Experiment Number : 2

Problem Statement: Write Python code to apply Linear Regression and Logistic Regression for different datasets. Study different types of Regression techniques.

NAME: Harshwardhan PatilROLLNO: 50

CLASS: IT-B BATCH: B1

CODE:

1. Linear Regression -

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

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

from sklearn.metrics import mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

data = pd.read\_csv('Salary\_Data.csv')

print(data.head())

X = data[['YearsExperience']]

y = data['Salary']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

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

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

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'Mean Squared Error (MSE): {mse}')

print(f'R-squared: {r2}')

plt.figure(figsize=(10, 6))

plt.scatter(y\_test.index, y\_test, label='Actual Salary', color='blue')

plt.scatter(y\_test.index, y\_pred, label='Predicted Salary', color='red', marker='x')

plt.title('Actual vs Predicted Salary')

plt.xlabel('Index')

plt.ylabel('Salary')

plt.legend()

plt.show()

  
  
A screenshot of a computer

Description automatically generated

1. Logistic Regression -

import pandas as pd

import numpy as np

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

from sklearn.linear\_model import LogisticRegression

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

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.datasets import load\_iris

iris = load\_iris()

iris\_df = pd.DataFrame(data=iris.data, columns=iris.feature\_names)

iris\_df['Species'] = iris.target

iris\_df['Species'] = iris\_df['Species'].map({0: 'Setosa', 1: 'Versicolor', 2: 'Virginica'})

features = iris.feature\_names

X = iris.data

y = iris\_df['Species']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LogisticRegression(max\_iter=1000, multi\_class='ovr')

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

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

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

conf\_matrix = confusion\_matrix(y\_test, y\_pred, labels=iris\_df['Species'].unique())

class\_report = classification\_report(y\_test, y\_pred)

print("Accuracy:", accuracy)

print("Confusion Matrix:\n", conf\_matrix)

print("Classification Report:\n", class\_report)

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

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=iris\_df['Species'].unique(), yticklabels=iris\_df['Species'].unique())

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()

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

sns.kdeplot(iris\_df[iris\_df['Species'] == 'Setosa']['sepal length (cm)'], color='green', label='Setosa', fill=True)

sns.kdeplot(iris\_df[iris\_df['Species'] == 'Versicolor']['sepal length (cm)'], color='blue', label='Versicolor', fill=True)

sns.kdeplot(iris\_df[iris\_df['Species'] == 'Virginica']['sepal length (cm)'], color='red', label='Virginica', fill=True)

plt.title('Density Distribution of Sepal Length for Each Species')

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

plt.ylabel('Density')

plt.legend()

plt.show()

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

sns.kdeplot(iris\_df[iris\_df['Species'] == 'Setosa']['petal length (cm)'], color='green', label='Setosa', fill=True)

sns.kdeplot(iris\_df[iris\_df['Species'] == 'Versicolor']['petal length (cm)'], color='blue', label='Versicolor', fill=True)

sns.kdeplot(iris\_df[iris\_df['Species'] == 'Virginica']['petal length (cm)'], color='red', label='Virginica', fill=True)

plt.title('Density Distribution of Petal Length for Each Species')

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

plt.ylabel('Density')

plt.legend()

plt.show()

plt.figure(figsize=(10, 8))

sns.heatmap(iris\_df[features].corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.title('Correlation Heatmap of Features')

plt.show()

OUTPUT:

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

A diagram of a number of species

Description automatically generated

A diagram of a number of species

Description automatically generated

A screenshot of a graph

Description automatically generated