# Implement a Simple Linear Regression

In [ ]:

*# Importing necessary libraries*

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix

import seaborn as sns

import pandas as pd

*# Creating a small dataset*

X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)

Y = np.array([1, 2, 1.3, 3.75, 2.25])

*# Initializing and fitting the linear regression model*

model = LinearRegression()

model.fit(X, Y)

*# Making predictions*

Y\_pred = model.predict(X)

*# Plotting the data points and the regression line*

plt.scatter(X, Y, color='blue', label='Actual data')

plt.plot(X, Y\_pred, color='red', label='Regression line')

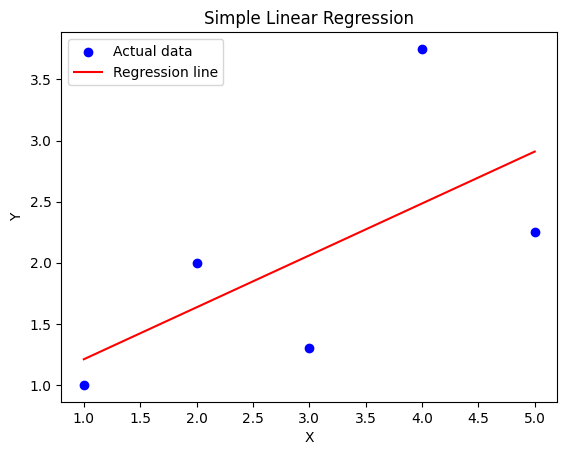
plt.title('Simple Linear Regression')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.show()



# Classify Data with a Decision Tree

In [ ]:

*# Loading the Iris dataset*

iris = load\_iris()

X, y = iris.data, iris.target

*# Splitting the dataset into training and testing sets*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

*# Initializing and training the decision tree classifier*

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

*# Making predictions*

y\_pred = clf.predict(X\_test)

*# Evaluating the model performance*

accuracy = accuracy\_score(y\_test, y\_pred)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

print(f"Accuracy: {accuracy \* 100:.2f}%")

*# Plotting the confusion matrix*

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=iris.target\_names, yticklabels=iris.target\_names)

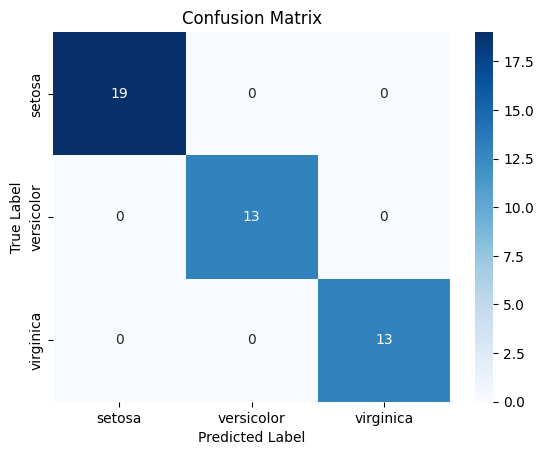
plt.title("Confusion Matrix")

plt.xlabel("Predicted Label")

plt.ylabel("True Label")

plt.show()

Accuracy: 100.00%



# Visualize Data with a Scatter Plot

In [ ]:

*# Loading the Iris dataset*

iris = sns.load\_dataset('iris')

*# Scatter plot of two features (e.g., sepal\_length vs petal\_length)*

sns.scatterplot(x='sepal\_length', y='petal\_length', hue='species', data=iris)

*# Adding labels and title*

plt.title("Sepal Length vs Petal Length in Iris Dataset")

plt.xlabel('Sepal Length (cm)')

plt.ylabel('Petal Length (cm)')

plt.show()

