Scikit-learn documentation: https://scikit-learn.org/stable/user_guide.html
- Pandas documentation: https://pandas.pydata.org/pandas-docs/stable/
- Seaborn documentation: https://seaborn.pydata.org/
- Linear Regression: https://en.wikipedia.org/wiki/Linear_regression