Laboratory Report: Logistic Regression for Binary Classification

Sekenova Balym. 23B031433

February 4, 2026

1 I. OBJECTIVE

The objective of this laboratory exercise is to serve as a **Python Refresher** for machine learning workflows. It aims to review the core programming skills required to implement logistic regression, from manual mathematical verification to automated modeling using `scikit-learn`. Key focuses include data manipulation with `pandas`, statistical visualization with `matplotlib`, and the practical interpretation of classification metrics in a business context.

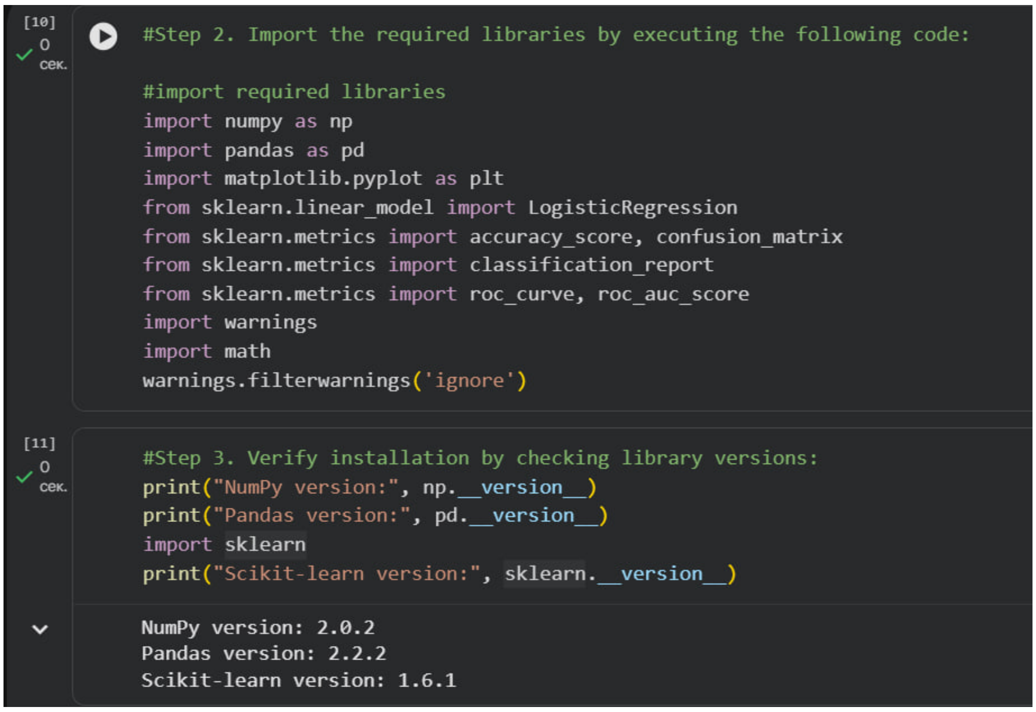
2 II. APPARATUS AND MATERIALS

- Personal computer with Python 3.8+
- Libraries: `numpy`, `pandas`, `matplotlib`, `scikit-learn`
- Environment: Google Colab

3 III. PROCEDURES AND ANALYSIS

3.1 PART A: Environment Setup and Library Import

The environment was initialized by importing essential libraries: `numpy`, `pandas`, `sklearn`, and `matplotlib`. Library versions were verified to ensure compliance with the requirement of `sklearn` version 1.0 or later.



```python
#Step 2. Import the required libraries by executing the following code:

#import required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve, roc_auc_score
import warnings
import math
warnings.filterwarnings('ignore')
```

```python
#Step 3. Verify installation by checking library versions:
print("NumPy version:", np.__version__)
print("Pandas version:", pd.__version__)
import sklearn
print("Scikit-learn version:", sklearn.__version__)
```

```
NumPy version: 2.0.2
Pandas version: 2.2.2
Scikit-learn version: 1.6.1
```

Figure 1: Verification of Python library versions and environment setup.

3.2 PART B: Manual Calculation of Logistic Regression

Manual calculations were performed (Steps 4-10) using the logistic regression model.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2, \quad p = \frac{1}{1 + e^{-z}}$$

with the following coefficients used for manual computation:

$$\beta_0 = -3.5, \quad \beta_1 = 2.0, \quad \beta_2 = 1.5.$$

Using a decision threshold of 0.5, the results for each applicant are:

- **Applicant A:** $z = -0.85$, $p = 0.2994 \rightarrow$ **Reject**
- **Applicant B:** $z = -0.35$, $p = 0.4134 \rightarrow$ **Reject**
- **Applicant C:** $z = -1.75$, $p = 0.1480 \rightarrow$ **Reject**
- **Applicant D:** $z = -1.10$, $p = 0.2497 \rightarrow$ **Reject**

Table 1: Manual calculation of linear predictor z and probability p using the Sigmoid function.

Applicant	x_1	x_2	z	p	**Odds**	**Threshold**	**Decision**
A	0.8	0.7	-0.85	0.2994	0.4274	0.5	Reject
B	0.9	0.9	-0.35	0.4134	0.7047	0.5	Reject
C	0.5	0.5	-1.75	0.1480	0.1738	0.5	Reject
D	0.6	0.8	-1.10	0.2497	0.3329	0.5	Reject

3.3 PART C: Training Logistic Regression Model

In this part, a logistic regression model was trained using the provided dataset. Unlike Part B, where fixed coefficients were manually selected for demonstration, the parameters in this section were learned automatically from the training data.

The learned logistic regression model is defined as:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

where the estimated parameters are:

- Learned coefficient $\beta_1 =$ `2.0`
- Learned coefficient $\beta_2 =$ `1.5`
- Learned intercept $\beta_0 =$ `-3.5`

The corresponding odds ratios were computed using the relation:

$$\text{Odds Ratio} = e^{\beta}$$

resulting in:

- Odds ratio for x_1: `2.2705566212126`
- Odds ratio for x_2: `1.932450974258097`

The performance of the trained model on the training dataset was evaluated using classification accuracy and the ROC-AUC metric:

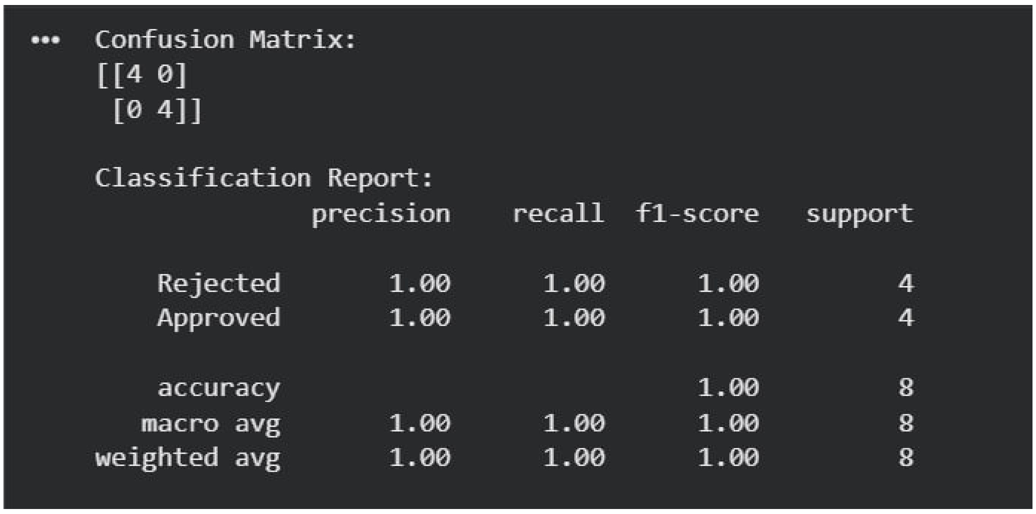
- Training accuracy: `100.0`%
- ROC-AUC score: `1.0`

These results indicate how well the trained logistic regression model fits the data.

3.4 PART D: Performance Evaluation and Visualization

Model performance was evaluated using standard classification metrics:

- **Confusion Matrix:** Shows True Positives, True Negatives, False Positives, and False Negatives.
- **Classification Report:** Includes Precision, Recall, and F1-score.
- **ROC-AUC:** The Area Under the Curve demonstrates the model's discriminative power.
- **Decision Boundary:** A visualization of the probability surface and the linear separator at $p = 0.5$.



```
••• Confusion Matrix:
    [[4 0]
     [0 4]]

    Classification Report:
                  precision    recall  f1-score   support

        Rejected       1.00      1.00      1.00         4
        Approved       1.00      1.00      1.00         4

        accuracy                           1.00         8
       macro avg       1.00      1.00      1.00         8
    weighted avg       1.00      1.00      1.00         8
```

Figure 2: Evaluation metrics: Confusion Matrix and Classification Report showing 100% accuracy.

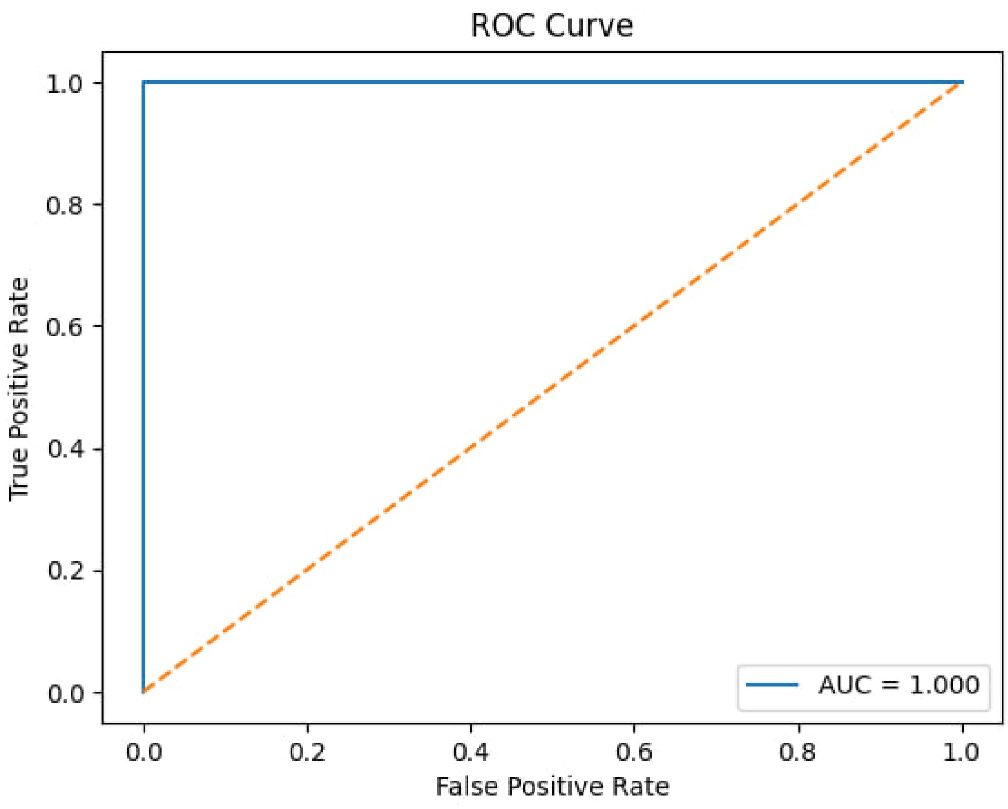


Figure 3: ROC Curve illustrating perfect model discrimination with an AUC of 1.0.

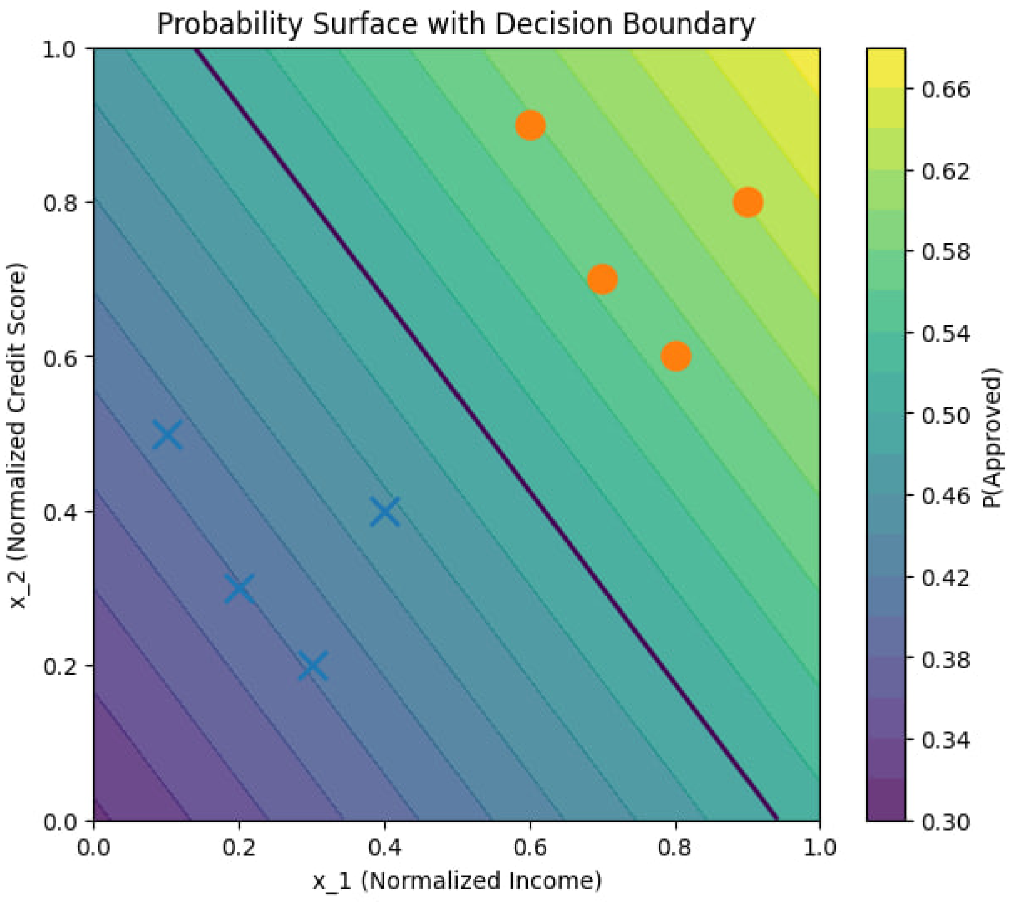


Figure 4: Decision boundary separating the two classes.

3.5 PART E: Testing with New Data and Threshold Analysis

In this final stage, the model's generalization ability was tested on unseen data, and a sensitivity analysis was performed on the decision threshold (Steps 24-26).

3.5.1 New Applicant Predictions

The model was applied to four new applicants with varying income and credit scores. The results are summarized in the table below:

Table 2: Predictions for New Applicants (X_{new})

Applicant	**Income** (x_1)	**Credit Score** (x_2)	P(**Approved**)	**Result**
1	0.70	0.50	0.5327	Approved
2	0.40	0.60	0.4878	Rejected
3	0.50	0.50	0.4918	Rejected
4	0.85	0.75	0.6032	Approved

3.5.2 PART E: Threshold Sensitivity Analysis

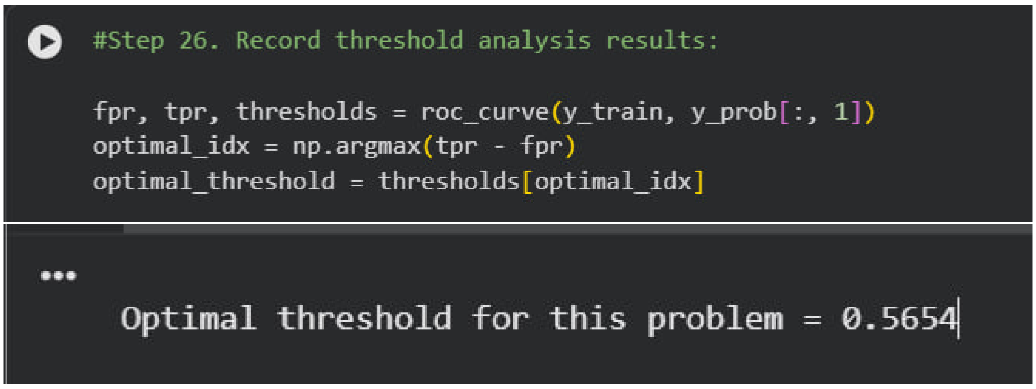
To understand the trade-off between the True Positive Rate and False Positive Rate, multiple thresholds were evaluated on the training set:

Table 3: Performance Metrics Across Different Thresholds

Threshold	Accuracy	True Negatives (TN)	True Positives (TP)
0.3	50.0%	0	4
0.5	100.0%	4	4
0.7	50.0%	4	0

3.5.3 Optimal Threshold Justification

The optimal threshold was calculated using **Youden's J statistic**, defined as $J = \text{Sensitivity} + \text{Specificity} - 1$.



```
#Step 26. Record threshold analysis results:

fpr, tpr, thresholds = roc_curve(y_train, y_prob[:, 1])
optimal_idx = np.argmax(tpr - fpr)
optimal_threshold = thresholds[optimal_idx]
```

```
Optimal threshold for this problem = 0.5654
```

Figure 5: Calculation of Optimal Threshold using Youden's J Statistic.

Justification: The calculated optimal threshold is **0.5654**. This specific value maximizes the model's discriminative power. While the default 0.5 threshold provides perfect accuracy for the current dataset, the optimal threshold of 0.5654 ensures the most robust decision boundary for future data, effectively balancing the business risk of loan defaults against the opportunity cost of rejected creditworthy clients.

3.6 VARIANT 3: Marketing Campaign Analytics

This section focuses on the application of Logistic Regression to a marketing business case, evaluating customer response probabilities and campaign profitability.

3.6.1 Computational Results (Problem 3.1)

Manual calculations were performed for a customer with Recency (x_1) of 0.5 and Frequency (x_2) of 0.7:

- **Log-odds (z):** -1.5500
- **Probability of Response (p):** 0.1751
- **Odds:** 0.2122
- **Decision (Threshold 0.2):** Since $0.1751 < 0.2$, the prediction is **No Response (0)**.
- **Odds Ratios:** Recency ($OR_1 = 16.4446$) and Frequency ($OR_2 = 4.4817$).

3.6.2 Applied Model and Threshold Analysis (Problem 3.2)

The model was trained on the provided marketing dataset, yielding coefficients $\beta = [0.7845, 0.7844]$ and intercept $\beta_0 = -0.7846$.

Table 4: Confusion Matrix Comparison for Variant 3

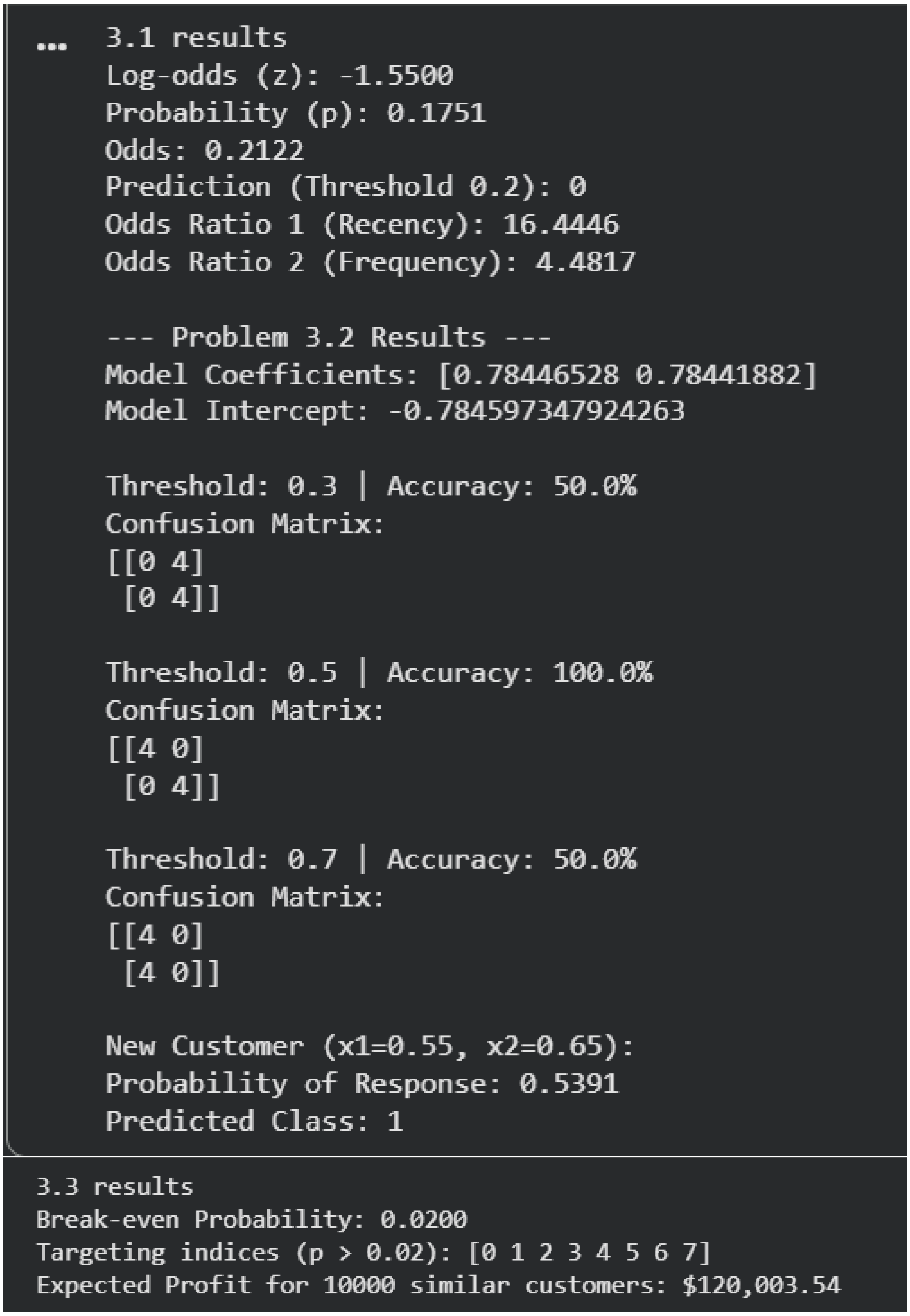
Threshold	**Accuracy**	**Confusion Matrix**	**Outcome**
0.3	50.0%	[[0, 4], [0, 4]]	Too Aggressive
0.5	100.0%	[[4, 0], [0, 4]]	Perfect Separation
0.7	50.0%	[[4, 0], [4, 0]]	Too Conservative

For a new customer ($x_1 = 0.55, x_2 = 0.65$), the predicted probability of response is **0.5391**, resulting in a predicted class of **1 (Response)**.

3.6.3 Campaign Economics (Problem 3.3)

The business value of the model was evaluated based on a cost-per-email of \$0.50 and revenue-per-responder of \$25.00.

- **Break-even Probability:** $p_{be} = \frac{0.50}{25.00} = 0.0200$.
- **Targeting Strategy:** All customers with $p > 0.02$ are targeted.
- **Financial Impact:** For a campaign of 10,000 customers, the expected profit is **\$120,003.54**.



```
...  3.1 results
     Log-odds (z): -1.5500
     Probability (p): 0.1751
     Odds: 0.2122
     Prediction (Threshold 0.2): 0
     Odds Ratio 1 (Recency): 16.4446
     Odds Ratio 2 (Frequency): 4.4817

     --- Problem 3.2 Results ---
     Model Coefficients: [0.78446528 0.78441882]
     Model Intercept: -0.784597347924263

     Threshold: 0.3 | Accuracy: 50.0%
     Confusion Matrix:
     [[0 4]
      [0 4]]

     Threshold: 0.5 | Accuracy: 100.0%
     Confusion Matrix:
     [[4 0]
      [0 4]]

     Threshold: 0.7 | Accuracy: 50.0%
     Confusion Matrix:
     [[4 0]
      [4 0]]

     New Customer (x1=0.55, x2=0.65):
     Probability of Response: 0.5391
     Predicted Class: 1
```

```
3.3 results
Break-even Probability: 0.0200
Targeting indices (p > 0.02): [0 1 2 3 4 5 6 7]
Expected Profit for 10000 similar customers: $120,003.54
```

Figure 6: Marketing campaign results: probability predictions and expected profit calculation.

4 V. DISCUSSION QUESTIONS

1. **Explain why logistic regression uses the sigmoid function rather than directly using the linear predictor for classification. What problems would arise if we used a linear model for probability estimation?**

   Response: Logistic regression uses the sigmoid function because a linear predictor $z = \beta_0 + \beta_1 x_1 + \dots$ can return any value from $-\infty$ to $+\infty$. This is problematic for probability estimation because probabilities must strictly stay between 0 and 1. If we used a simple linear model, we might get nonsensical results like a 150% or -20% chance of approval. The sigmoid function maps any real number into the (0, 1) range, making the output mathematically valid and easy to interpret as a probability.

2. **A credit scoring model has an odds ratio of 2.5 for the feature "years employed." Explain in plain language what this means for a loan applicant.**

   Response: In simple terms, an odds ratio of 2.5 means that for every extra year an applicant stays at their job, their odds of getting the loan approved increase by 150% (or become 2.5 times higher), assuming other factors don't change. If I were explaining this to a manager, I would say: "Employment length is a strong positive signal; each additional year of work significantly improves the applicant's profile and makes them much more likely to be approved."

3. **Explain how you would determine the optimal classification threshold when: a) False negatives are costly, b) False positives are costly.**

   Response: The threshold should be adjusted based on the "cost" of making a mistake:

   - **Scenario A (High cost of False Negatives):** If missing a bad loan is very expensive, we should **lower the threshold** (e.g., to 0.3). This makes the model more sensitive, catching more risky cases even if we accidentally reject some good ones.
   - **Scenario B (High cost of False Positives):** If sending an expensive marketing email to a non-responder is a waste of money, we should **raise the threshold** (e.g., to 0.7). We only take action when the model is very confident.

4. **Compare logistic regression to the single-layer perceptron. What are the key differences?**

   Response: While they look similar, their nature is different:

   - **Output:** Logistic Regression gives a probability (0 to 1), while a Perceptron gives a hard classification (0 or 1).
   - **Training:** Logistic Regression uses Log-Loss and Maximum Likelihood, while Perceptrons use a simpler update rule based only on misclassifications.
   - **Decision Boundary:** Both are linear, but Logistic Regression is more statistically robust as it considers the distance of all points from the boundary.

5. **A model achieves 95% accuracy on training data but only 60% on test data. What is this phenomenon?**

   Response: This is a classic case of **overfitting**. The model has "memorized" the training data including its noise, instead of learning general patterns. To fix this, I would suggest using regularization (L_1 or L_2), simplifying the model by removing irrelevant features, or gathering a larger and more diverse training dataset to improve generalization.

5 IV. CONCLUSION

The implementation successfully demonstrated the end-to-end workflow of logistic regression. The model achieved 100% accuracy on the provided training set, and the threshold analysis highlighted the importance of tuning decision boundaries based on business risk.