

# EXPERIMENT 3

## 1. Introduction

Logistic Regression is a machine learning algorithm used for binary classification problems. This document provides a step-by-step guide to implementing Logistic Regression using Python and Scikit-Learn.

## 2. Code Implementation

```
#Logistic Regression
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

#Create a simple dataset (Study Hours & Pass/Fail)
study_hours = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]) #
Independent variable (X)
```

```
pass_fail = np.array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1]) #  
Dependent variable (Y)
```

```
#Reshape data (needed for sklearn)
```

```
study_hours = study_hours.reshape(-1, 1)
```

```
#Split data into training and testing sets
```

```
x_train, x_test, y_train, y_test =  
train_test_split(study_hours, pass_fail, test_size=0.2,  
random_state=42)
```

```
#Train logistic regression model
```

```
model = LogisticRegression()
```

```
model.fit(x_train, y_train)
```

```
#Make predictions
```

```
y_pred = model.predict(x_test)
```

```
#Check accuracy
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print(f"Model Accuracy: {accuracy:.2f}")
```

```
#Plot results

plt.scatter(study_hours, pass_fail, color='red',
label="Actual Data")

plt.plot(study_hours,
model.predict_proba(study_hours)[:,1], color='blue',
label="Predicted Probability")

plt.xlabel("Study Hours")

plt.ylabel("Pass Probability")

plt.title("Logistic Regression - Pass/Fail Prediction")

plt.legend()

plt.show()
```

### 3. Model Evaluation Output:

