Multiple Linear Regression .

Introduction

Multiple Linear Regression is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. This guide will take you through the steps required to create a Multiple Linear Regression model using Python. The steps include data preparation, model building, evaluation, and interpretation.

Step 1: Import Necessary Libraries

The first step is to import the necessary libraries that will help in data manipulation, model building, and evaluation. Here are the primary libraries you'll need:

- import pandas as pd (pandas are used for data manipulation)
- import numpy as np (numpy used for numerical operations)
- from sklearn.model_selection import train_test_split (For splitting the data)
- from sklearn.linear_model import LinearRegression (For building the regression model)
- from sklearn.metrics import mean_squared_error, r2_score (For evaluating the model)

Step 2: Load and Explore the Dataset

Load the dataset into a Pandas DataFrame and explore it to understand its structure, missing values, and basic statistics.

- Load dataset

```
data = pd.read_csv('your_dataset.csv')
```

- Display the first few rows

```
print(data.head())
```

- Display basic statistics

```
print(data.describe())
```

Step 3: Preprocess the Data

Data preprocessing includes handling missing values, encoding categorical variables, and scaling the data if necessary. Ensure the data is clean and ready for modeling.

- Handling missing values

data = data.dropna() (Drop rows with missing values)

Step 4: Define Features and Target Variable

Separate the independent variables (features) and the dependent variable (target). Ensure your target variable is the one you want to predict.

- Define features and target variable

X = data[['feature1', 'feature2', 'feature3']] (Replace with your feature names)
y = data['target'] (Replace with your target variable)

Step 5: Split the Data into Training and Testing Sets

Split the dataset into training and testing sets to evaluate the model's performance on unseen data.

- Split the data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

Step 6: Train the Multiple Linear Regression Model

Fit the Linear Regression model using the training data.

- Create the model

model = LinearRegression()

- Train the model

model.fit(X_train, y_train)
```

**Step 7: Evaluate the Model**

Use evaluation metrics like Mean Squared Error (MSE) and R-squared to assess the model's performance.
```

- Predict on the test set

```
y_pred = model.predict(X_test)
```

- Calculate evaluation metrics

```
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}')
```

Step 8: Interpret the Results

Interpret the coefficients of the model to understand the relationship between the features and the target variable. The coefficients represent the change in the target variable for a one-unit change in the feature.

- Display coefficients

```
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print(coefficients)
```

Step 9: ConclusionIn this we removed 3 columns because the data is not necessary.

.Car purchase amount is the independent column and Country,gender,AgE,Annual Salary,credit Card ,Debt networth is the dependent column.

Train Accuracy : -> 0.9999999812764105

Train Loss : 2.1990016683907267

train loss using mean absolute 1.176518760849249

Test Accuracy : -> 0.9999999806028682

Test Loss : 2.0943696032432655

test loss using mean absolute 1.153570893924407