

10/4/24

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

Ex - 9

Free Work

## logical Regression

• Aim :

To implement logistic Regression and show graphs.

• Code :

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

data = load_iris()
X, y = data.data, data.target
y_binary = (y > np.median(y)).astype(int)
X_train, X_test, y_train, y_test = train_test_split(
    X, y_binary, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
y_test = scaler.transform(X_test)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: {:.3f} %".format(accuracy * 100))
```

Output

Accuracy : 73.03%

Confusion Matrix :

[[ 36 13 ]]

[ 11 29 ]]

Classification Report :

	precision	recall	f1-score	support
0	0.77	0.73	0.75	49
1	0.69	0.72	0.71	40

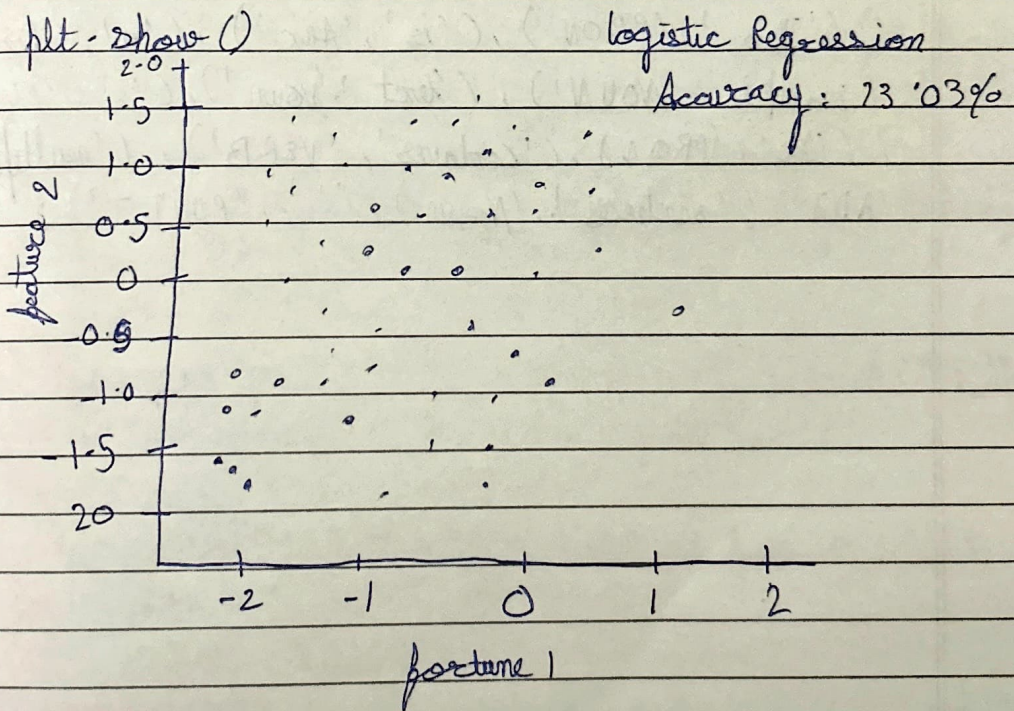
accuracy			0.73	89
macro avg	0.73	0.73	0.73	89
weighted	0.73	0.73	0.73	89



```

print ("Confusion Matrix: \n", confusion_matrix
      (y_test, y_pred))
print ("\n Classification Report: \n", Classification
      report(y_test, y_pred))
plt.figure(figsize=(8,6))
sns.scatterplot(x=x_test[:,2], y=x_test
     [:,8], hue=y_test, palette=[0:'blue', 1:'red'],
      markers='o')
plt.xlabel("feature 1")
plt.ylabel("feature 2")
plt.title("Logistic Regression \n Accuracy: 73.03%
      " format (accuracy * 100))

```



Result:

Logistic Regression was successfully demonstrated