SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
ASPproved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Redesigning Common Mind & Business Towards Excellence

3P urpose rocess eople Culture

sns INSTITUTIONS
www.snsgroups.com

1st GenAI Powered Design Thinking FrameWork

Build an Entrepreneurial Mindset Through Our Design Thinking FrameWork

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

23AMB201 - MACHINE LEARNING

II YEAR IV SEM

UNIT II – SUPERVISED LEARNING ALGORITHMS

TOPIC 9 – Implementing Linear and Logistic regression using Scikit Learn Library

Linear Regression

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('salary_data.csv')
print("Original Data:")
print(df.head())
```

```
Original Data:
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
4              2.2  39891.0
```

```
[ ]  x = df.iloc[:,:-1].values
     y = df.iloc[:,1].values
     y
```

```
array([ 39343.,  46205.,  37731.,  43525.,  39891.,  56642.,  60150.,
        54445.,  64445.,  57189.,  63218.,  55794.,  56957.,  57081.,
        61111.,  67938.,  66029.,  83088.,  81363.,  93940.,  91738.,
        98273., 101302., 113812., 109431., 105582., 116969., 112635.,
       122391., 121872.])
```

Linear Regression

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
```

```
LinearRegression
LinearRegression()
```

```
[ ]  viz_train = plt
     viz_train.scatter(x_train, y_train, color='red')
     viz_train.plot(x_train, regressor.predict(x_train), color='blue')
     viz_train.title('Salary Vs Experience (Training set)')
     viz_train.xlabel('Year of Experience')
     viz_train.ylabel('Salary')
     viz_train.show()
```

Linear Regression-Prediction of salary for 20 year of experience

```
viz_test = plt
viz_test.scatter(x_test, y_test, color='red')
viz_test.plot(x_test, regressor.predict(x_test), color='blue')
viz_test.title('Salary VS Experience (Test set)')
viz_test.xlabel('Year of Experience')
viz_test.ylabel('Salary')
viz_test.show()
```


```
[ ] y_pred=regressor.predict([[20]])
    y_pred
```

```
array([214618.91713627])
```

Logistic Regression

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
data = {
    "Age": [25, 32, 47, 50, 35, 60, 28, 55, 40, 45],
    "BloodPressure": [120, 130, 140, 135, 128, 145, 118, 150, 138, 132],
    "Glucose": [85, 90, 160, 150, 88, 170, 80, 180, 140, 110],
    "Diabetic": [0, 0, 1, 1, 0, 1, 0, 1, 1, 0]
}
df = pd.DataFrame(data)
df
```

Logistic Regression

	Age	BloodPressure	Glucose	Diabetic
0	25	120	85	0
1	32	130	90	0
2	47	140	160	1
3	50	135	150	1
4	35	128	88	0
5	60	145	170	1
6	28	118	80	0
7	55	150	180	1
8	40	138	140	1
9	45	132	110	0

```
X = df[["Age", "BloodPressure", "Glucose"]]  # Input Features
y = df["Diabetic"]  # Output Label
X
```

	Age	BloodPressure	Glucose
0	25	120	85
1	32	130	90
2	47	140	160
3	50	135	150
4	35	128	88
5	60	145	170
6	28	118	80
7	55	150	180
8	40	138	140
9	45	132	110

Logistic Regression

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
    X_test
```

	Age	BloodPressure	Glucose
3	50	135	150
2	47	140	160

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

single_patient = np.array([[25, 120, 85]])
single_prediction = model.predict(single_patient)
```

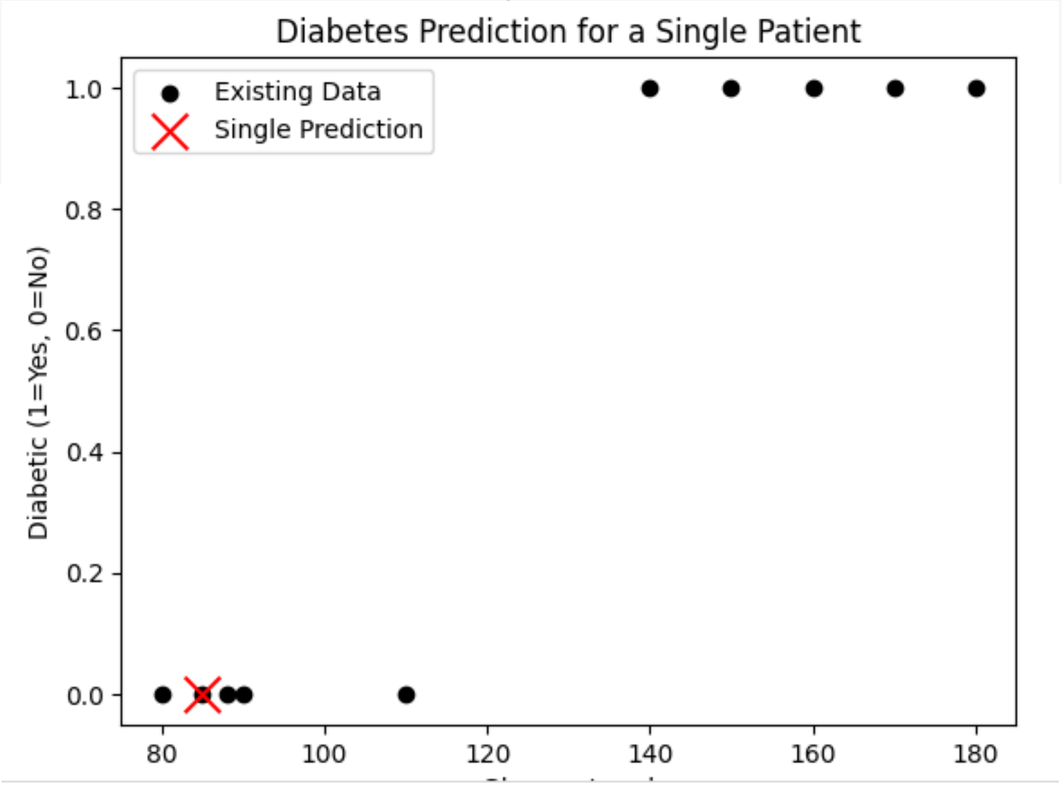
Logistic Regression- Prediction

```
result = "Diabetic" if single_prediction == 1 else "Non-Diabetic"
print(f"📌 Prediction for Single Patient:")
print(f"Age: {single_patient[0][0]}, BP: {single_patient[0][1]}, Glucose: {single_patient[0][2]}")
print(f"Prediction: {result}")
```

```
📌 Prediction for Single Patient:
Age: 25, BP: 120, Glucose: 85
Prediction: Non-Diabetic
```

Logistic Regression- Visualization

```
# Visualization: Showing the single patient prediction
plt.scatter(X["Glucose"], y, color="black", label="Existing Data")
plt.scatter(single_patient[:, 2], single_prediction, color="red", label="Single Prediction", marker="x", s=200)
plt.xlabel("Glucose Level")
plt.ylabel("Diabetic (1=Yes, 0=No)")
plt.legend()
plt.title("Diabetes Prediction for a Single Patient")
plt.show()
```



References

1. Tom M. Mitchell, “Machine Learning”, McGraw-Hill Education (India) Private Limited, 2013.
2. Sebastian Raschka , Yuxi (Hayden) Liu Machine Learning with PyTorch and Scikit-Learn: Developmachine learning and deep learning models with Python Packt Publishing Limited (23 December 2022).