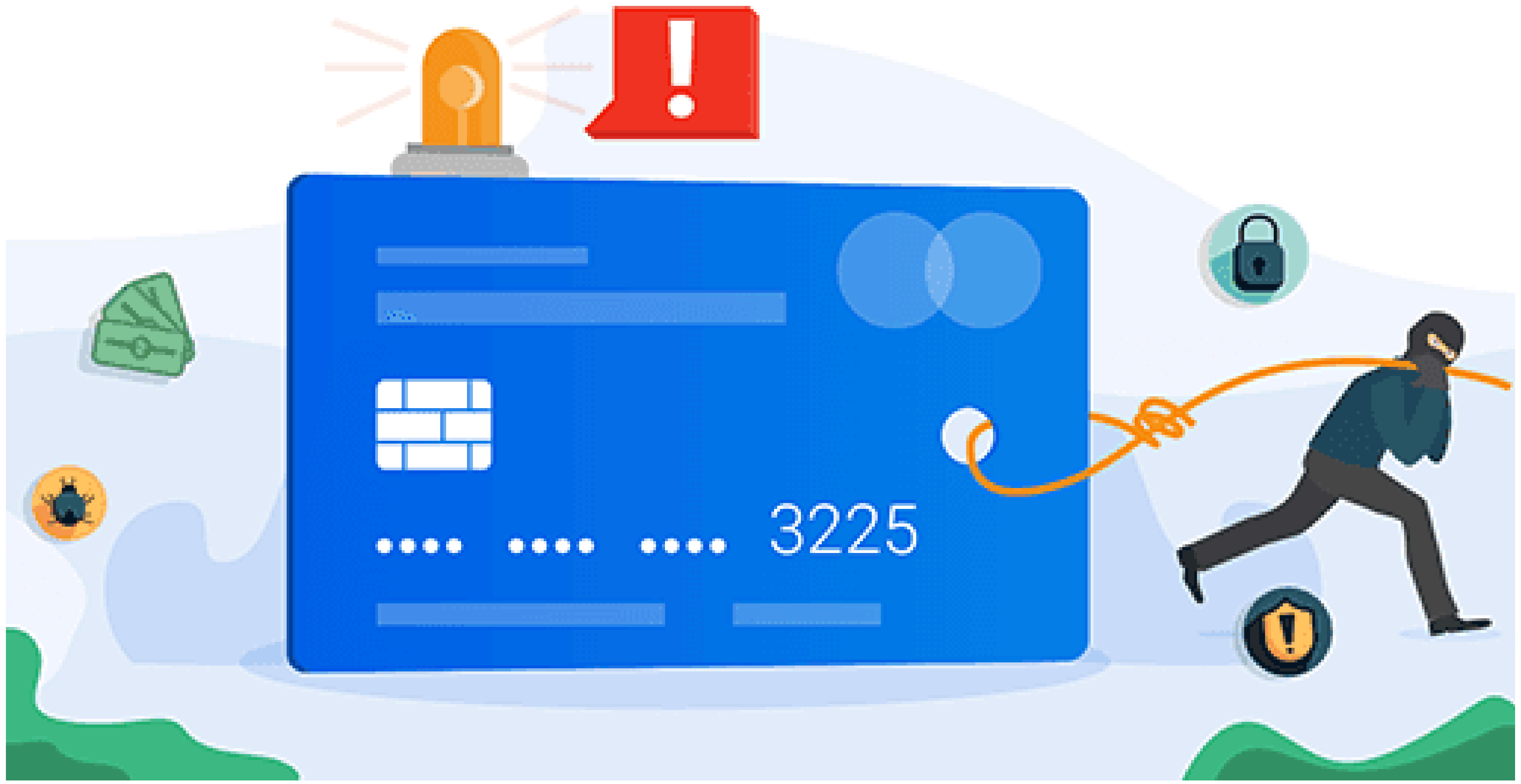


Credit Card Fraud Detection

Presented By : Ololade Akinsanola

COMP 479- MACHINE LEARNING

The Problem

Less than 0.02%

Total credit card transactions are fraudulent, hence they are very hard to predict

$33.83 Billion

Lost worldwide due to credit card frauds in 2023

Our dataset:

Recorded over 2 days in September 2013

284,807 transactions

492 Fraudulent transactions

0.00172% of total transactions

How Fraud is Recognized

These methods are generally employed by credit card companies:

- Location: Purchase made from different location
- Items Bought: deviate from your regular pattern
- Frequency: large number of transactions in a short time period
- Amount: total cost of items purchased

Understanding the Data

- Data is highly skewed
- Contains 31 features
- V1 to V28 have been transformed by PCA for privacy users
- Other features: time, frequency and class (either 1 for fraud or 0 otherwise.)

Data Split

To validate our model’s prediction, we split:

- 80% for training (227,845 rows)
- 20% for testing (57,025 rows)

Oversampling using SMOTE

🔴 Problems with the dataset

- The dataset was highly imbalanced.
  - Legitimate transactions ≫ Fraudulent transactions.
    284315 ≫ 492
  - This can cause classifiers to be biased toward the majority class, ignoring fraud cases.

🟢 Solution : SMOTE (Synthetic Minority Over-sampling Technique)

- SMOTE generates synthetic samples for the minority class (fraud) by interpolating between existing fraud examples.
- Applied only on the training data to avoid data leakage
- Class distribution

  before SMOTE: Counter({0: 227451, 1: 394})
  after SMOTE: Counter({0: 227451, 1: 227451})

⚙️ How SMOTE Works

- Selects minority class instances.
- Finds k-nearest neighbors.
- Generates new synthetic samples between these points.

📈 Results / Impact

- Improved model sensitivity (recall) to fraud cases.
- Reduced false negatives, which is critical in fraud detection.
- Balanced dataset helps the classifier learn both classes effectively.

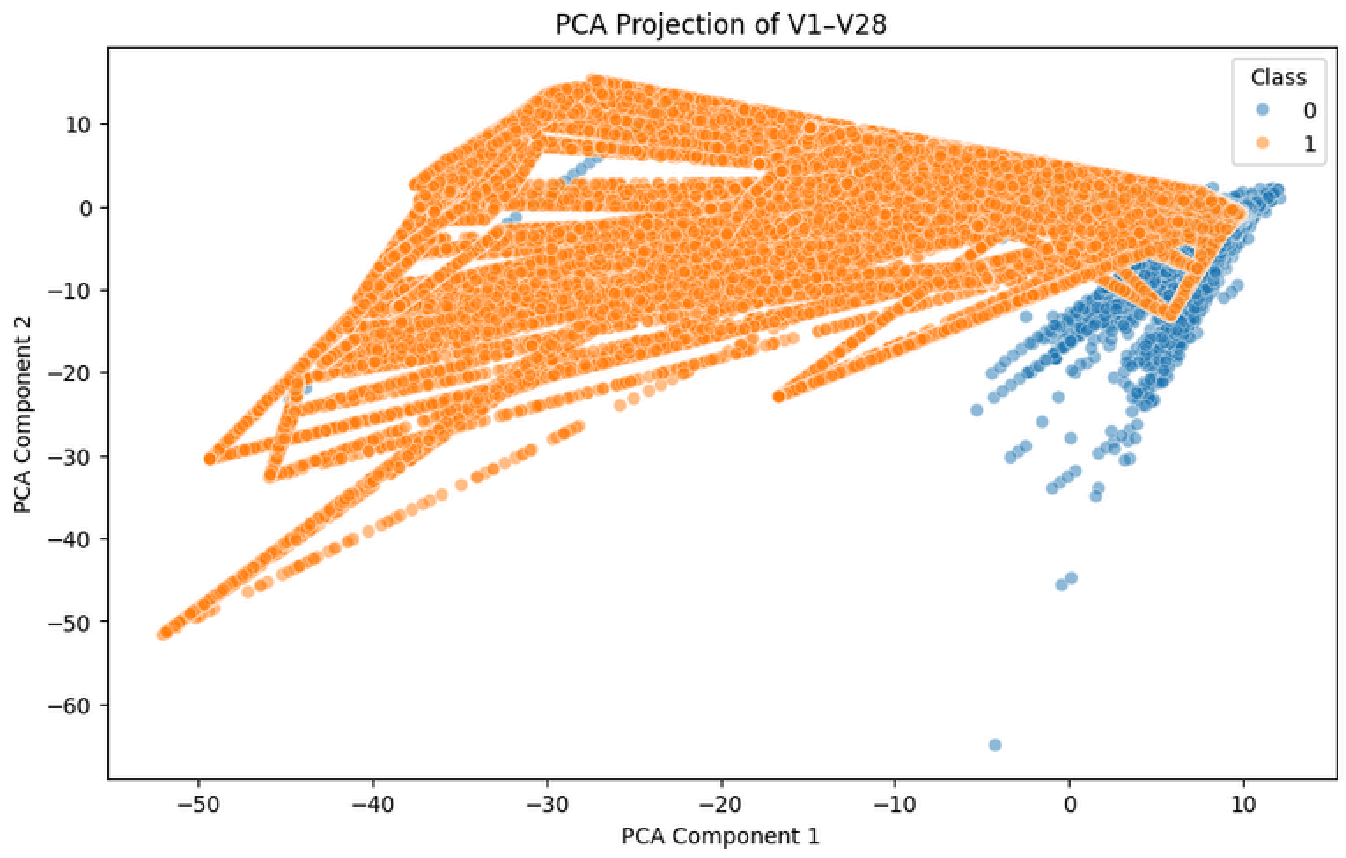
→

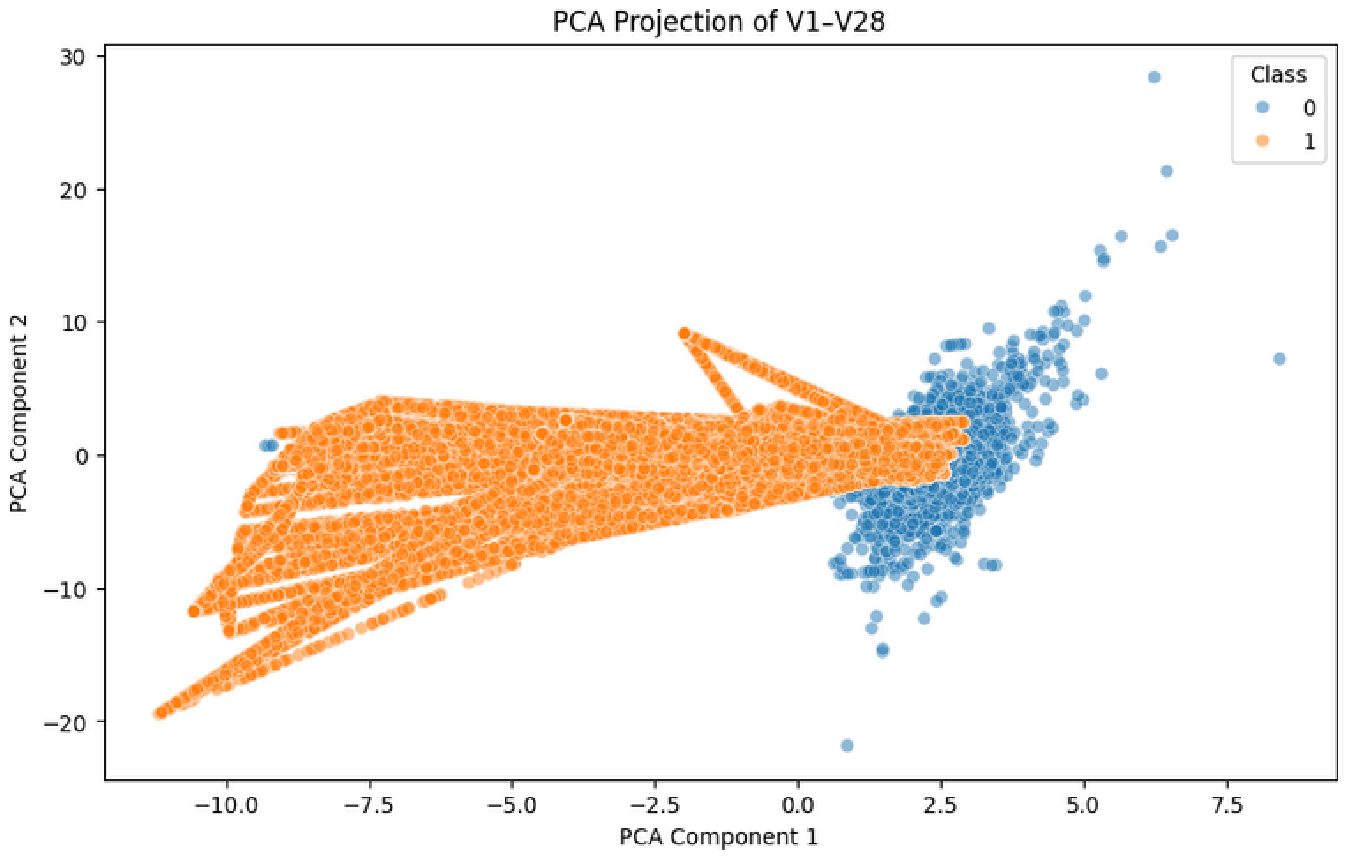
Data Visualization before and after SMOTE

Before

After

Effect of Normalization on PCA Visualization





Before Normalization

- Raw feature values
- Distorted PCA result
- Difficult to distinguish between classes

After Normalization

- Scaled feature values
- PCA reveals clearer class separation
- More suitable for classification

"Normalization ensures PCA and machine learning models focus on real patterns, not on dominant feature scales."

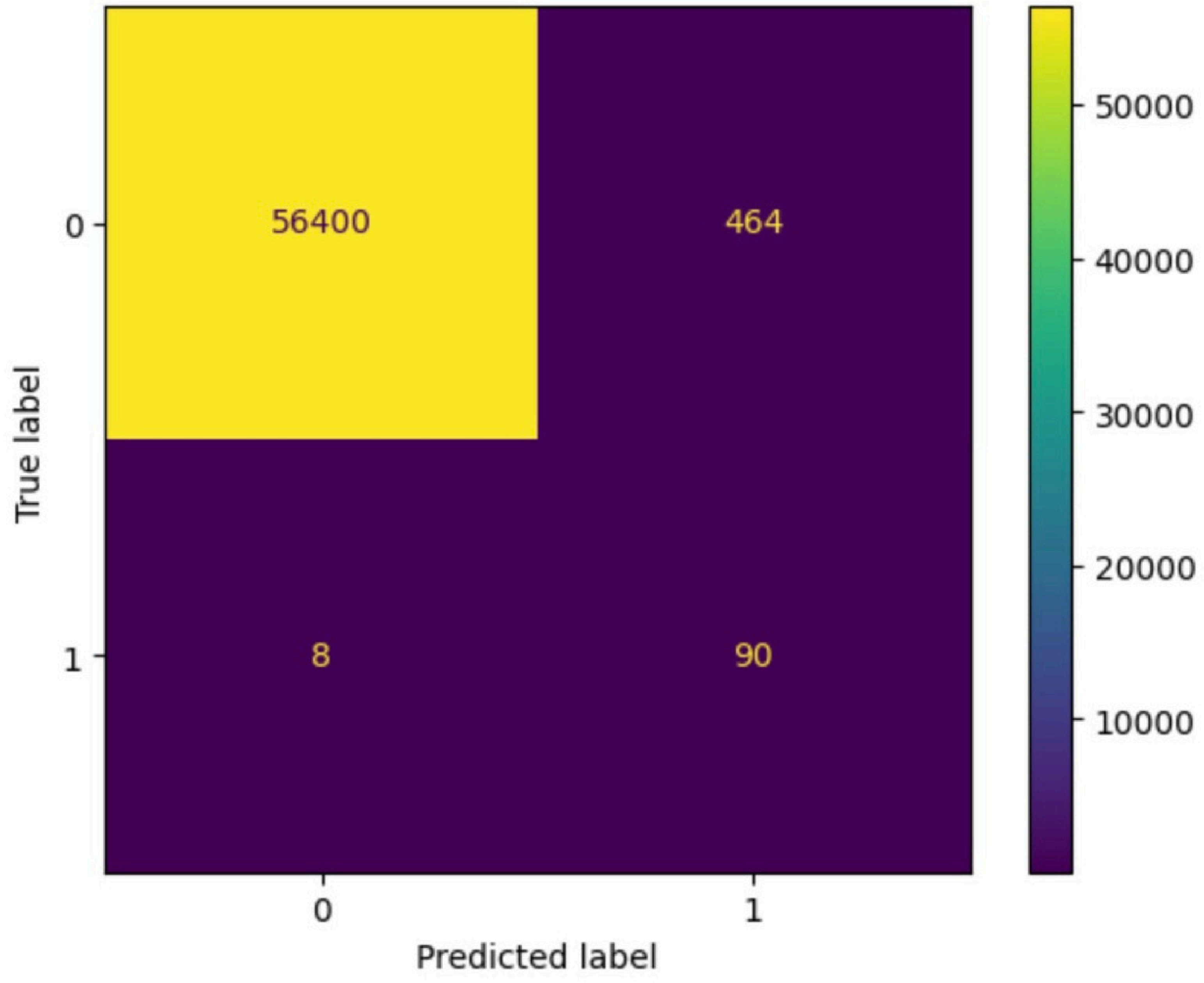
Model Selection: Training

We trained several different models on the data to find the best possible model:

- Logistic Regression
- Support Vector Machine (RBF kernel) with balanced class weight.
- K-Nearest Neighbors with 5 neighbors.
- Decision Trees
  - XGboost: A regularized collection of several decision trees
  - Random Forest: A simple combination of multiple decision trees (100 estimators).

Performance Metrics

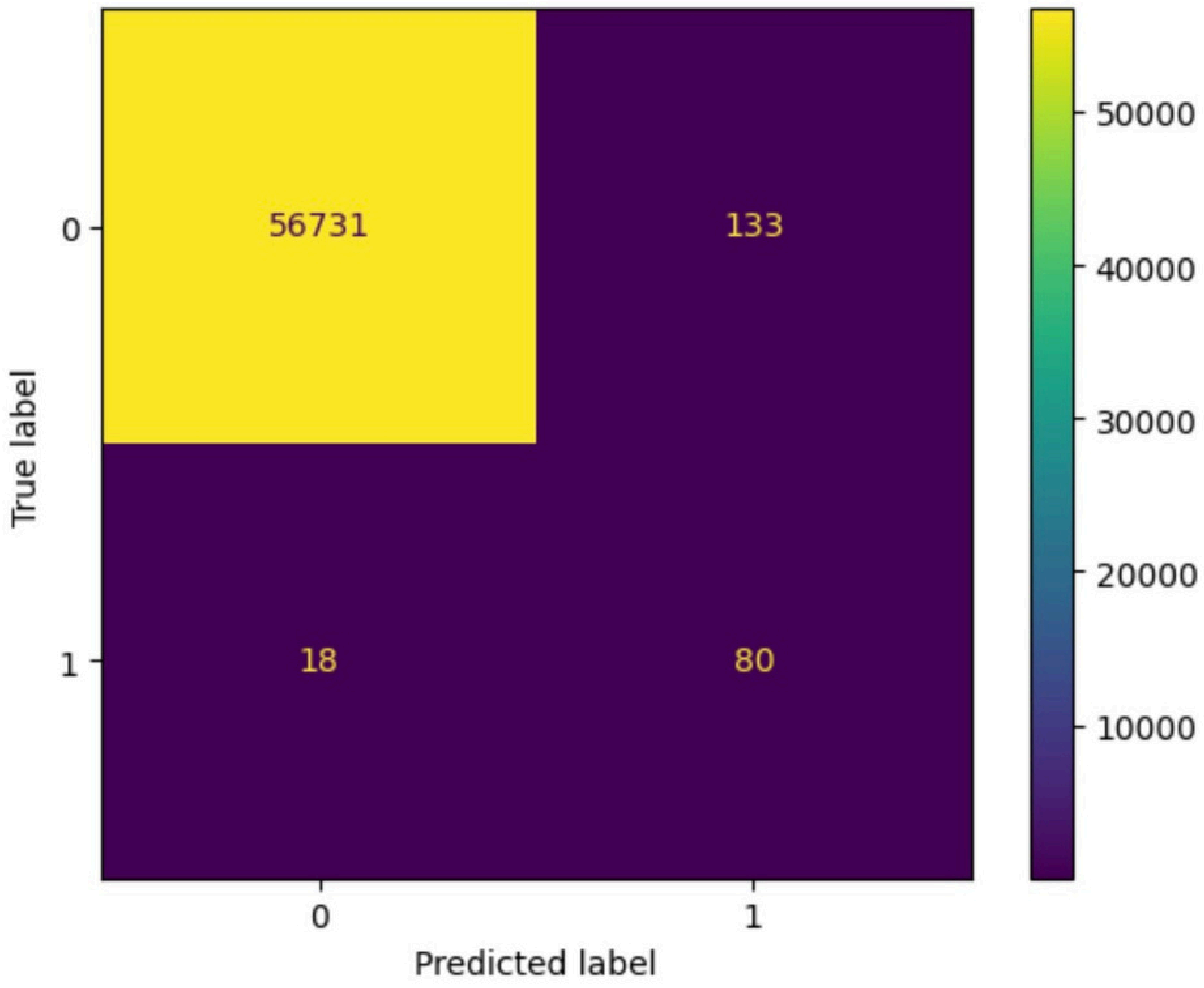
Logistic Regression:



accuracy	precision	recall	f1
0.992	0.58	0.96	0.64

Performance Metrics

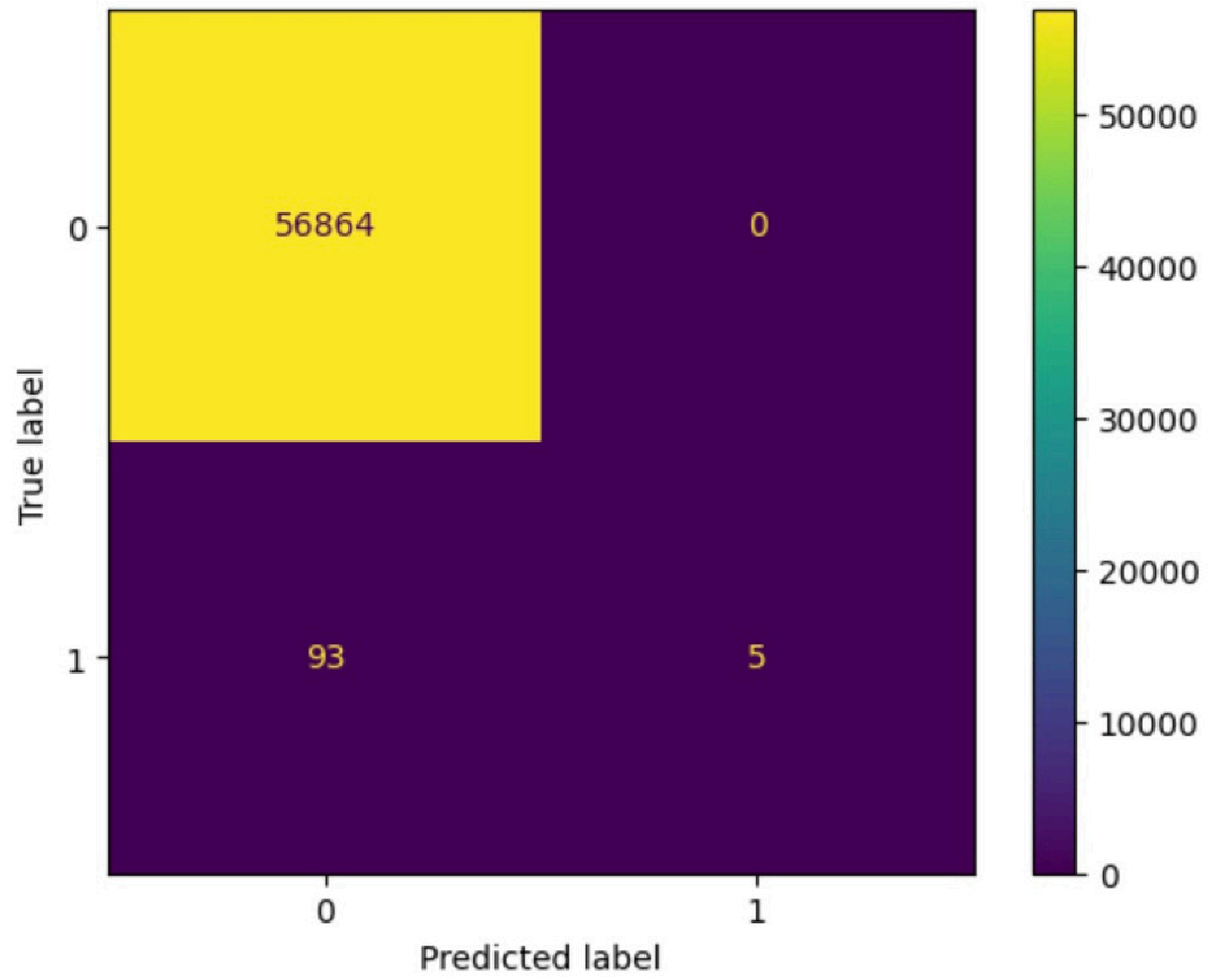
Support Vector Machine:



accuracy	precision	recall	f1
0.997	0.69	0.91	0.76

Performance Metrics

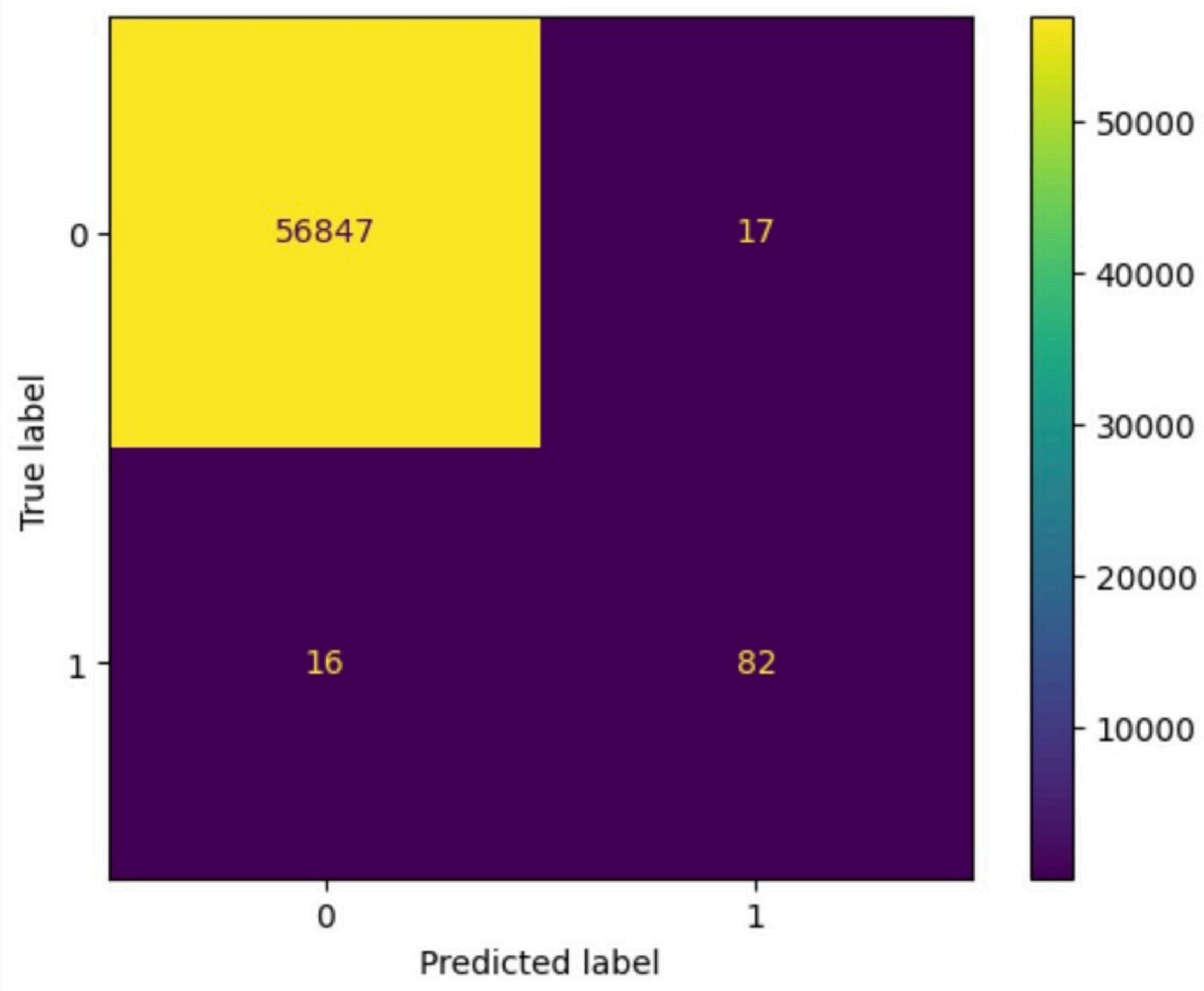
K-Nearest Neighbors:



accuracy	precision	recall	f1
0.998	1.00	0.53	0.55

Performance Metrics

XGBoost:

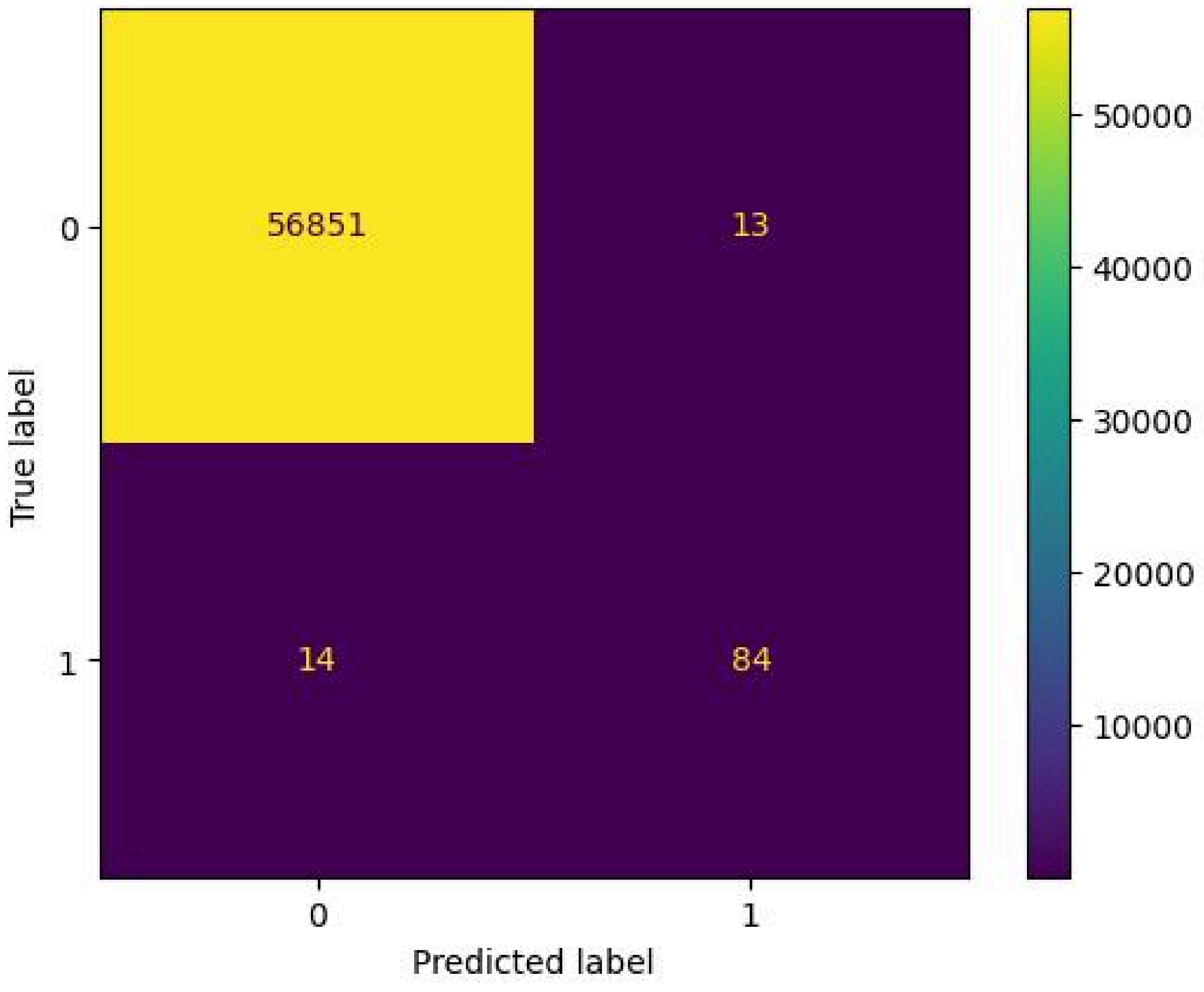


accuracy	precision	recall	f1
0.994	0.91	0.92	0.92

→

Performance Metrics

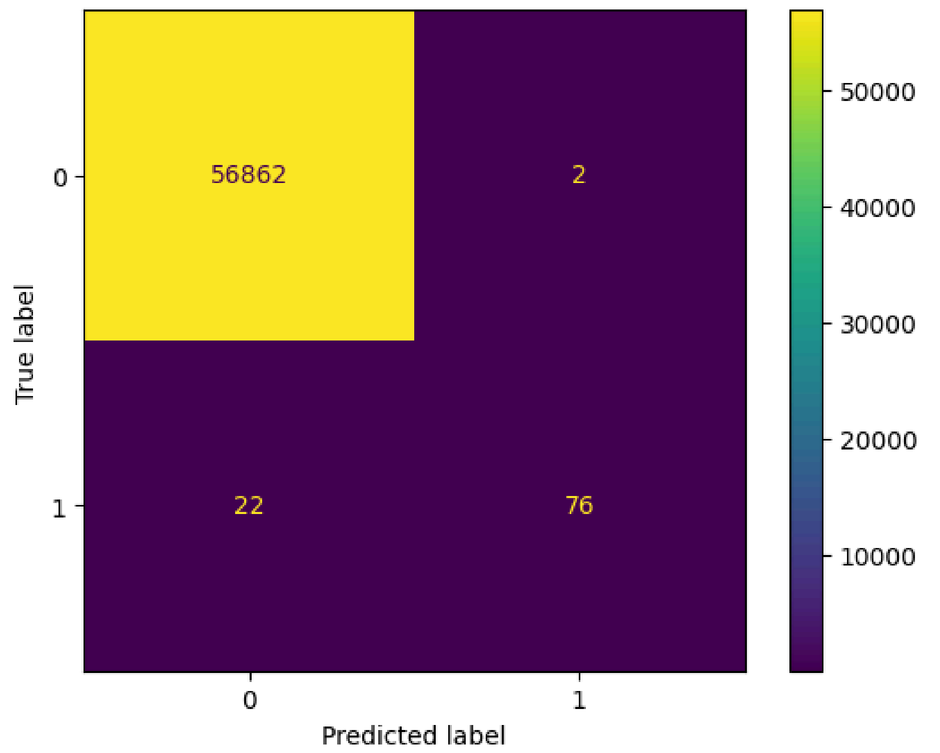
Random Forest:



accuracy	precision	recall	f1
0.999	0.94	0.93	0.93

Random Forest Before & After SMOTE

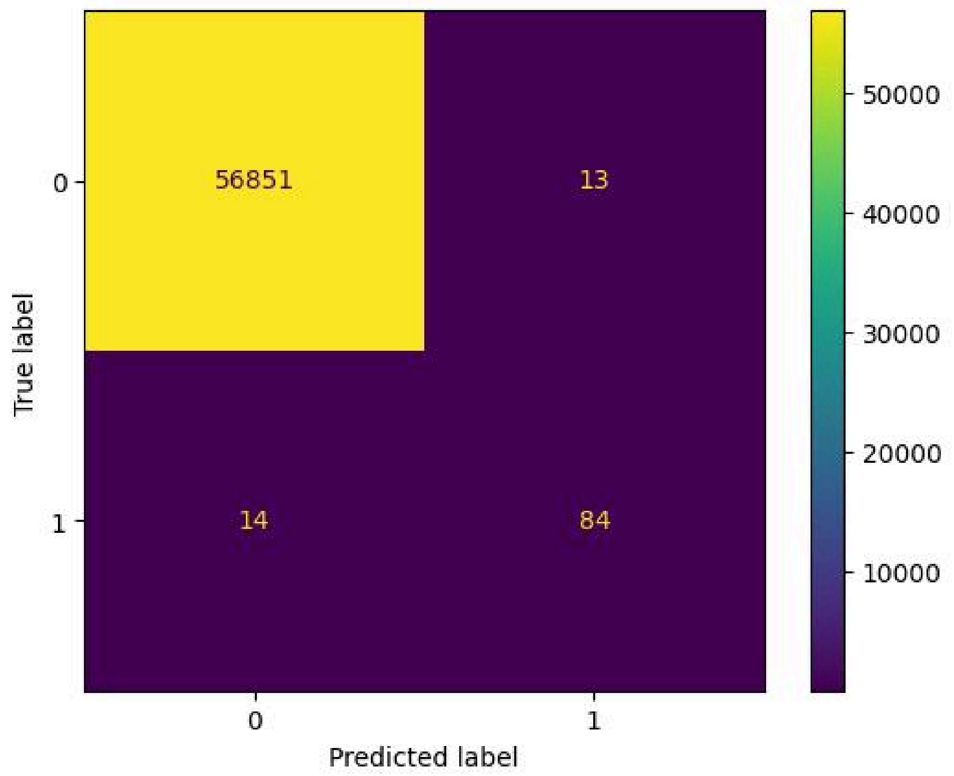
Before SMOTE



- True Negatives (TN) = 56862 → Correctly predicted normal transactions.
- False Positives (FP) = 2 → Normal predicted as fraud (false alarm).
- False Negatives (FN) = 22 → Missed fraud cases.
- True Positives (TP) = 76 → Correctly detected fraud.

accuracy	precision	recall	f1
0.999	0.99	0.89	0.93

After SMOTE



Improvement:

- Better at catching fraud (TP ↑ from 76 to 84).
- Misses fewer frauds (FN ↓ from 22 to 14).
- Slightly more false alarms (FP ↑ from 2 to 13), but worth it for higher fraud detection.

accuracy	precision	recall	f1
0.999	0.94	0.93	0.93

→

Conclusion

- According to every classification metric we used, the Random Forest Model was the best at predicting credit card fraud.

- Ranking:
    a. Random Forest Model
    b. XGBoost Model
    c. Support Vector Machine
    d. Logistic Regression
    e. K-Nearest Neighbors

- Both of the Decision Tree models were very successful on the test partition and had very low False Negative counts, which is critical in classifying fraud correctly.

Thank You

Any Questions?