機器學習

LABORATORY: REGRESSION

NAME: 黃仕翔	**STUDENT ID#:** 313513054

Objectives:

- Develop a deeper understanding of classification models, focusing on logistic regression and probit regression.
- Learn the mathematical foundations and differences between sigmoid (logistic) and normal CDF (probit) activation functions.
- Implement logistic and probit regression from scratch using NumPy without relying on sklearn's built-in classifiers.
- Train models using gradient descent, analyze their convergence, and evaluate their performance.
- Apply key classification metrics, including confusion matrices, ROC curves, and AUC, to compare model effectiveness.

Part 1. Background

Classification is a fundamental machine learning task that involves predicting categorical labels.

- Logistic Regression uses the sigmoid function to model probabilities and is widely used for binary classification.
- Probit Regression models probabilities using the cumulative normal distribution (CDF), which assumes a normally distributed error term.

Tasks: In this assignment, you will implement both logistic and probit regression using only NumPy. You will get no points by simply calling *sklearn.linear_model.LogisticRegression.* Your task is to train these models on a provided dataset, evaluate their performance, and test to classify them. Compare confusion matrices, ROC curves, and AUC for both models to understand their strengths and limitations. The dataset and sample code can be found here: https://github.com/Satriosnjya/ML-Labs.git

Classification Tasks: The goal of the classification model (Logistic/Probit Regression) is to predict y, which is a binary label (0 or 1). **What are we classifying?** We are predicting whether an individual earns more than 50K per year based on demographic and work-related factors.

- Input Features:
  - Categorical variables (age_bin, occupation_bin, education_bin, etc.), which need to be converted into numerical features using one-hot encoding before training.
- Output (Target Variable y):
  - A binary classification label (0 or 1).

Part 2. Arithmetic Instructions.

Step **Procedure**

1 **Logistic Regression:**

1. (__init__) We initialize the weights and bias:

$$w = 0,\ b = 0$$

2. (fit)

- Linear Model Equation:

$$z = Xw + b$$

- Sigmoid Function (Logistic Activation Function):

$$P(Y = 1 \mid X) = \sigma(Xw + b) = \frac{1}{1 + e^{-(Xw+b)}}$$

- Gradient Descent Updates:

$$w = w - \alpha \cdot \frac{1}{n}\sum_{i=1}^{n} X_i \cdot (\sigma(z_i) - Y_i)$$

$$b = b - \alpha \cdot \frac{1}{n}\sum_{i=1}^{n} (\sigma(z_i) - Y_i)$$

3. (predict_proba) Uses the sigmoid function to compute probability:

$$P(Y = 1 \mid X) = \sigma(Xw + b) = \frac{1}{1 + e^{-(Xw+b)}}$$

This outputs a probability value between 0 and 1.

4. (predict)

Convert probabilities to class labels using a threshold $\tau = 0.5$:

$$\hat{Y} = \begin{cases} 1, & \text{if } \sigma(Xw + b) > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

2 **Probit Regression:**

1. (__init__) We initialize the weights and bias:

$$w = 0,\ b = 0$$

2. (fit)

- Linear Model Equation:

$$z = Xw + b$$

- Probit Function (Cumulative Distribution Function of Standard Normal Distribution):

$$P(Y = 1 \mid X) = \Phi(Xw + b)$$

Where $\Phi(z)$ is the CDF of the standard normal distribution:

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

- Gradient Descent Updates:

$$w = w - \alpha \cdot \frac{1}{n}\sum_{i=1}^{n} X_i \cdot (\Phi(z_i) - Y_i)$$

$$b = b - \alpha \cdot \frac{1}{n}\sum_{i=1}^{n} (\Phi(z_i) - Y_i)$$

3. (predict_proba) Uses the probit function to compute probability:

$$P(Y = 1 \mid X) = \Phi(Xw + b)$$

This outputs a probability value between 0 and 1. Approximate normal CDF using the tanh function:

$$\Phi(z) \approx 0.5\left(1 + \tanh\left(\frac{z}{\sqrt{2}}\right)\right)$$

4. Convert probabilities to class labels using a threshold $\tau = 0.5$:

$$\hat{Y} = \begin{cases} 1, & \text{if } \Phi(Xw + b) > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

3 Confusion Matrix

A confusion matrix is a table used to evaluate the performance of a classification model. It consists of the following four elements:

$$CM = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

where:

- True Positive (TP): The model correctly predicts class 1.
- True Negative (TN): The model correctly predicts class 0 .
- False Positive (FP): The model predicts 1 when the actual class is 0 .
- False Negative (FN): The model predicts 0 when the actual class is 1 .

Thus, the confusion matrix is constructed as:

$$CM = \begin{bmatrix} \sum (y_{\text{test}} = 0 \text{ and } \hat{y} = 0) & \sum (y_{\text{test}} = 0 \text{ and } \hat{y} = 1) \\ \sum (y_{\text{test}} = 1 \text{ and } \hat{y} = 0) & \sum (y_{\text{test}} = 1 \text{ and } \hat{y} = 1) \end{bmatrix}$$

4 ROC Curve

The ROC curve is a plot that shows the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.

True Positive Rate (TPR), Also known as recall or sensitivity:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (FPR), The proportion of incorrectly predicted positives out of all actual negatives:

$$FPR = \frac{FP}{FP + TN}$$

Equation Derivation, Given different classification thresholds t, we calculate:

$$y_{\text{predicted}}^{(t)} = \begin{cases} 1, & \text{if } P(Y = 1 \mid X) > t \\ 0, & \text{otherwise} \end{cases}$$

For each threshold t, compute:

$$\text{TPR}\ (t) = \frac{\sum \left(y_{\text{test}} = 1 \text{ and } y_{\text{predicted}}^{(t)} = 1 \right)}{\sum\ (y_{\text{test}} = 1)}$$

$$\text{FPR}\ (t) = \frac{\sum \left(y_{\text{test}} = 0 \text{ and } y_{\text{predicted}}^{(t)} = 1 \right)}{\sum\ (y_{\text{test}} = 0)}$$

The ROC curve is then obtained by plotting TPR (t) against $FPR(t)$ for all possible thresholds.

5 AUC

The AUC (Area Under the Curve) quantifies the overall performance of the classifier. It is computed as:

$$AUC = \int_0^1 TPR(FPR)d(FPR)$$

Using numerical integration (e.g., the trapezoidal rule), we approximate:

$$AUC = \left| \sum_{i=1}^{n} \frac{(TPR_i + TPR_{i-1})}{2} (FPR_i - FPR_{i-1}) \right|$$

Or, in discrete form:

$$AUC = \left| \int_0^1 TPR(FPR)dFPR \right| = \left| \sum_i TPR_i \cdot \Delta FPR_i \right|$$

where:

$$\Delta FPR_i = FPR_i - FPR_{i-1}$$

Thus, the AUC measures the classifier's ability to distinguish between classes.

Part 3. Data Transfer Instructions.

Step **Procedure**

1
- Download dataset and example code from https://github.com/Satriosnjya/ML-Labs.git
- Open colab.research.google.com
- Make a New Notebook
- Upload classification.csv in Colab Notebook

2 **Load Libraries**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2 **Generate Data**

```
# Load Dataset
file_path = "00 df.csv"
df = pd.read_csv(file_path)

# Data Preparation
cat_feats = ['age_bin','capital_gl_bin','education_bin','hours_per_week_bin','msr_bin','occupation_bin','race_sex_bin']
```

3 **Split Dataset**

```
# Split dataset into train and test
y_train = df[df['flag']=='train']['y']
x_train = df[df['flag']=='train'][cat_feats]
x_train = pd.get_dummies(x_train, columns=cat_feats, drop_first=True)

y_test = df[df['flag']=='test']['y']
x_test = df[df['flag']=='test'][cat_feats]
x_test = pd.get_dummies(x_test, columns=cat_feats, drop_first=True)
```

4 **Convert to np arrays and apply manual feature scaling**

```
# Convert to numpy arrays
x_train, y_train = x_train.values, y_train.values
x_test, y_test = x_test.values, y_test.values

# Apply manual feature scaling
mean_train = np.mean(x_train, axis=0)
std_train = np.std(x_train, axis=0)
x_train_scaled = (x_train - mean_train) / std_train
x_test_scaled = (x_test - mean_train) / std_train
```

Grading & Submission Instructions

Assignment (70%):

1. (20%) Implement Logistic Regression
   a. Train the model using learning rate = 0.01, and iteration =3000
2. (20%) Implement Probit Regression
   a. Implement and train the probit function to estimate probabilities
3. (10%) Compute Confusion Matrices
   a. Construct a confusion matrix for both logistic and probit regression
   b. Compute TP, TN, FP, FN
4. (10%) Generate and Analyze the ROC Curve
   a. Compute TPR and FPR for different threshold
   b. Plot the ROC curve
5. (10%) Compute and Compare AUC
   a. Compute the AUC for logistic and probit regression

Question (30%):

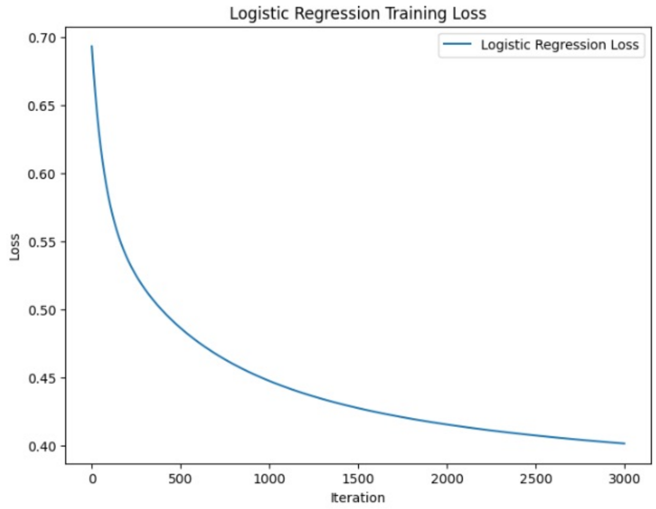
1. (5%) **Show the comparison** of the **confusion matrices** of logistic and probit regression. Do they produce similar results? Why or why not?
2. (5%) How does the ROC curve and AUV of logistic regression compare to that of probit regression? Are there any key differences? Explain and **show a side-by-side plot comparison.**
3. (5%) Discuss the impact of different learning rates and iterations on the convergence of logistic and probit regression. How does hyperparameter tuning affect performance?
   ***Provide the results of ROC curves for different hyperparameters.**
4. (5%) Explain the **fundamental differences** between logistic regression and probit regression. When might you choose one over the other?
5. (5%) Discuss their **activation functions**: **sigmoid** for logistic and normal **CDF** for probit. How do these functions influence decision boundaries?
6. (5%) Define and explain the **significance** of Confusion Matrices, ROC Curves, and AUC in evaluating classification models. How do they contribute to model selection?

Submission :

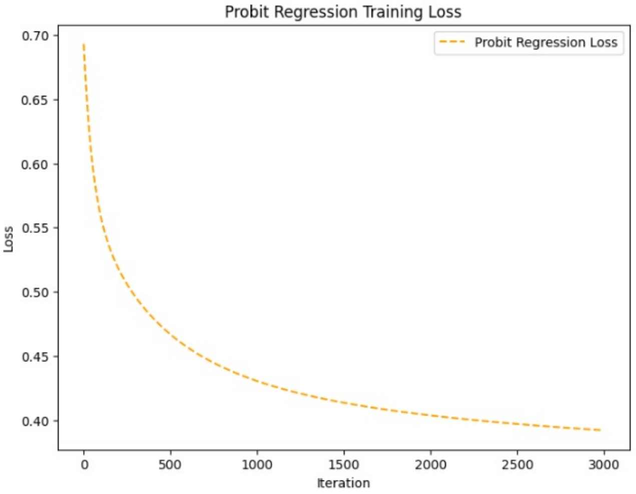
1. Report: Answer all conceptual questions. Include screenshots of your results in the last pages of this PDF File.
2. Code: Submit your complete Python script in either .py or .ipynb format.
3. Upload both your report and code to the E3 system. Name your files correctly:
   a. Report: StudentID_Lab2.pdf
   b. Code: StudentID_Lab2.py or StudentID_Lab2.ipynb
4. 1 day late: 10% deduction from total score.
5. Plagiarism is **strictly prohibited**. Submitting copied work from other students will result in penalties.

Code Results and Answer:

1. (20%) Implement Logistic Regression
   a. Train the model using learning rate = 0.01, and iteration =3000

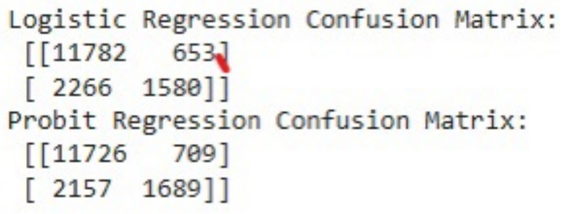


2. (20%) Implement Probit Regression
   a. Implement and train the probit function to estimate probabilities



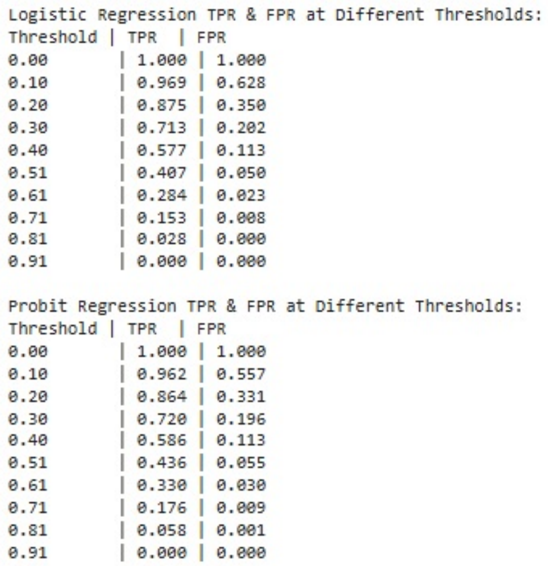
3. (10%) Compute Confusion Matrices
   a. Construct a confusion matrix for both logistic and probit regression
   b. Compute TP, TN, FP, FN

$$CM = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$



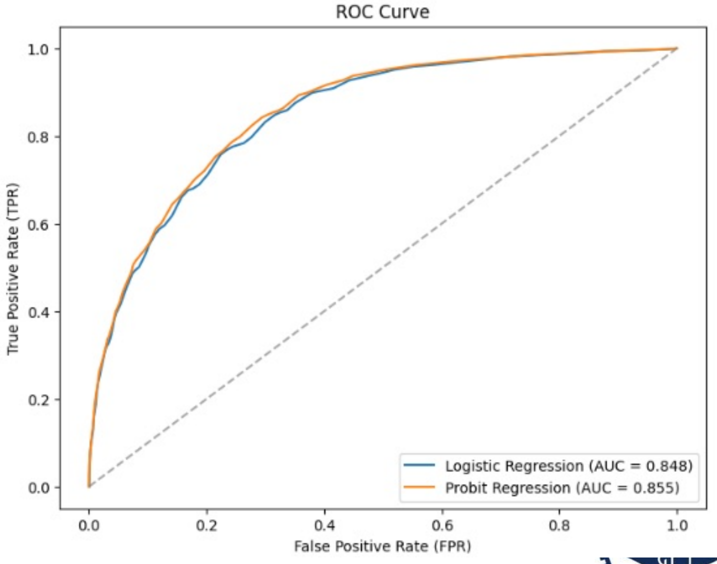
```
Logistic Regression Confusion Matrix:
 [[11782   653]
 [ 2266  1580]]
Probit Regression Confusion Matrix:
 [[11726   709]
 [ 2157  1689]]
```

4. (10%) Generate and Analyze the ROC Curve
   a. Compute TPR and FPR for different threshold
   b. Plot the ROC curve

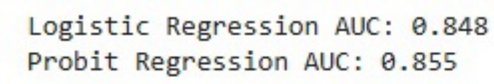


```
Logistic Regression TPR & FPR at Different Thresholds:
Threshold | TPR   | FPR
0.00      | 1.000 | 1.000
0.10      | 0.969 | 0.628
0.20      | 0.875 | 0.350
0.30      | 0.713 | 0.202
0.40      | 0.577 | 0.113
0.51      | 0.407 | 0.050
0.61      | 0.284 | 0.023
0.71      | 0.153 | 0.008
0.81      | 0.028 | 0.000
0.91      | 0.000 | 0.000

Probit Regression TPR & FPR at Different Thresholds:
Threshold | TPR   | FPR
0.00      | 1.000 | 1.000
0.10      | 0.962 | 0.557
0.20      | 0.864 | 0.331
0.30      | 0.720 | 0.196
0.40      | 0.586 | 0.113
0.51      | 0.436 | 0.055
0.61      | 0.330 | 0.030
0.71      | 0.176 | 0.009
0.81      | 0.058 | 0.001
0.91      | 0.000 | 0.000
```



5. (10%) Compute and Compare AUC
   a. Compute the AUC for logistic and probit regression



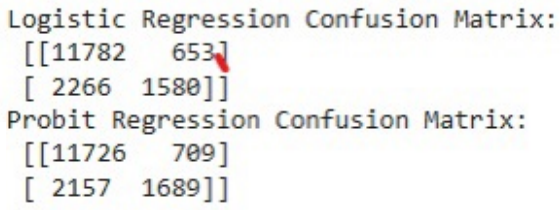
```
Logistic Regression AUC: 0.848
Probit Regression AUC: 0.855
```

Code Results and Answer:

Question

Ans 1.

$$CM = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

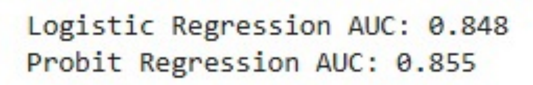


```
Logistic Regression Confusion Matrix:
 [[11782   653]
 [ 2266  1580]]
Probit Regression Confusion Matrix:
 [[11726   709]
 [ 2157  1689]]
```

- Logistic and Probit are similar.

  Logistic is based on sigmoid function.
  Probit is based on standard normal distribution.

Ans 2.



```
Logistic Regression AUC: 0.848
Probit Regression AUC: 0.855
```

- The two are almost equal, but Probit is slightly better.

Ans 3.

- ROC curve: Probit regression is more smooth.
- AUV : They are very similar.

Logistic regression: $P(Y=1|X) = \frac{1}{1+e^{-X\omega}}$

→ sigmoid function change rapidly, especially near zero.

→ cause more dramatic variation

Probit regression: $P(Y=1|x) = \Phi(X\omega)$

→ CDF of standard normal distribution are more smooth.

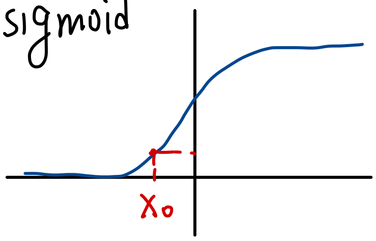
Ans 4. fundamental difference :

	Logistic regression	Probit regression
function	$P(Y=1\|X) = \frac{1}{1+e^{-X\omega}}$	$P(Y=1\|x) = \Phi(X\omega)$
complixity	more easy. NO integration.	more complex Need integral
AUC	Slightly smaller.	Slightly larger
Choose	1. Computation resource are limit 2. Data contain extreme value. we can convert it to 0~1.	1. When a smoother probability transformation is needed to prevent extreme value 2. When model's error term follow Gaussian distribution.

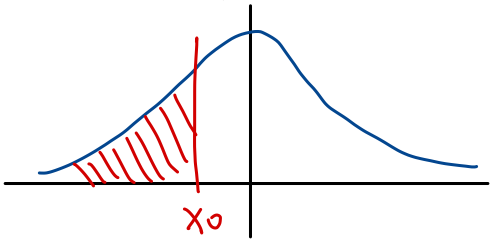
Code Results and Answer:

Ans 5.

sigmoid



CDF of Gaussian



$P(Y=1|X)=\frac{1}{1+e^{-Xw}}$

when $Xw=0$, $P(Y=1|X)=\frac{1}{2}$

maping value into $(0,1)$

$P(Y=1|X)=\Phi(Xw)$

when $Xw=0$, $P(Y=1|X)=\frac{1}{2}$

more smooth

Ans 6.

- Confusion matrices:

$$CM=\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

TP (true positive): real 1 and pred 1

TN (true negative): real 0 and pred 1

FP (false positive): real 0 but pred 1

FN (false negative): real 1 but pred 0

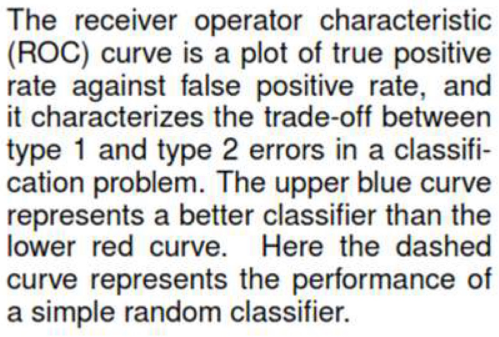
$\Rightarrow$ We can define

Accurcy: $\frac{TP+TN}{TP+TN+FP+FN}$

Precision: $\frac{TP}{TP+FP}$

Recall (TPR): $\frac{TP}{TP+FN}$

- ROC curve



The receiver operator characteristic (ROC) curve is a plot of true positive rate against false positive rate, and it characterizes the trade-off between type 1 and type 2 errors in a classification problem. The upper blue curve represents a better classifier than the lower red curve. Here the dashed curve represents the performance of a simple random classifier.

- AUC

- measure classification accurcy:
- Higher AUC: better classifier

AUC = 1 : perfect

AUC = 0.5 : worst

AUC < 0.5 : oppsite classification