MIT World Peace University

Data Science for Cybersecurity and Forensics
Third Year B. Tech, Semester 6

Simple Linear Regression

Assignment 5

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

April 16, 2024

Contents

1 Aim

To implement a simple linear regression model using Python.

2 Objectives

1. To understand the concept of simple linear regression.
2. To implement a simple linear regression model using Python.

3 Theory

3.1 Machine Learning

Machine learning is a subset of artificial intelligence that focuses on developing algorithms and models that allow computers to learn from and make predictions or decisions based on data. Machine learning algorithms can be broadly categorized into three types: supervised learning, unsupervised learning, and reinforcement learning.

3.2 Supervised Learning

Supervised learning is a type of machine learning where the model is trained on a labeled dataset, meaning that the input data is paired with the correct output or target variable. The goal of supervised learning is to learn a mapping function from the input to the output, so that the model can make predictions on new, unseen data.

3.3 Regression

Regression is a type of supervised learning algorithm that is used to predict continuous values or quantities. In regression, the goal is to learn a mapping function from the input features to a continuous output variable. Simple linear regression is a type of regression that models the relationship between a single input feature and a continuous output variable.

3.4 Simple Linear Regression

Simple linear regression is a statistical method that models the relationship between a single input feature (independent variable) and a continuous output variable (dependent variable). The relationship is modeled as a straight line, with the equation of the line given by:

$$y = mx + c$$

where y is the dependent variable, x is the independent variable, m is the slope of the line, and c is the y-intercept. The goal of simple linear regression is to find the values of m and c that best fit the data, so that the model can make accurate predictions on new data.

3.5 Fitting a Simple Linear Regression Model

To fit a simple linear regression model, we use the least squares method, which minimizes the sum of the squared differences between the observed values and the values predicted by the model. The formula for the slope m and y-intercept c of the line are given by:

$$m = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$c = \frac{\sum y - m(\sum x)}{n}$$

where n is the number of data points, $\sum x$ is the sum of the input feature values, $\sum y$ is the sum of the output variable values, $\sum xy$ is the sum of the product of the input feature and output variable values, and $\sum x^2$ is the sum of the squared input feature values.

3.6 Predicting with a Simple Linear Regression Model

Once the model has been trained and the values of m and c have been determined, we can use the model to make predictions on new, unseen data. To predict the output variable value for a given input feature value, we substitute the input feature value into the equation of the line:

$$y_{\text{pred}} = mx_{\text{new}} + c$$

where y_{pred} is the predicted output variable value, x_{new} is the input feature value, m is the slope of the line, and c is the y-intercept.

3.7 Evaluating a Simple Linear Regression Model

To evaluate the performance of a simple linear regression model, we can use metrics such as the mean squared error (MSE) and the coefficient of determination (R^2). The mean squared error measures the average squared difference between the observed and predicted values, while the coefficient of determination measures the proportion of the variance in the output variable that is predictable from the input feature. Their formulas are given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where n is the number of data points, y_i is the observed output variable value, $\hat{y}_i$ is the predicted output variable value, and $\bar{y}$ is the mean of the output variable values.

4 Procedure

1. Import the required python packages.
2. Load the dataset.
3. Data analysis.
4. Split the dataset into dependent/independent variables.
5. Split data into Train/Test sets.
6. Train the regression model.
7. Predict the result.

5 Platform

Operating System: Windows 11
IDEs or Text Editors Used: Visual Studio Code
Compilers or Interpreters: Python 3.10.1

6 Requirements

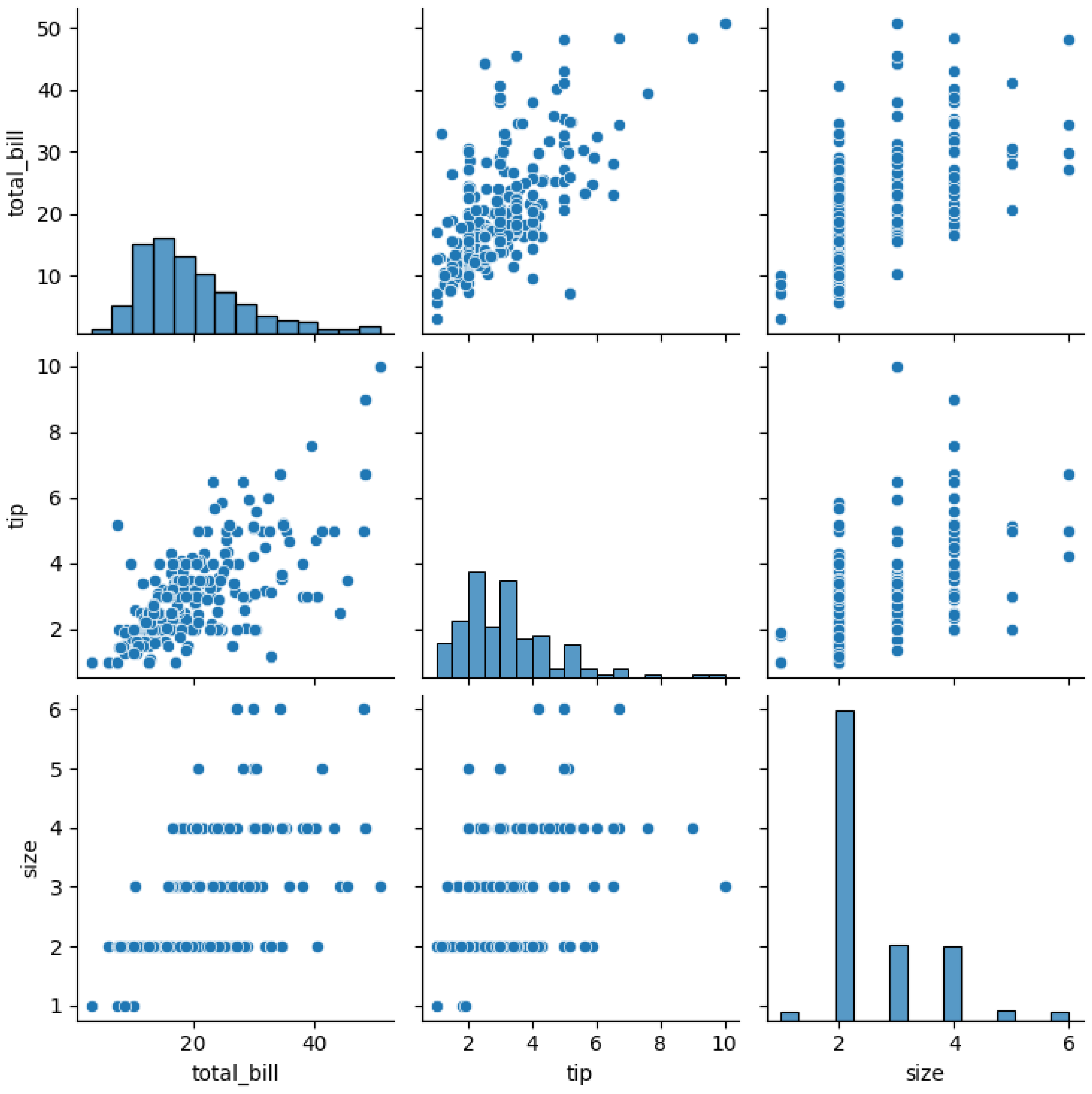
```
python==3.10.1
matplotlib==3.8.3
numpy==1.26.4
pandas==2.2.2
seaborn==0.13.2
scikit-learn==3.3.0
```

7 Code

[7]:
```
# Step 1: Import the required Python packages
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

[8]:
```
# Step 2: Load the dataset
df = sns.load_dataset("tips")
```

[9]:
```
# Step 3: Data analysis (optional)
# For example, you can visualize the relationship between variables using␣
↪Seaborn
sns.pairplot(df)
plt.show()
```



[10]:
```
# Step 4: Split the dataset into dependent/independent variables
X = df[["total_bill"]]  # Independent variable
y = df["tip"]  # Dependent variable
```

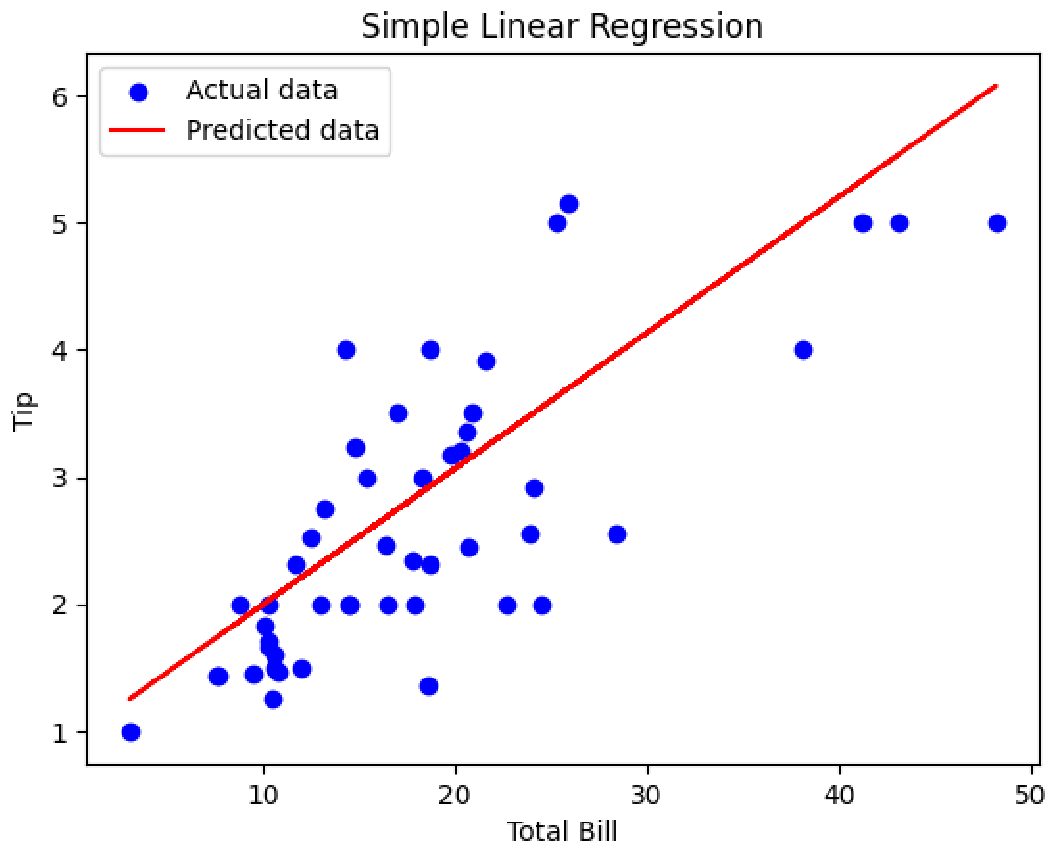
[11]:
```
# Step 5: Split data into Train/Test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

[12]:
```
# Step 6: Train the regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

[12]:
```
LinearRegression()
```

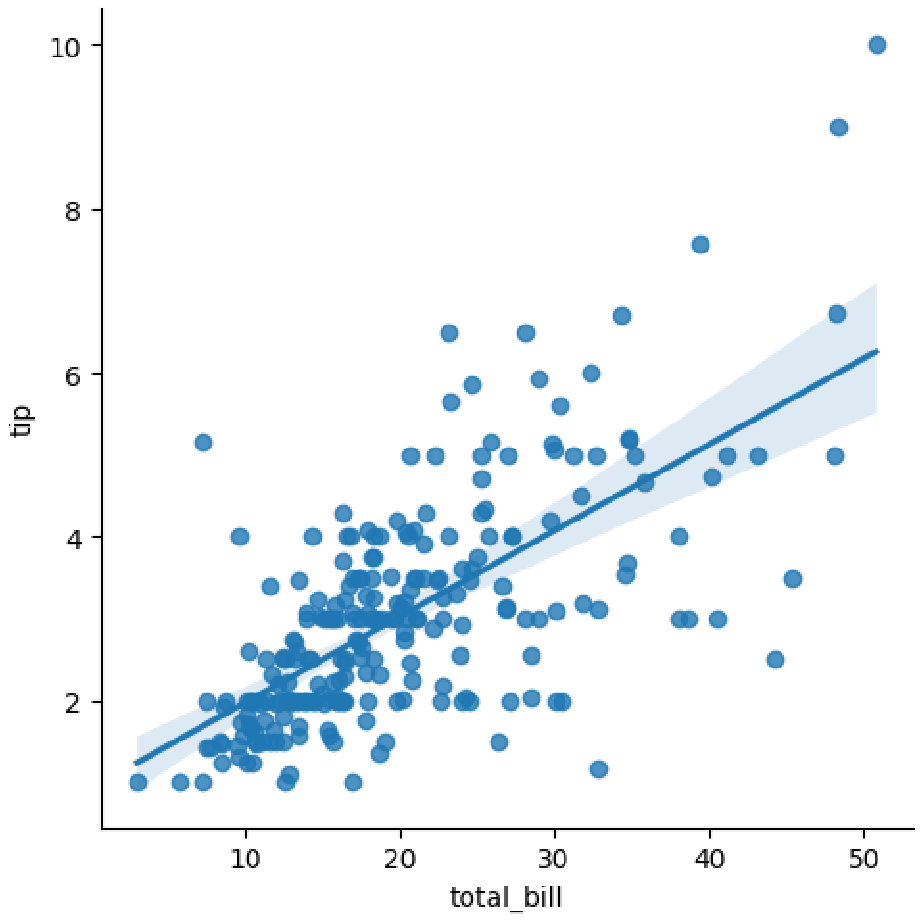
[13]:
```
# Step 7: Predict the result
y_pred = model.predict(X_test)
```

[14]:
```
# Visualize the predicted results against the actual data
plt.scatter(X_test, y_test, color="blue", label="Actual data")
plt.plot(X_test, y_pred, color="red", label="Predicted data")
plt.xlabel("Total Bill")
plt.ylabel("Tip")
plt.title("Simple Linear Regression")
plt.legend()
plt.show()
```



[15]:
```
# use lmplot from seaborn to plot the regression line
sns.lmplot(x="total_bill", y="tip", data=df)
```

[15]:
```
<seaborn.axisgrid.FacetGrid at 0x1ec8bba5180>
```


10
8
6
4
2
tip
10
20
30
40
50
total_bill

8 FAQs

1. **What is Linear Regression?**
   Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The equation takes the form $y = mx+b$, where y is the dependent variable, x is the independent variable, m is the slope, and b is the intercept. The goal of linear regression is to find the best-fitting line that minimizes the sum of the squared differences between the observed and predicted values.

2. **How does Linear Regression help in cybersecurity and forensics?**
   Linear regression is utilized in cybersecurity and forensics for various purposes:

   - **Anomaly Detection**: Linear regression can be used to identify anomalous behavior or deviations from normal patterns in network traffic, system logs, or user behavior. By analyzing historical data, linear regression models can detect unusual patterns that may indicate security breaches or cyberattacks.
   - **Intrusion Detection**: Linear regression models can be trained on features extracted from network traffic or system logs to classify incoming data as normal or malicious. By learning the underlying patterns of normal behavior, linear regression models can help detect and prevent cyber threats in real-time.
   - **Digital Forensics**: Linear regression can assist digital forensics analysts in investigating cybercrimes by analyzing large volumes of data collected from digital devices or networks. By correlating various factors such as timestamps, file sizes, and access patterns, linear regression models can provide insights into the sequence of events and potential perpetrators involved in cyber incidents.
   - **Risk Assessment**: Linear regression can be used to assess the risk of cyber threats and vulnerabilities in organizational systems and infrastructure. By analyzing historical data on security incidents, linear regression models can predict the likelihood of future security breaches and help prioritize mitigation efforts and resource allocation.

9 Conclusion

In this assignment, we implemented a simple linear regression model using Python. We loaded a dataset, performed data preprocessing, split the data into training and testing sets, trained the regression model, and made predictions on new data. We also evaluated the performance of the model using metrics such as the mean squared error and the coefficient of determination. By implementing a simple linear regression model, we gained a better understanding of how regression models work and how they can be used to make predictions on continuous data.