**Name:- Medhansh K**

**PRN:-122A8035**

**Batch:-B2**

**Experiment No. 5**

**AIM:** Compare the Linear regression, Logistic regression and SVM and do the analysis.

**THEROY**:

The code compares three models—**Logistic Regression**, **SVM (Support Vector Machine)**, and **Linear Regression**—on the **Iris dataset**, a classification task. The goal is to predict the species of an iris flower based on four features.

* **Data Preprocessing**:
* The **Iris dataset** is split into training (70%) and testing (30%) sets.
* **Feature scaling** is applied using **StandardScaler** to standardize the data, which is important for models like **SVM**.
* **Model Training**:
* **Logistic Regression** and **SVM** are trained as classification models.
* **Linear Regression** is used for comparison, even though it’s typically for regression, by rounding the continuous predictions to the nearest class.
* **Model Evaluation**:
* **Accuracy**, **precision**, **recall**, and **F1-score** are calculated for each model.
* A **bar chart** visualizes the accuracy of all models for easy comparison.

**CODE:**

import numpy as np import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LogisticRegression, LinearRegression from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, classification\_report from sklearn.datasets import load\_iris

# Load the iris dataset data = load\_iris()

x = data.data y = data.target

# Split data 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)

# Standardize the data scaler = StandardScaler()

x\_train\_scaled = scaler.fit\_transform(x\_train) x\_test\_scaled = scaler.transform(x\_test)

# Logistic Regression

log\_reg = LogisticRegression(max\_iter=200) log\_reg.fit(x\_train\_scaled, y\_train)

# Support Vector Machine (SVM) svm = SVC(kernel='linear') svm.fit(x\_train\_scaled, y\_train)

# Linear Regression (for comparison, although it's not ideal for classification tasks) linear\_reg = LinearRegression()

linear\_reg.fit(x\_train\_scaled, y\_train)

# Predictions

y\_pred\_log\_reg = log\_reg.predict(x\_test\_scaled) y\_pred\_svm = svm.predict(x\_test\_scaled)

y\_pred\_linear\_reg = np.round(linear\_reg.predict(x\_test\_scaled))

# Accuracy scores

accuracy\_log\_reg = accuracy\_score(y\_test, y\_pred\_log\_reg) accuracy\_svm = accuracy\_score(y\_test, y\_pred\_svm) accuracy\_linear\_reg = accuracy\_score(y\_test, y\_pred\_linear\_reg)

# Print accuracy results

print(f"Logistic Regression Accuracy: {accuracy\_log\_reg:.4f}") print(f"SVM Accuracy: {accuracy\_svm:.4f}")

print(f"Linear Regression Accuracy: {accuracy\_linear\_reg:.4f}")

# Print classification reports

print("\nLogistic Regression Classification Report:") print(classification\_report(y\_test, y\_pred\_log\_reg))

print("\nSVM Classification Report:") print(classification\_report(y\_test, y\_pred\_svm))

print("\nLinear Regression Classification Report:") print(classification\_report(y\_test, y\_pred\_linear\_reg))

# Model comparison plot

models = ['Logistic Regression', 'SVM', 'Linear Regression']

accuracies = [accuracy\_log\_reg, accuracy\_svm, accuracy\_linear\_reg]

plt.figure(figsize=(8, 6))

plt.bar(models, accuracies, color=['blue', 'green', 'orange'])

plt.xlabel('Models') plt.ylabel('Accuracy')

plt.title('Model Comparison: Accuracy')

plt.ylim([0, 1]) # To show the full range of accuracy plt.show()

**OUTPUT:**







**CONCLUSION:** The comparison shows that **Logistic Regression** and **SVM** perform well on the classification task, yielding high accuracy and good precision, recall, and F1 scores. In contrast, **Linear Regression**, which is not designed for classification, performs worse since it treats the problem as regression. The bar chart highlights that Logistic Regression and SVM are more suitable for classification tasks, while Linear Regression is less effective in this context.