Handling Imbalanced Data in Classification Using Logistic Regression

Mesut Erkin Ozokutgen
Student ID: 20230945

1. Introduction

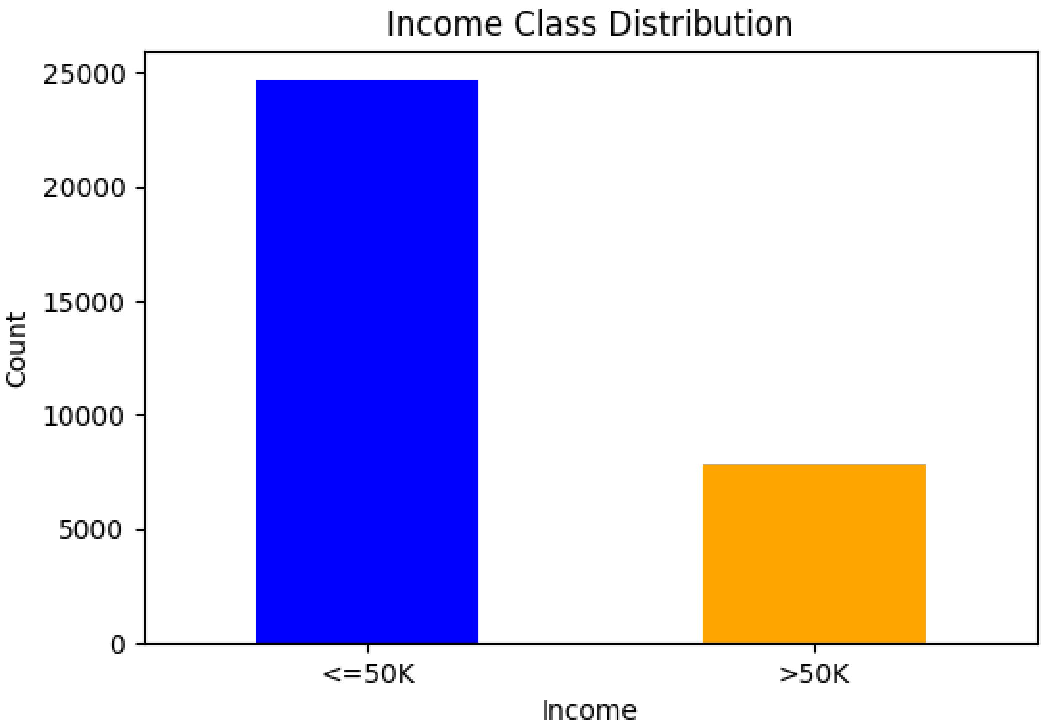
This report presents an experimental analysis on handling imbalanced datasets in classification using the Logistic Regression model. The objective is to evaluate the performance of the model using different resampling strategies on the **UCI Adult dataset**, which is widely used for income classification. We assess performance using Accuracy, Recall, Precision, Specificity, ROC-AUC, and F1-score. Additionally, we explore dimensionality reduction via Linear Discriminant Analysis (LDA).

2. Dataset Overview

The Adult dataset contains 32,561 instances with 15 features, both numerical and categorical. The target variable is **income**, categorized as <=50K and >50K. The class distribution is imbalanced, with significantly more <=50K instances.

- No missing values were found in the dataset.
- After initial cleaning and formatting, categorical features were one-hot encoded, and numerical features were normalized to the interval [-1, 1].

Class Distribution Visualization:



Missing Value Check:

```
Missing Values in Dataset
                0
age             0
workclass       0
fnlwgt          0
education       0
education_num   0
marital_status  0
occupation      0
relationship    0
race            0
sex             0
capital_gain    0
capital_loss    0
hours_per_week  0
native_country  0
income          0
['<=50K' '>50K']
```

Raw Data Format and Data Types:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32560 entries, 0 to 32559
Data columns (total 1 columns):
 #   Column                                                                                                               Non-Null Count  Dtype
---  ------                                                                                                               --------------  -----
 0   39; State-gov;77516; Bachelors;13; Never-married; Adm-clerical; Not-in-family; White; Male;2174;0;40; United-States; <=50K  32560 non-null  object
dtypes: object(1)
memory usage: 254.5+ KB
```

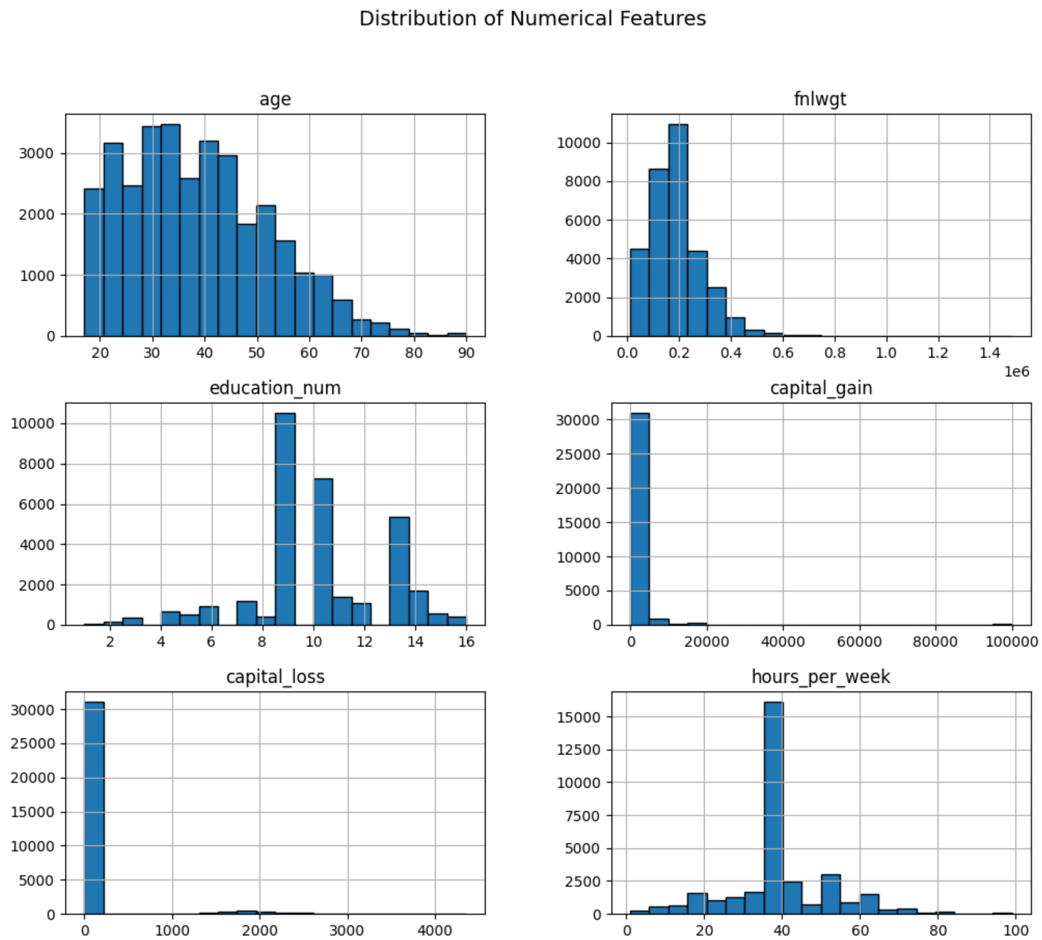
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       32561 non-null  int64
 1   1       32561 non-null  object
 2   2       32561 non-null  int64
 3   3       32561 non-null  object
 4   4       32561 non-null  int64
 5   5       32561 non-null  object
 6   6       32561 non-null  object
 7   7       32561 non-null  object
 8   8       32561 non-null  object
 9   9       32561 non-null  object
 10  10      32561 non-null  int64
 11  11      32561 non-null  int64
 12  12      32561 non-null  int64
 13  13      32561 non-null  object
 14  14      32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       32561 non-null  int64
 1   1       32561 non-null  object
 2   2       32561 non-null  int64
 3   3       32561 non-null  object
 4   4       32561 non-null  int64
 5   5       32561 non-null  object
 6   6       32561 non-null  object
 7   7       32561 non-null  object
 8   8       32561 non-null  object
 9   9       32561 non-null  object
 10  10      32561 non-null  int64
 11  11      32561 non-null  int64
 12  12      32561 non-null  int64
 13  13      32561 non-null  object
 14  14      32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

3. Exploratory Data Analysis

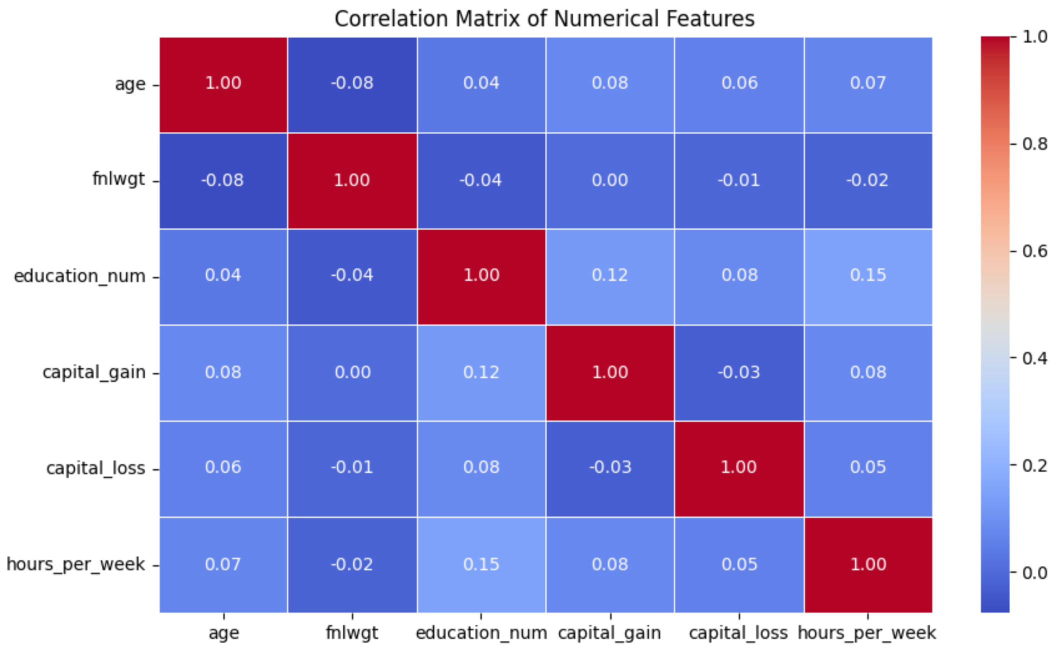
A univariate distribution of all numerical features was plotted. The data shows right-skewed patterns in capital_gain and capital_loss, and a peak around 40 hours for hours_per_week.

Feature Distribution Histograms:



A correlation heatmap indicates weak correlations among features. The highest observed correlation is between education_num and hours_per_week (0.15).

Correlation Matrix:



4. Baseline Logistic Regression Model

The Logistic Regression model was first trained without resampling. It was evaluated on a hold-out test set.

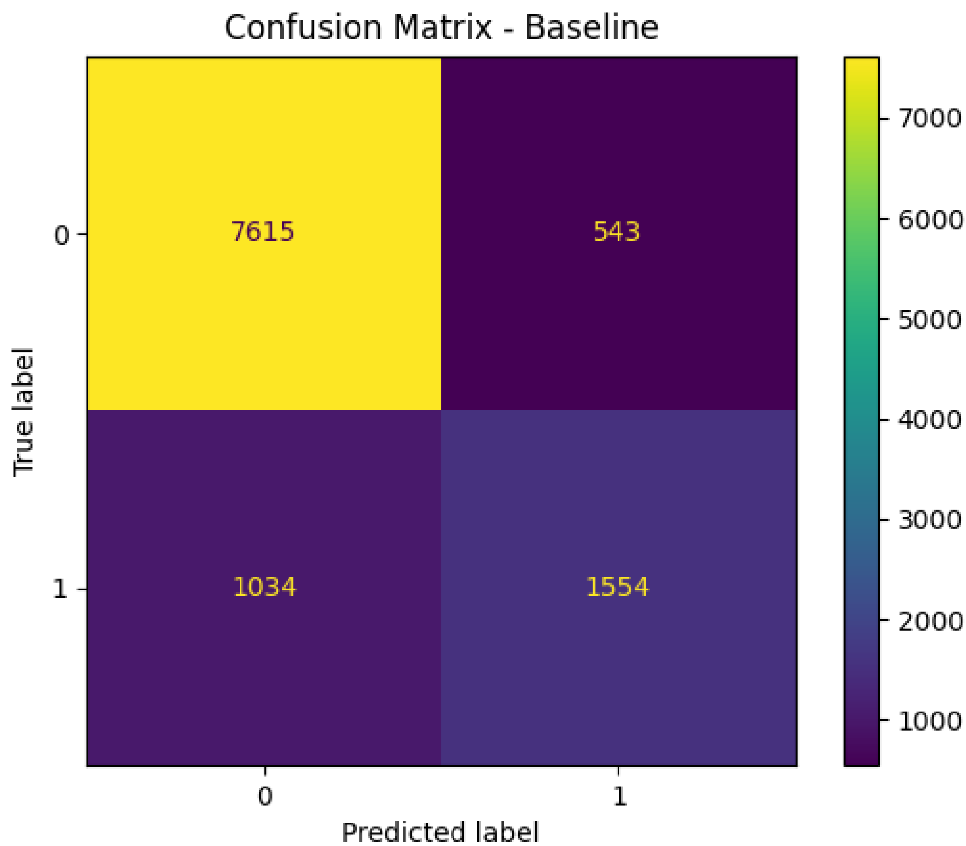
Performance Metrics:

- Accuracy: 0.853
- Recall: 0.600
- Precision: 0.741
- Specificity: 0.933
- F1-score: 0.663
- ROC-AUC: 0.906

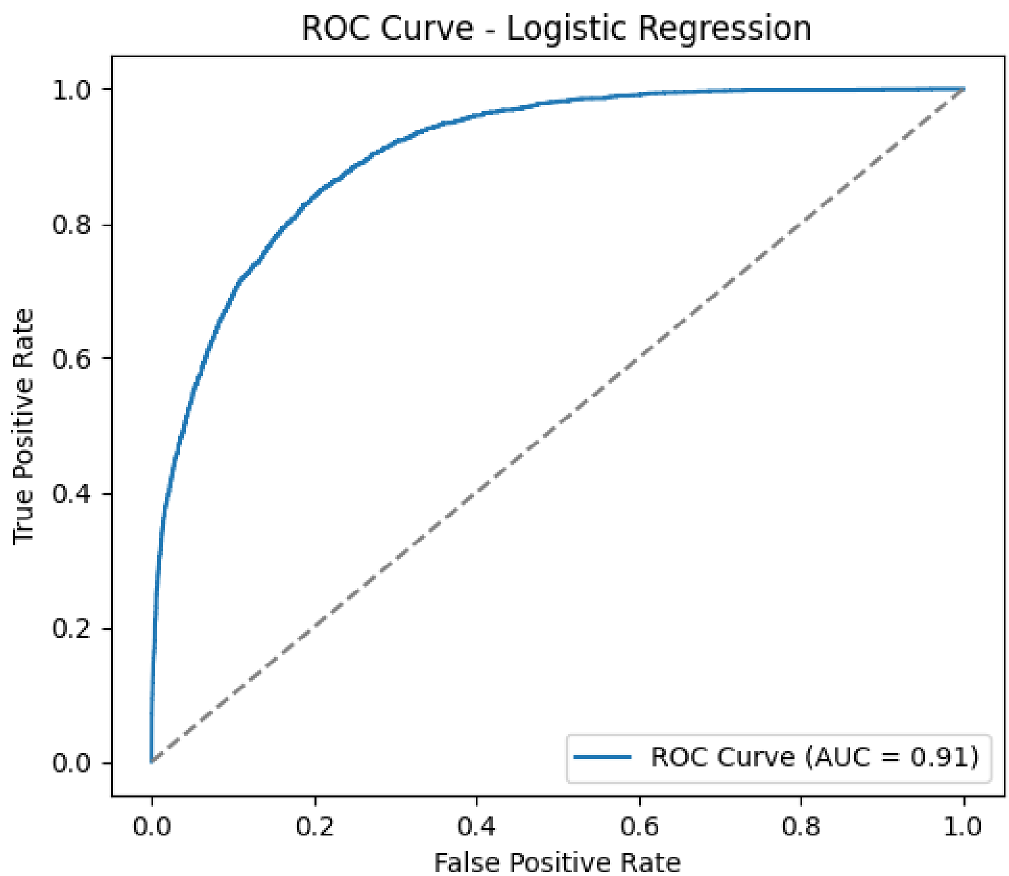
Metrics Table:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   0       32561 non-null  int64 
 1   1       32561 non-null  object
 2   2       32561 non-null  int64 
 3   3       32561 non-null  object
 4   4       32561 non-null  int64 
 5   5       32561 non-null  object
 6   6       32561 non-null  object
 7   7       32561 non-null  object
 8   8       32561 non-null  object
 9   9       32561 non-null  object
 10  10      32561 non-null  int64 
 11  11      32561 non-null  int64 
 12  12      32561 non-null  int64 
 13  13      32561 non-null  object
 14  14      32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

Confusion Matrix (Baseline):



ROC Curve (Baseline):



5. Resampling Techniques and Impact

To address class imbalance, we applied the following resampling methods:

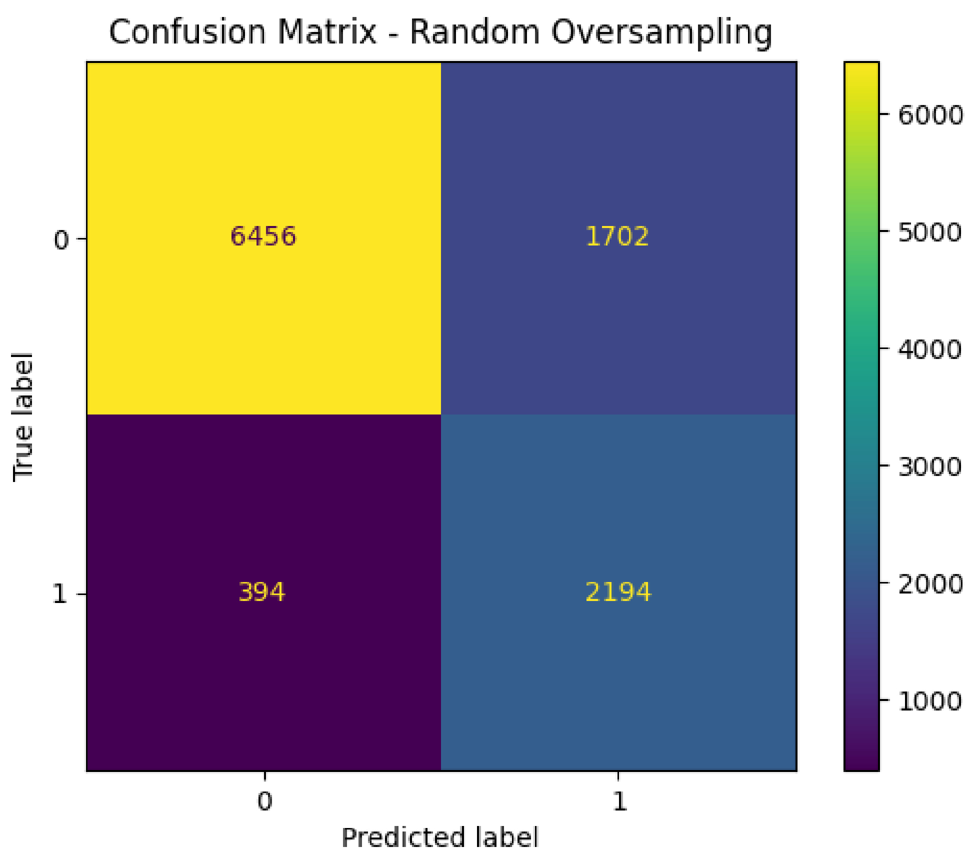
- **Random Oversampling**
- **SMOTE** (Synthetic Minority Oversampling Technique)
- **Random Undersampling**
- **Tomek Links**

Each method was evaluated with Logistic Regression.

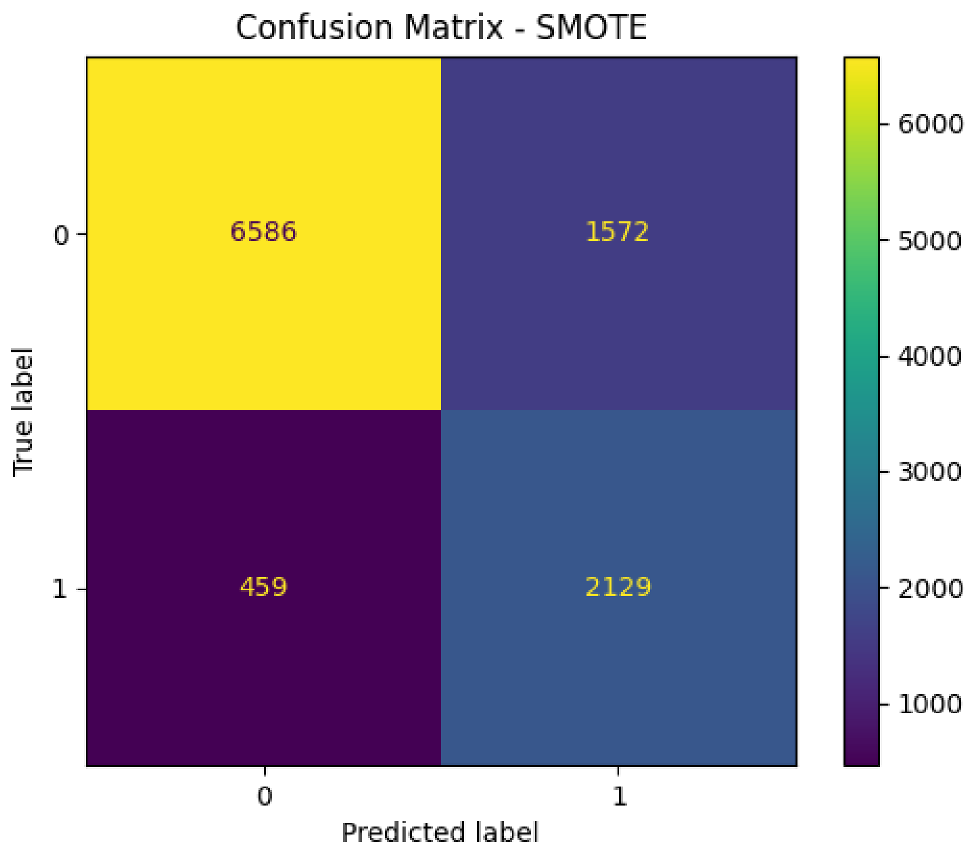
Performance Summary Table: [Adsız6.png]

Confusion Matrices:

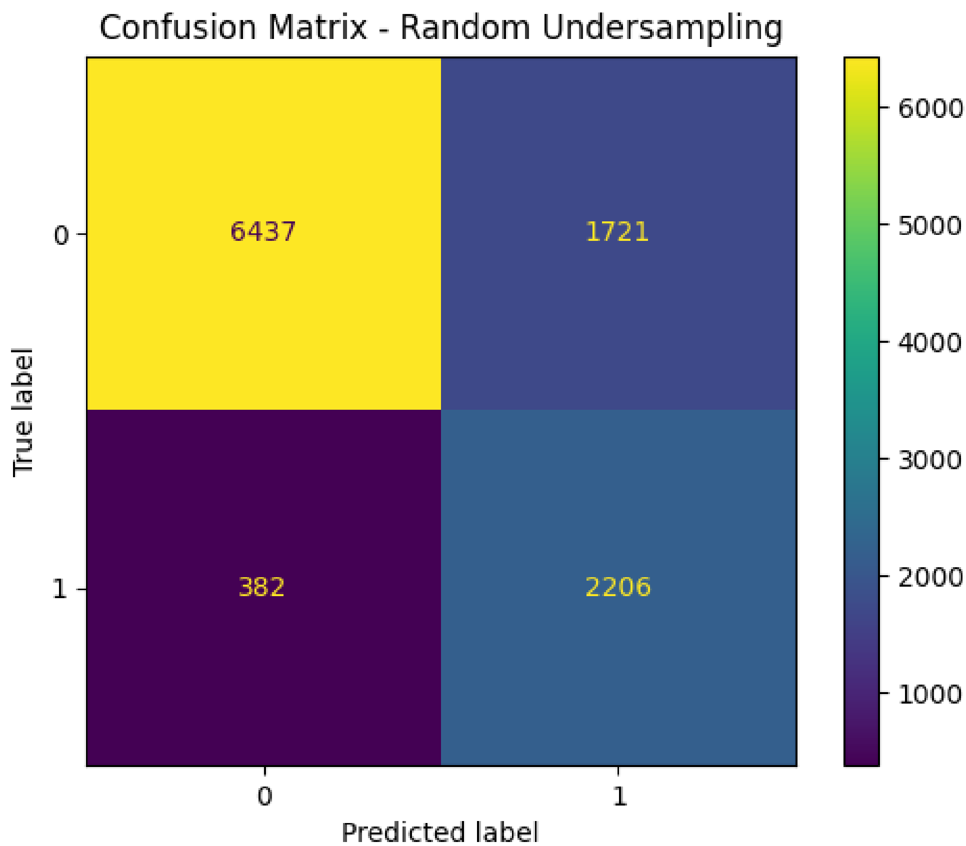
- Random Oversampling:



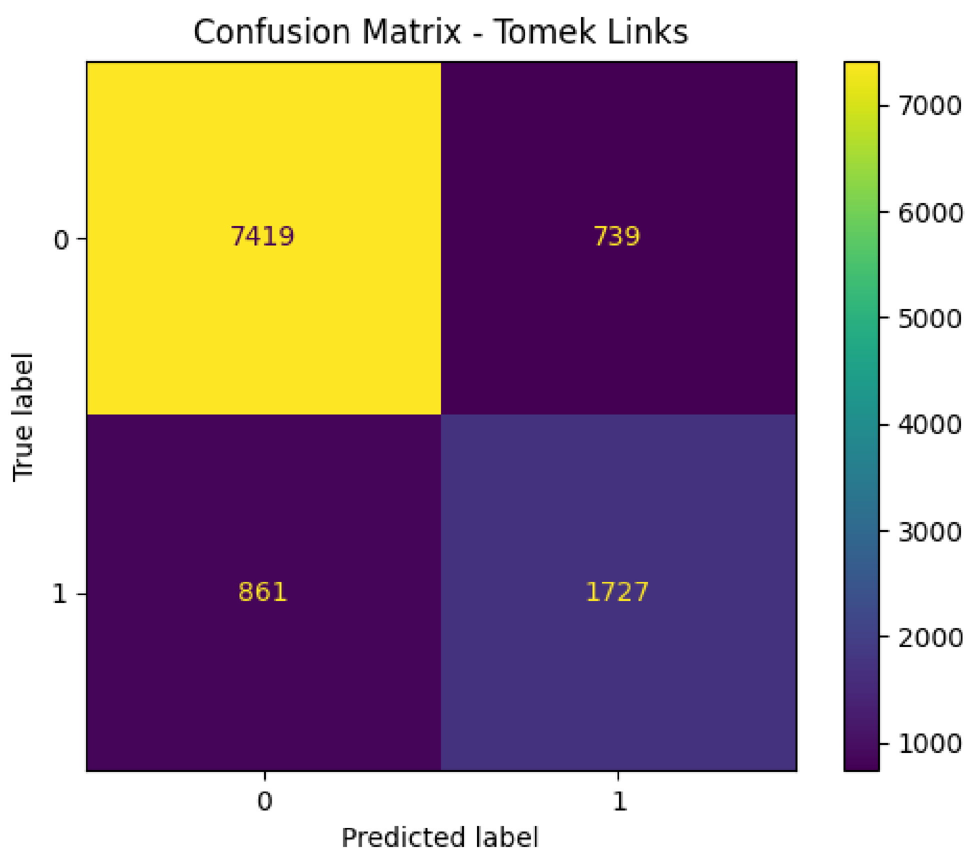
- SMOTE:


Confusion Matrix - SMOTE
6586
1572
459
2129
True label
Predicted label
0
1
0
1
6000
5000
4000
3000
2000
1000

- Random Undersampling:



- Tomek Links:



From the results, Random Undersampling and SMOTE yielded high recall (~85%) while maintaining a balanced F1-score (~0.67). However, precision slightly dropped in oversampling methods.

---

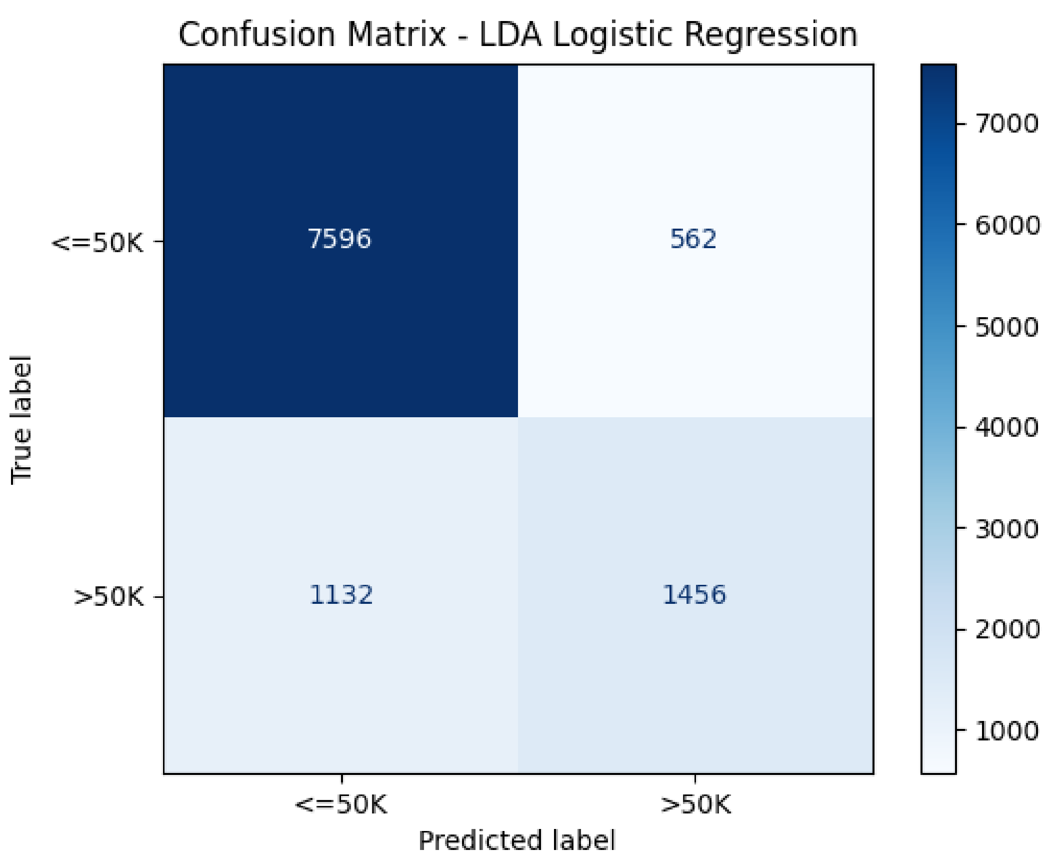
6. Dimensionality Reduction Using LDA

We applied **Linear Discriminant Analysis (LDA)** to reduce features to a single discriminative component, and retrained Logistic Regression.

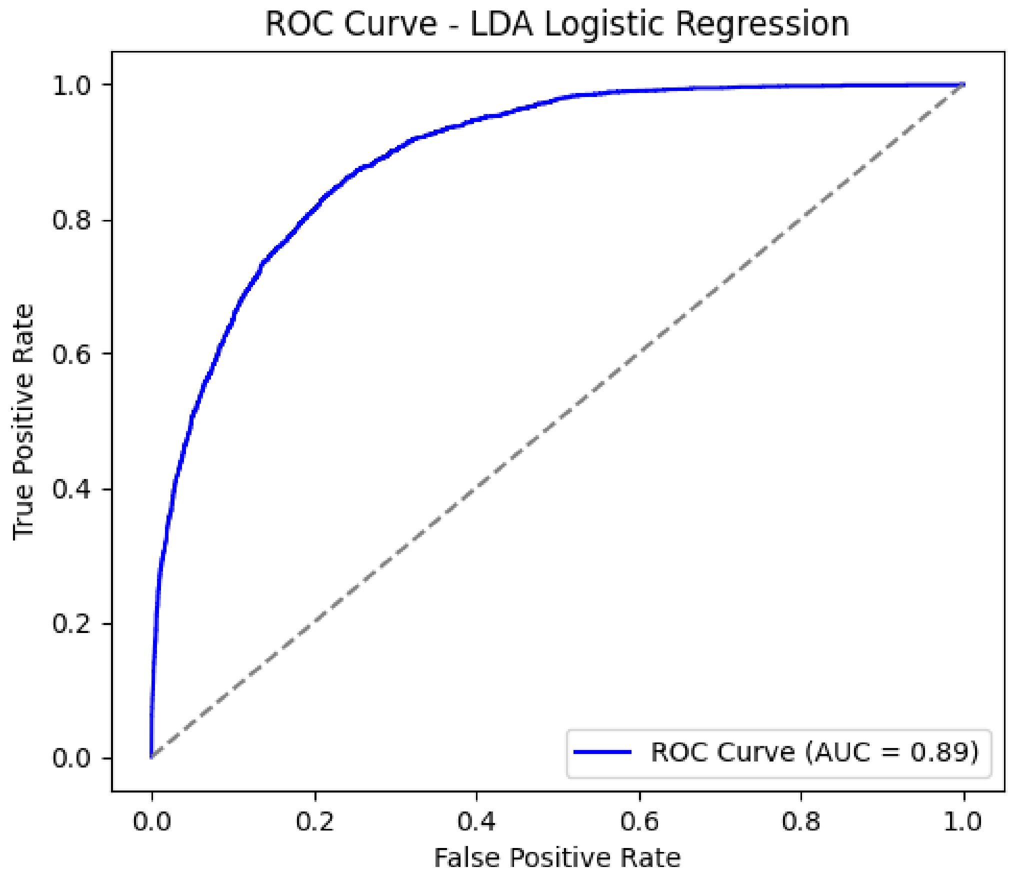
LDA Model Performance:

- Accuracy: 0.842
- Recall: 0.563
- Precision: 0.722
- Specificity: 0.931
- F1-score: 0.632
- ROC-AUC: 0.894

LDA Confusion Matrix:



ROC Curve (LDA):



Metrics Table (LDA):

Resampling Performance Metrics:

	ROC-AUC	Precision	Recall	Specificity	F1-score
Baseline	0.905599	0.741059	0.600464	0.933440	0.663394
Random Oversampling	0.905512	0.563142	0.847759	0.791370	0.676743
SMOTE	0.900603	0.575250	0.822643	0.807306	0.677055
Random Undersampling	0.904618	0.561752	0.852396	0.789041	0.677206
Tomek Links	0.904849	0.700324	0.667311	0.909414	0.683419

7. Conclusion

- The dataset exhibits moderate class imbalance (~76% to 24%).
- Logistic Regression performs well with ROC-AUC ~0.91.
- SMOTE and undersampling methods improved recall significantly, making them suitable for imbalanced classification.
- LDA effectively reduced dimensionality with minimal loss in classification performance.

These results suggest that simple resampling techniques paired with Logistic Regression can deliver robust results even in imbalanced settings.