Sardar Patel Institute of Technology, Mumbai
Department of Electronics and Telecommunication Engineering
T.E.-CSE (2023-2024)
PC 307 - Machine Learning

Name: Deepali Daga **Date:2-02-24**
UID: 2021600012 **Batch: A**

Experiment 2: Regression

AIM/OBJECTIVE: Implement Simple regression technique (Linear and Least square) using open-source software**.**

Outcomes:

- **Explore the Dataset suitable for regression problem**
- **Explore the hidden pattern from the dataset and apply suitable algorithm**

System Requirements:

Linux OS with Python and libraries or R or windows with MATLAB

Theory: Part I: (Linear regression and Ordinary Least square Regression)

ALGORITHM:
Step 1: Create Database for Linear Regression
Step 2: Finding Hypothesis of Linear Regression
Step 3: Training a Linear Regression model
Step 4: Evaluating the model
Step 5: Scikit-learn implementation
Step 6: End

Libraries:

1) Matplotlib 2) Seaborn 3) Pandas 4) Numpy

dataset link and description:

A	B
Distance (miles)	Speed (mph)
1	3.8
2	4.5
4	5.2
4	5.8
5	6
6	7.5
7	7.8
8	8
10	9.2
10	9
11	10.5
12	11
9	11.8
14	12.5
15	13.2
16	13
17	14.5
18	15.2
19	15.8
20	16.5

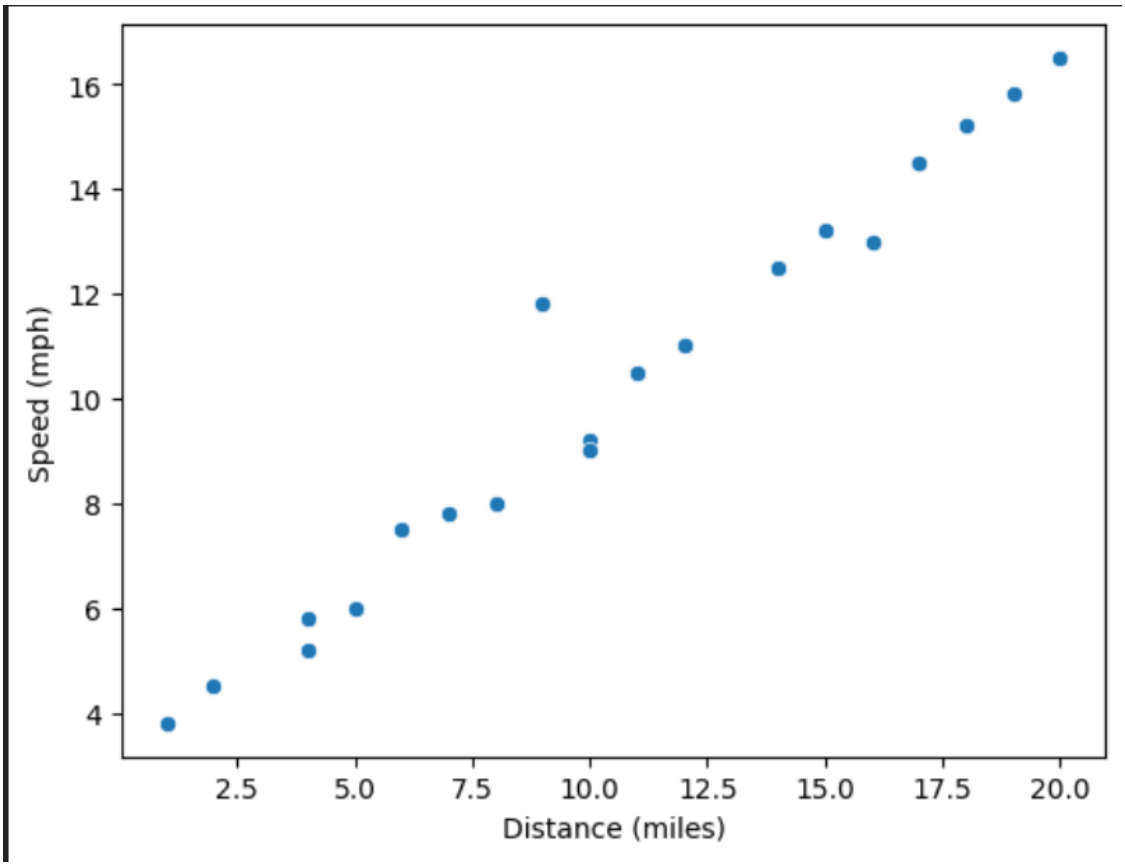
Created a dummy dataset for speed and distance.

Program:

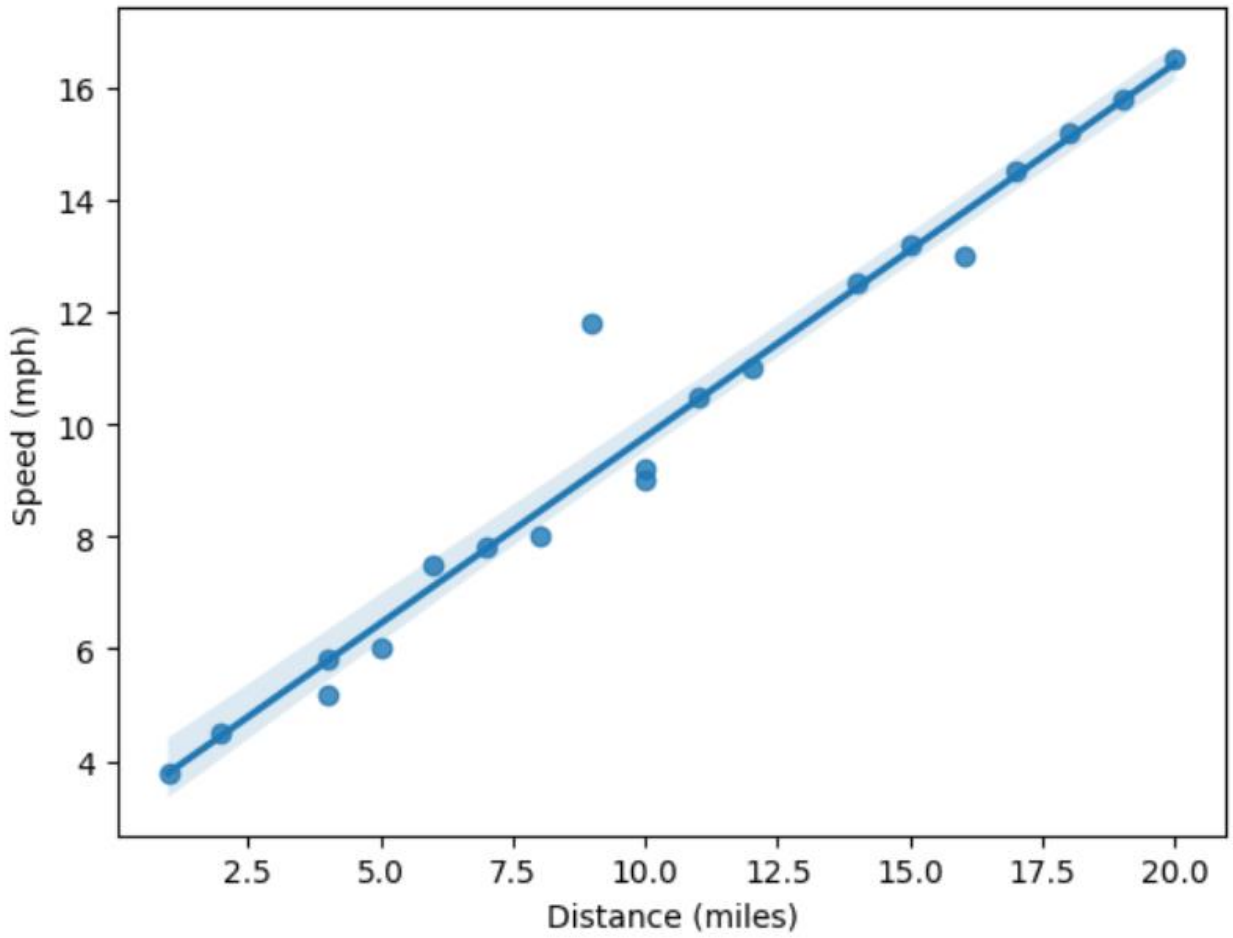
```
import numpy as np
import matplotlib.pyplot as plt

dist = df['Distance (miles)']
speed = df['Speed (mph)']

learning_rate = 0.0001
epochs = 10000

# Initial values for slope and intercept
slope = 0.1
intercept = 3
```

```
errors = []

for epoch in range(epochs):
    predicted_speed = slope * dist + intercept

    d_slope = (-2/len(dist)) * np.sum(dist * (speed - predicted_speed))
    d_intercept = (-2/len(dist)) * np.sum(speed - predicted_speed)

    error = np.sum((predicted_speed - speed) ** 2)
    errors.append(error)

    slope -= learning_rate * d_slope
    intercept -= learning_rate * d_intercept

test_dists = np.array([3, 11, 20])
predicted_speeds = slope * test_dists + intercept

plt.plot(range(epochs), errors, label='Least Squares Error')
plt.xlabel('Epochs')
plt.ylabel('Error')
plt.title('Least Squares Error over Epochs')
plt.legend()
plt.show()

plt.scatter(dist, speed, label='Actual data')
plt.plot(dist, slope * dist + intercept, color='red', label='Regression line')
plt.scatter(test_dists, predicted_speeds, color='green', label='Predicted data')
plt.xlabel('Distance (miles)')
plt.ylabel('Speed (mph)')
plt.legend()
plt.show()
```

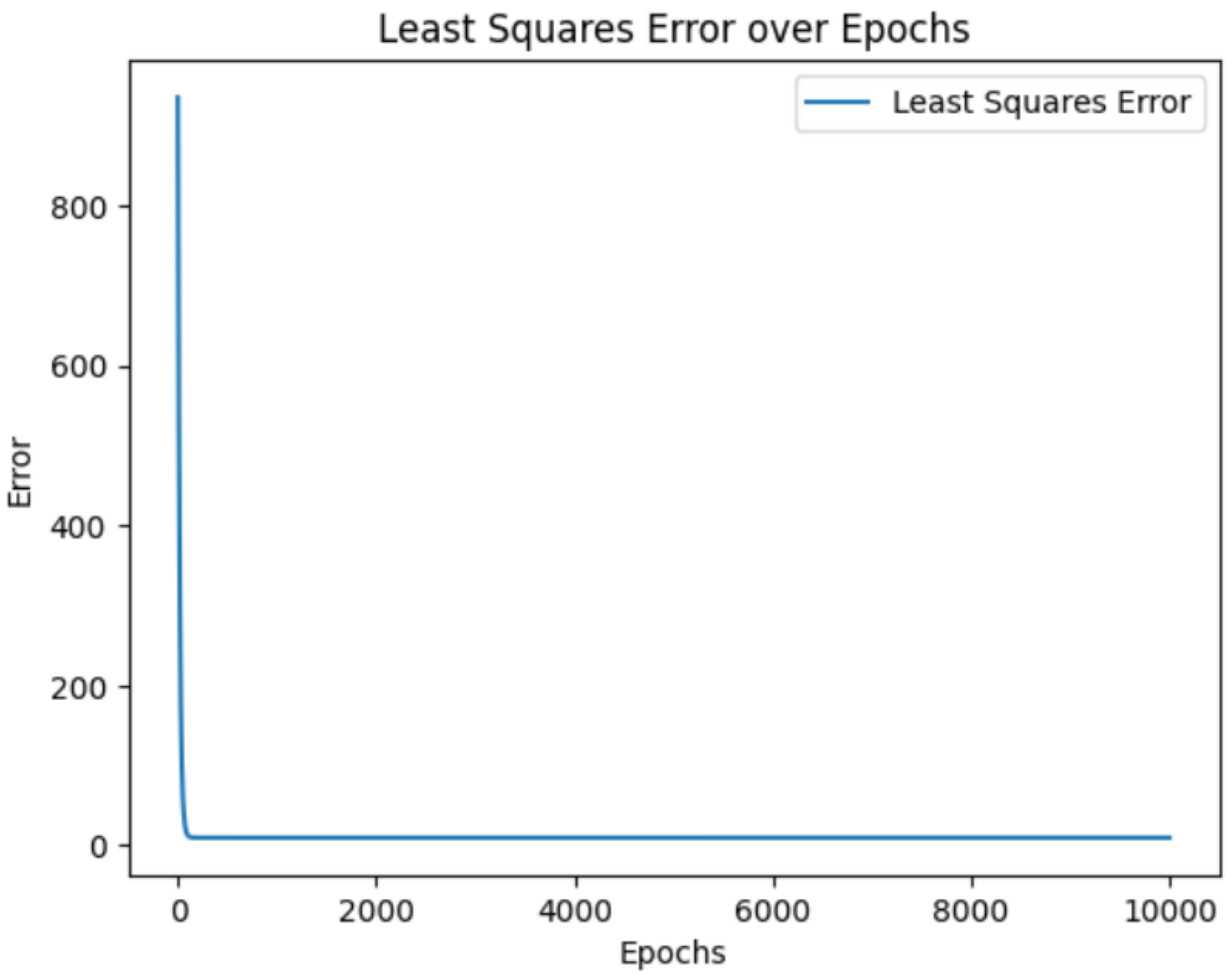
Output and Interpretation:



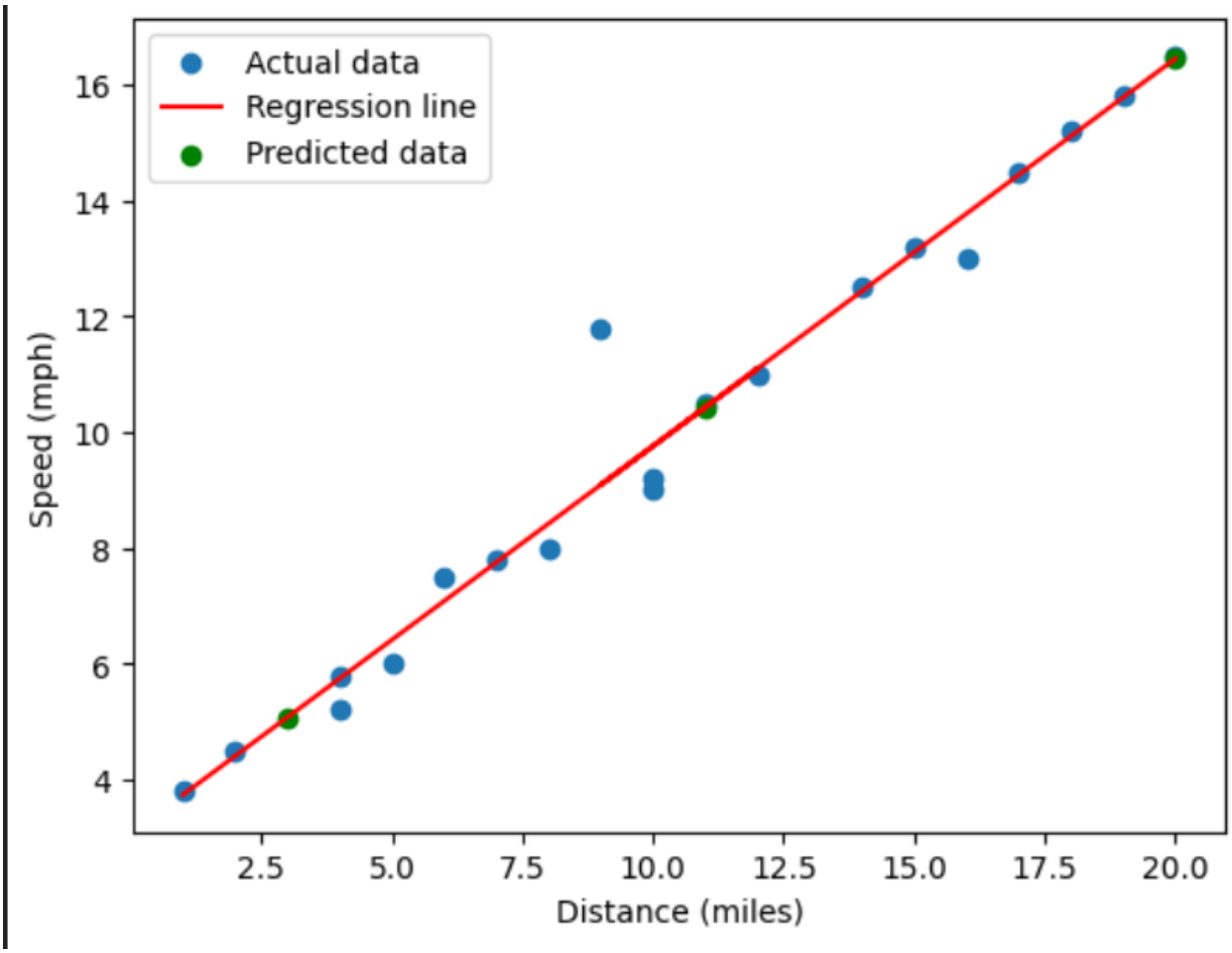
A basic scatterplot of the dataset which shows a linear relationship between distance and speed



A regression line of the scatterplot



After building the model plotting the loss over the period. And we can see the loss is reducing over time



Finally predicting the values on the model and finding the least square error and plotting the regression line on the predicted values.

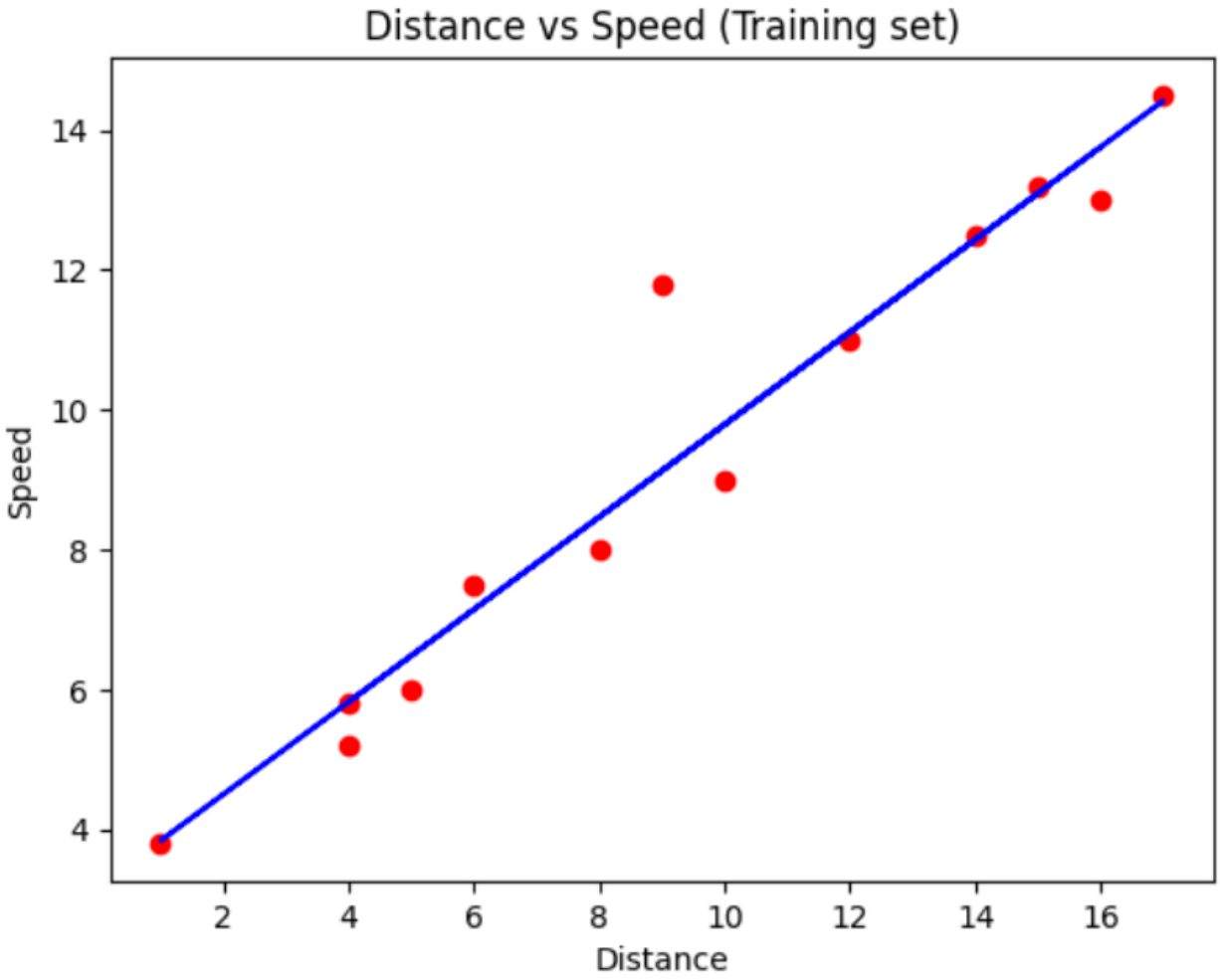
Sci-Kit Learn Implementation

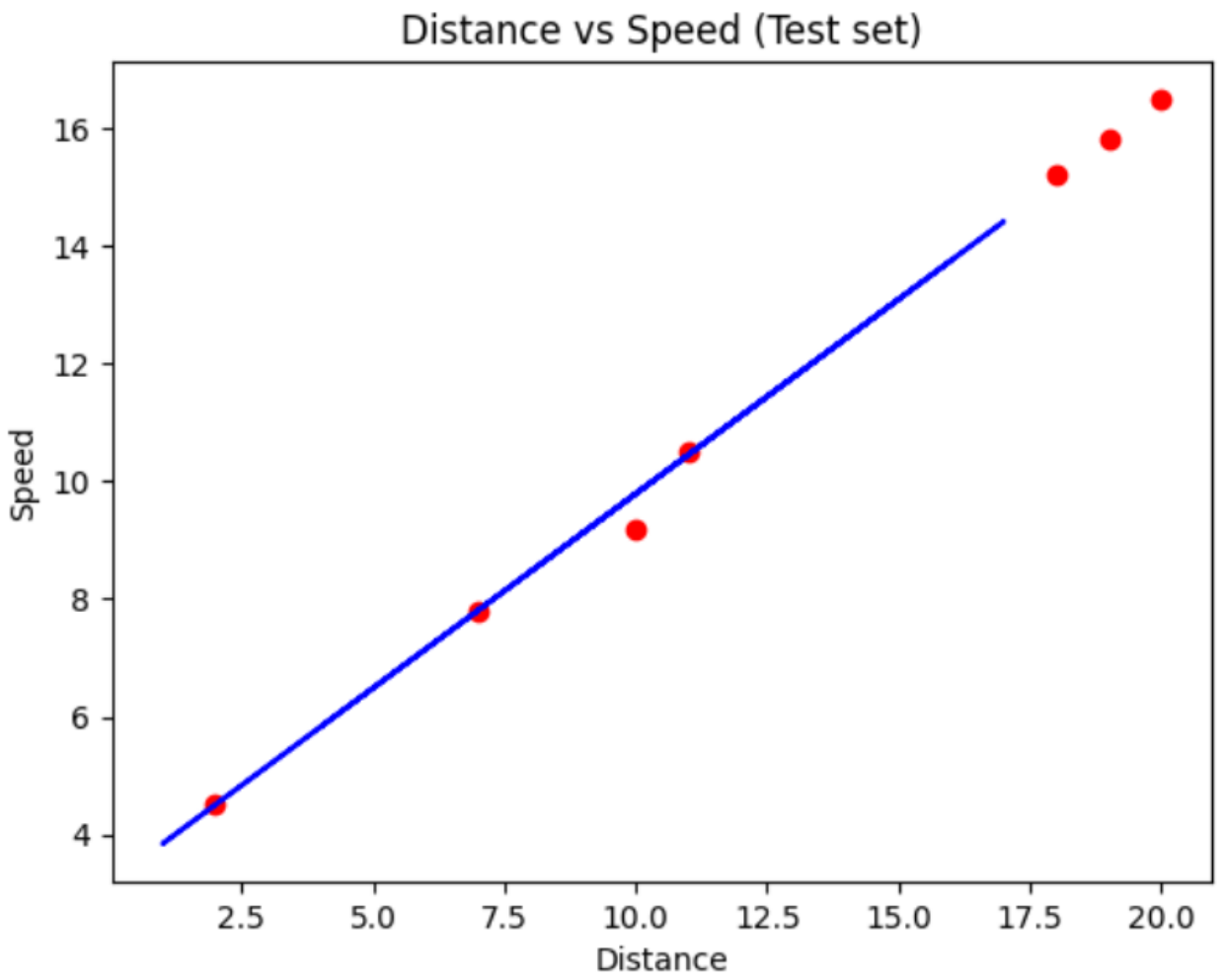
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


dataset = pd.read_csv('dataset.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values


from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)


from sklearn.linear_model import LinearRegression
```

```
regressor = LinearRegression()
regressor.fit(X_train, y_train)


y_pred = regressor.predict(X_test)


plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Distance vs Speed (Training set)')
plt.xlabel('Distance')
plt.ylabel('Speed')
plt.show()


plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Distance vs Speed (Test set)')
plt.xlabel('Distance')
plt.ylabel('Speed')
plt.show()
```

Output:


Distance vs Speed (Training set)
Speed
Distance
14
12
10
8
6
4
2
4
6
8
10
12
14
16



Conclusion:

Test your skill:

1. What are the four assumptions of linear regression (simple linear and multiple)?
   Linearity: The relationship between variables is assumed to be linear. Independence of Errors: Residuals (prediction errors) should be independent. Homoscedasticity: The variance of residuals should be constant across all levels of predictors. Normality of Errors (for inference): Residuals should be normally distributed, especially for smaller sample sizes.

2. What is meant by dependent and independent variables? (y is dependent, x are independents)
   The dependent variable (Y) is what you're trying to predict or explain, while the independent variable(s) (X) are the factors influencing or predicting the dependent variable.

3. What is difference between regression model, and estimated regression equation?
   The regression model is a theoretical framework describing the relationship between variables, while the estimated regression equation is a specific formula derived from data to represent that relationship in a given dataset.

4. What is a residual? How is it computed?
   A residual is the difference between the observed and predicted values in a regression analysis.
5. Compare between simple linear and least square regression.

   Simple linear regression focuses on predicting a dependent variable using one independent variable, while least squares regression is a method used to find the best-fitting line by minimizing the sum of squared differences between observed and predicted values, applicable to both simple and multiple linear regression.