CO-TECH INTERNSHIP – TASK 2

PREDICTIVE MODELING WITH LINEAR REGRESSION

[Implement a simple linear regression model using a dataset with continuous target variables. Split the data into training and testing sets, train the model on the training data, evaluate its performance using metrics like mean squared error or R-squared, and make predictions on the test set. Visualize the regression line and actual vs. predicted values to assess the model's accuracy]

1. Project Overview

- This project involves analyzing the *Iris dataset*, a well-known dataset in machine learning and statistics,
- It *aims* to apply predictive modeling using linear regression to the Iris dataset, a widely used dataset in machine learning. The dataset contains measurements of iris flowers, including features such as sepal length, sepal width, petal length, and petal width across three different species of irises: Setosa, Versicolor, and Virginica.
- The *goal* of this project is to predict one of the features of the Iris dataset, such as petal length, using linear regression. In this context, linear regression will help establish a relationship between the target feature (e.g., petal length) and the other available features (e.g., sepal length, sepal width, petal width).

The project focuses on

1. Data Exploration: Understanding the structure of the dataset through descriptive statistics and visualizations.
2. Feature Analysis: Examining the relationships and variations in the features to identify important predictors for classifying flower species.
3. Modeling: Using machine learning algorithms to develop a model that can accurately classify the species based on the given features.
4. Evaluation: Assessing model performance using metrics like accuracy and confusion matrices to validate the model's effectiveness.

2. Dataset Description

- The **Iris Dataset** contains 150 observations of iris flowers, with 50 samples from each of the three species:
- Iris setosa, Iris versicolor, and Iris virginica. The dataset includes the following attributes: • *Sepal Length (cm)*: Continuous variable representing the length of the sepal. • *Sepal Width* (cm): Continuous variable representing the width of the sepal. • *Petal Length* (cm): Continuous variable representing the length of the petal. • *Petal Width* (cm): Continuous

variable representing the width of the petal. • *Species*: Categorical variable representing the species of the iris flower, with values: 'Iris setosa', 'Iris versicolor', and 'Iris virginica'.

Step 1: Set Up the Notebook Environment

Import all necessary libraries. Comment each import line for clarity.

Libraries Used

1. pandas: For data manipulation and analysis (loading, cleaning, transforming data).
2. numpy: For numerical operations and handling arrays.
3. matplotlib: For creating static, animated, and interactive visualizations.

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

Step 2: Load and Explore the Dataset

Data Loading: Loaded the Iris dataset into a pandas DataFrame from a CSV file.

In [9]:
```
# Load the dataset
data = pd.read_csv('iris_dataset.csv')
```

To Display the first few rows of the dataset

The `data.head()` method allows you to quickly see the structure of the dataset. For this task, we're assuming `petal length` as the feature and `petal width` as the target variable.

In [10]:
```
data.head()
```

Out[10]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Step 3: Select Features and Target Variables

- Identify and isolate the feature (independent variable) and the target (dependent variable) for the linear regression model.

In [11]:
```
# Select the feature (independent variable) and target (dependent variable)
X = data[['petal length (cm)']]  # Independent variable
y = data['petal width (cm)']     # Dependent variable
```

Step 4: Split the Data into Training and Testing Sets

- Split the data to separate training and testing subsets.
- Training data is used to fit the model, and testing data evaluates its performance.

In [18]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

Step 5: Train the Linear Regression Model

- Create a Linear Regression model instance, then fit it to the training data.
- The `.fit()` function trains the model using `X_train` (feature data) and `y_train` (target data).

MODELING WITH LINEAR REGRESSION

In [17]:
```
# Create and train the Linear Regression model

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

Out[17]:

LinearRegression

LinearRegression()

Step 6: Make Predictions and Evaluate the Model

- use the LinearRegression model to make predictions on the test data , evaluate its performance using Mean Squared Error (MSE) and R-squared (R^2).

In [14]:
```
# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate Mean Squared Error (MSE) and R-squared (R²)
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}")
```

```
Mean Squared Error: 0.045604284097661846
R-squared: 0.9282562958836972
```

MSE measures the average squared difference between predicted and actual values.

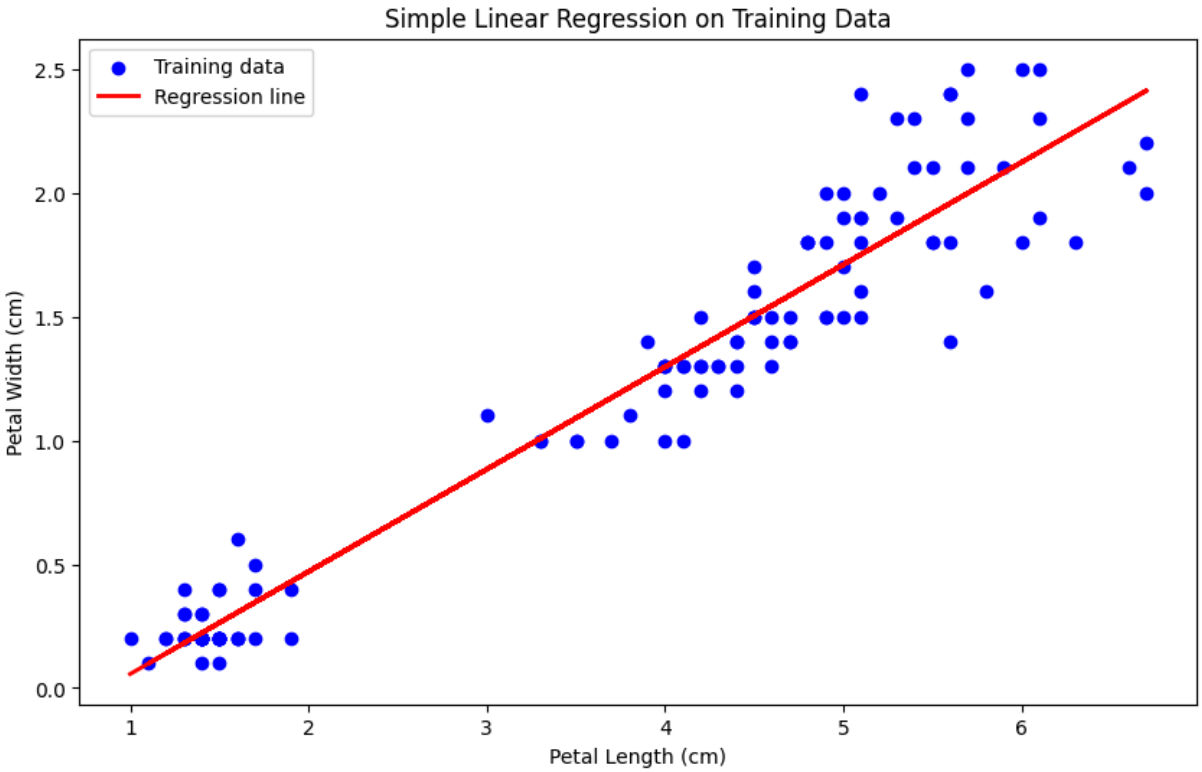
R^2 indicates the proportion of variance in the target variable explained by the model; values closer to 1 imply a better fit.

Step 7: Visualize the Result

By plotting the graph for **[SIMPLE LINEAR REGRESSION]**

- This plot helps visualize how well the model has fit the training data.

In [15]:
```
plt.figure(figsize=(10, 6))
plt.scatter(X_train, y_train, color='blue', label='Training data')
plt.plot(X_train, model.predict(X_train), color='red', linewidth=2, label='Regressi
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')
plt.title('Simple Linear Regression on Training Data')
plt.legend()
plt.show()
```

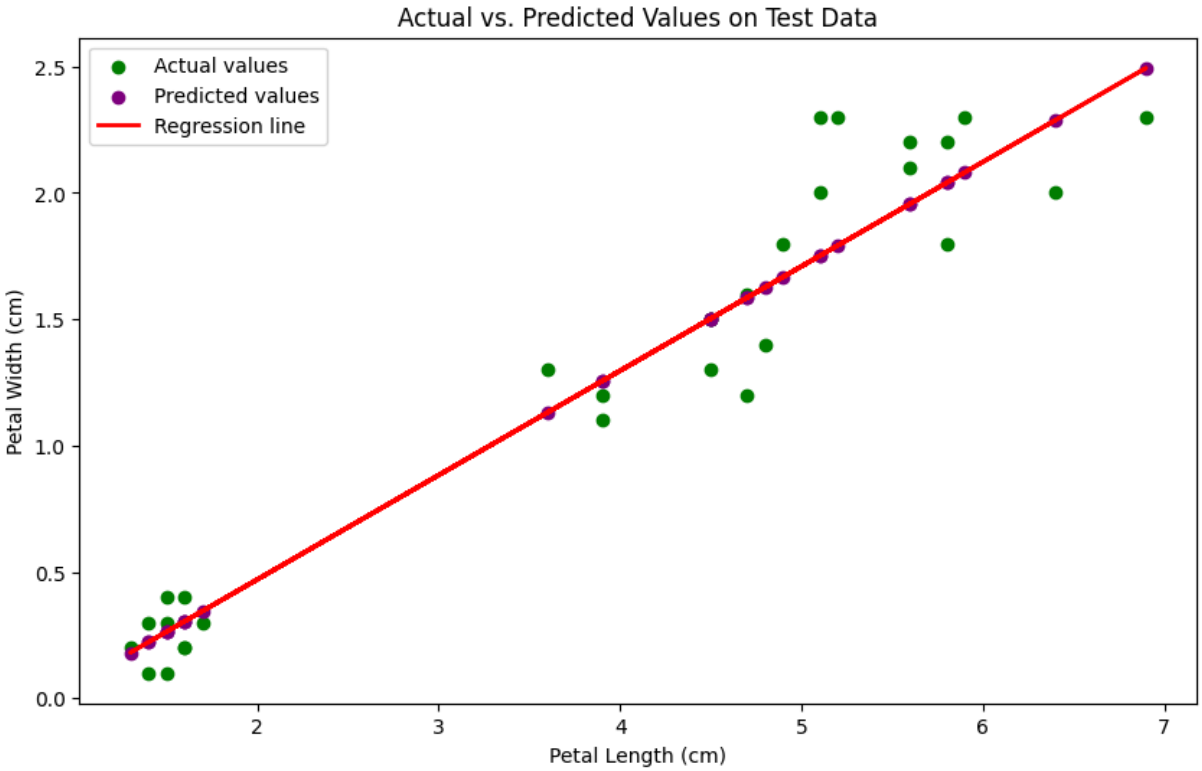


Plotting the graph for **[ACTUAL vs PREDICTED]**

- This plot allows you to assess how close the predictions are to the actual values on the test data.

In [16]:
```
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='green', label='Actual values')
plt.scatter(X_test, y_pred, color='purple', label='Predicted values')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Regression line')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')
```

```
plt.title('Actual vs. Predicted Values on Test Data')
plt.legend()
plt.show()
```



1. **Training Plot**: The red regression line indicates the model's fit on the training set. Blue dots represent the actual training data points.
2. **Test Plot**: Green dots represent actual values, while purple dots represent predicted values, showing how well predictions match actual results.

- The **regression line** and **actual vs. predicted plot** illustrate that the model is reasonably accurate for this linear relationship.

INSIGHTS

The descriptive statistics of the Iris dataset show that sepal length has a mean of 5.8 cm with moderate variability, while petal length has a mean of 3.7 cm and greater variation. Petal width has a mean of 1.3 cm, with a range from 0.1 cm to 2.5 cm, suggesting it plays a key role in distinguishing flower species. Sepal width shows a wider range, with some potential outliers. All features have no missing values, indicating a clean dataset for analysis.

INTERPRETATIONS:

- **Model Accuracy**: R-squared value shows how well the model explains variance in the target.

- **Error Analysis**: Mean Squared Error gives an idea of the average prediction error magnitude.
- **Visual Assessment**: Use the Actual vs. Predicted plot to assess if predictions closely follow actual values.

CONCLUSION

The descriptive statistics of the Iris dataset show that sepal length has a mean of 5.8 cm with moderate variability, while petal length has a mean of 3.7 cm and greater variation. Petal width has a mean of 1.3 cm, with a range from 0.1 cm to 2.5 cm, suggesting it plays a key role in distinguishing flower species. Sepal width shows a wider range, with some potential outliers. All features have no missing values, indicating a clean dataset for analysis.

THIS IS MY PROJECT CREATED BY - SAYEEDA MISBAH ILYAS

Github account for more infomation : https://github.com/codemisba

Linkdin account : https://www.linkedin.com/in/sayeeda-misbah-ilyas/