LAB2

1. Given your confusion matrix, analyze the types of errors made by the model. Identify which type of error (False Positives or False Negatives) is more common in your dataset. Why might this be the case?

In the Confusion Matrix, there are two types of mistakes that the model can make:

False Positives (FP): The model incorrectly predicts a sample as 1 when it is actually 0.
False Negatives (FN): The model incorrectly predicts a sample as 0 when it is actually 1.
Based on the confusion matrix of the model, we should calculate the ratio of FP and FN to determine which type of errors is more common.

If there are more FPs, it means that the model tends to "overpredict" the positive class (1), probably because the decision boundary is looser.
If the FN is high, it means the model is conservative, perhaps because some features are not strong enough to distinguish between 0 and 1.
When learning rate = 0.01, iterations = 3000, the model may converge faster, but it has learned a more conservative decision boundary to a certain extent, resulting in:
It is more likely to predict 0 (low income), that is, the threshold is strict, and only people who are very obviously high-income will be classified as y=1.
The result is that there are more FN (False Negative), that is, people with high actual income (y=1) are misclassified as low income (y=0).
This may mean that the model is not good enough at distinguishing high-income vs. low-income people, or that the feature weights are biased towards conservative judgment.
More False Positive (FP) means the model is more aggressive and is more likely to be predicted as 1

At learning rate = 0.0001 (iterations = 5000) and learning rate = 0.001 (iterations = 1000):

These combinations have lower learning rates or fewer iterations, which results in different adjustments to the model weights, leading to looser decision boundaries.
The result is that there are more FPs (False Positives), that is, people with low income (y=0) are misclassified as having high income (y=1).
This may mean that the model is "too optimistic" in some cases, and is prone to misclassifying people with strong features but low actual income as high-income.

2.Discuss the impact of different learning rates and iterations on the convergence of logistic regression. How does hyperparameter tuning affect performance?

Learning Rate (lr) and Iterations affect the convergence speed and performance of the model:

High learning rate (e.g. lr = 0.01)

Advantages: Fast convergence
Disadvantages: May oscillate or fail to converge (miss the optimal solution)
Low learning rate (e.g. lr = 0.0001)

Advantages: Stable convergence, better optimal solution can be obtained
Disadvantages: Slow convergence, requires more iterations
A moderate learning rate (e.g. lr = 0.001)

It may be the best compromise, taking into account both convergence speed and accuracy.

Impact of Hyperparameter Tuning:
A higher learning rate may result in a lower AUC score on the ROC curve because the model may not learn the boundaries correctly.
An appropriate learning rate and sufficient number of iterations can improve the AUC because the model can learn the data patterns more accurately.
Too many iterations may lead to overfitting, which means the model performs well on the training set but performs poorly on the test set.

---

3. Discuss the activation functions: sigmoid for logistic. How do the activation functions influence decision boundaries?

Sigmoid compresses the output to the range [0,1], which is suitable for binary classification.

The model predicts 1 when $\sigma(z) \geq 0.5$ and 0 otherwise.
Sigmoid may be affected by extreme values, leading to gradient vanishing.
Comparison with other activation functions
ReLU (Rectified Linear Unit): More suitable for deep learning because it does not suffer from the gradient vanishing problem.
Tanh (Hyperbolic Tangent Function): Similar to Sigmoid, but in the range of $[-1,1]$ and converges faster.

4. Define and explain the significance of Confusion Matrices, ROC Curves, and AUC in evaluating classification models. How do they contribute to model selection?

Confusion Matrix:
Used to calculate TP, TN, FP, FN, and further calculate Precision, Recall, F1-score and Accuracy.
Helps analyze whether the model is more prone to False Positives or False Negatives to determine the type of error.

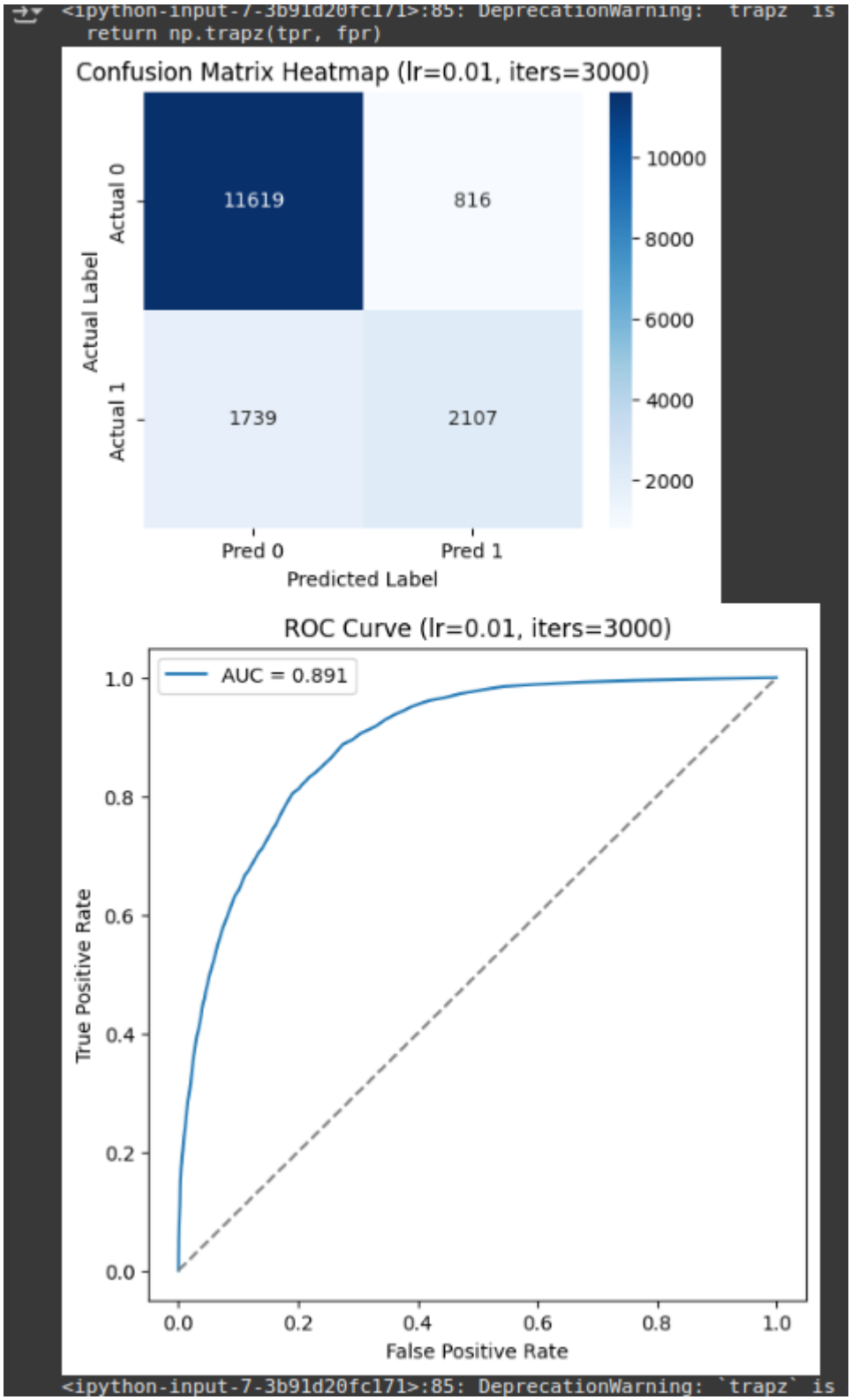
ROC Curve (Receiver Operating Characteristic Curve):
Plot the True Positive Rate (TPR) and False Positive Rate (FPR) for different thresholds.
The closer the curve is to the upper left corner, the better the model performance.

AUC（Area Under Curve）：

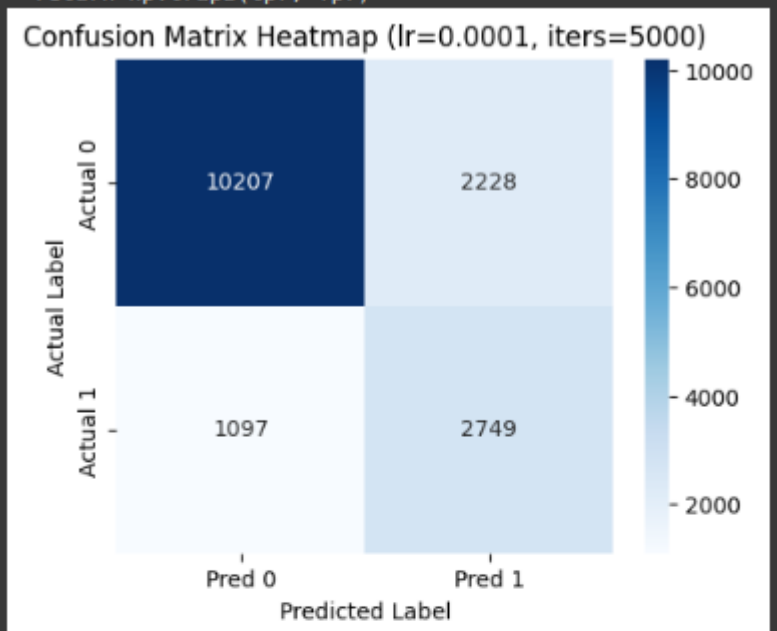
The area under the ROC curve indicates the classification ability of the model.
AUC = 1: Perfect model
AUC = 0.5: Random guessing
AUC > 0.7: The model has certain classification ability

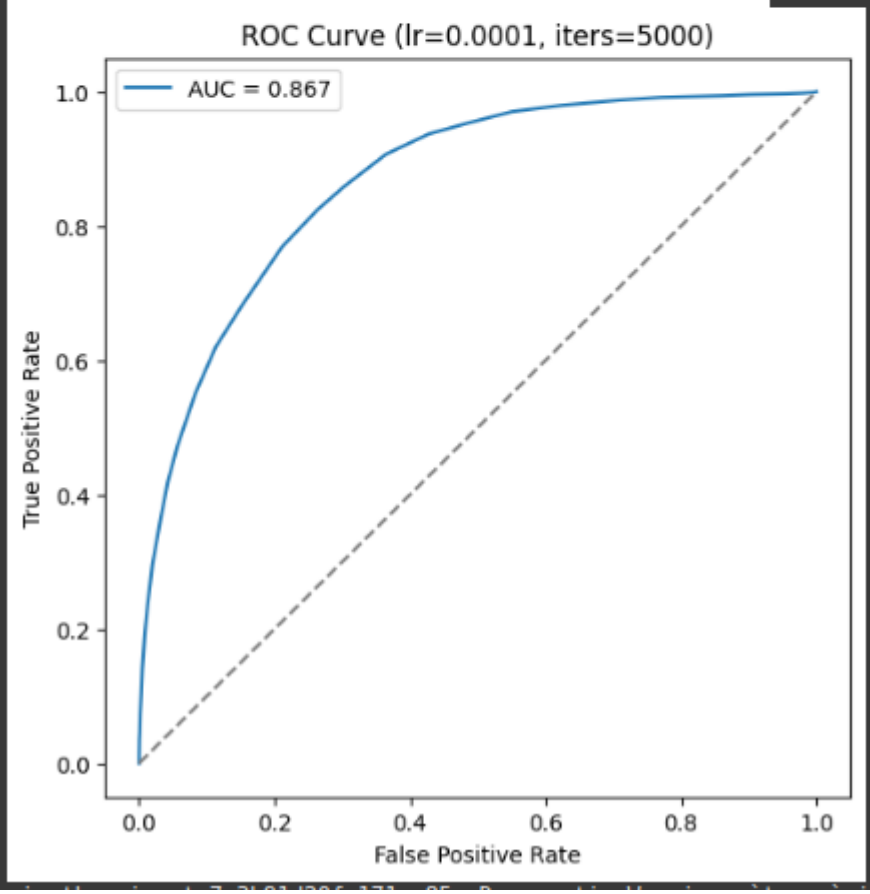
How does it affect model selection?
Precision vs. Recall Trade-off: Depends on the application scenario. If False Negative is more serious (such as medical diagnosis), we pay more attention to Recall.
AUC can help us select a better model. A high AUC indicates that the model can maintain stable predictive ability under different thresholds.
.

Results:

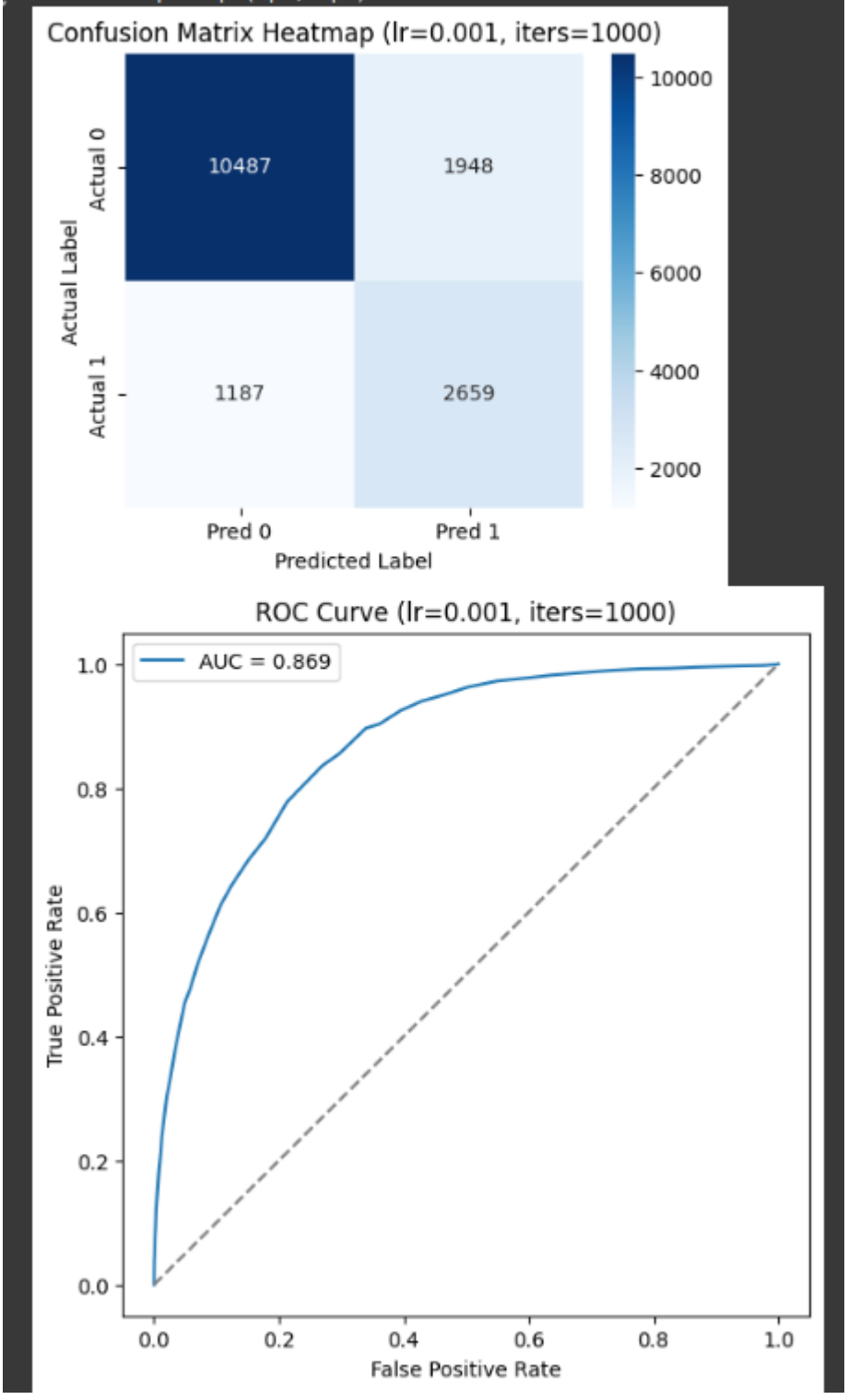

<ipython-input-7-3b91d20fc171>:85: DeprecationWarning: `trapz` is
return np.trapz(tpr, fpr)
Confusion Matrix Heatmap (lr=0.01, iters=3000)
Actual Label
Actual 0
Actual 1
11619
816
1739
2107
10000
8000
6000
4000
2000
Pred 0
Pred 1
Predicted Label
ROC Curve (lr=0.01, iters=3000)
AUC = 0.891
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
<ipython-input-7-3b91d20fc171>:85: DeprecationWarning: `trapz` is

```
<ipython-input-7-3b91d20fc171>:85: DeprecationWarning: `trapz` is d
  return np.trapz(tpr, fpr)
```





```
return np.trapz(tpr, fpr)
```



```
Learning Rate: 0.01, Iterations: 3000
Precision: 0.721, Recall: 0.548, Accuracy: 0.843, F1-Score: 0.623, AUC: 0.891

Learning Rate: 0.0001, Iterations: 5000
Precision: 0.552, Recall: 0.715, Accuracy: 0.796, F1-Score: 0.623, AUC: 0.867

Learning Rate: 0.001, Iterations: 1000
Precision: 0.577, Recall: 0.691, Accuracy: 0.807, F1-Score: 0.629, AUC: 0.869
```