ASTANA IT UNIVERSITY

Course: AI Essentials: Theory and Applications

Assignment title: Identification of credit card fraud cases through supervised and unsupervised learning models

Author: Nurlybay Bexultan
Group: AAI-2502

1. Introduction

Credit card fraud detection is a critical application of machine learning due to the increasing volume of digital transactions and the substantial financial losses caused by fraudulent activity. In this project, a combination of supervised and unsupervised learning techniques is applied to the widely used Credit Card Fraud Detection dataset provided by ULB on Kaggle. The dataset consists of real transaction records transformed with PCA for privacy, containing highly imbalanced binary labels where fraudulent cases represent less than 0.2% of all observations. This imbalance introduces challenges that require careful data preprocessing and resampling strategies.

The goal of the project is to evaluate how different machine learning approaches, specifically a supervised model (XGBoost) and an unsupervised anomaly detection model (Isolation Forest), perform on the same fraud-detection task. The workflow includes dataset preparation, exploratory analysis, feature selection using multiple techniques (SelectKBest, RFE, and PCA), and handling class imbalance using SMOTE. XGBoost is used as a robust classifier capable of capturing complex patterns in tabular data, while Isolation Forest represents a model better suited for unsupervised anomaly detection where fraudulent transactions can be treated as outliers. Finally, both models are compared using relevant evaluation metrics to assess their effectiveness and limitations in detecting rare fraudulent cases.

2. Data Preparation

The dataset used in this project is the Credit Card Fraud Detection dataset consisting of 284,807 anonymized transactions, each represented by 28 PCA-transformed features (V1–V28), along with the original Time and Amount attributes and a binary fraud label (Class). An initial inspection shows a severe class imbalance: only 492 transactions (0.1727%) are labeled as fraudulent, confirming that fraud detection constitutes an extreme rare-event classification task. The dataset contains no missing values, allowing preprocessing to focus on scaling and feature handling rather than imputation.

A descriptive analysis (figure 1) demonstrates that transaction amounts are heavily right-skewed, with most values clustered close to zero, while transaction times display a multimodal distribution corresponding to daily activity cycles. These observations are illustrated in the histograms of Amount and Time, confirming the need for robust scaling techniques to avoid distortion caused by outliers and long-tailed distributions.

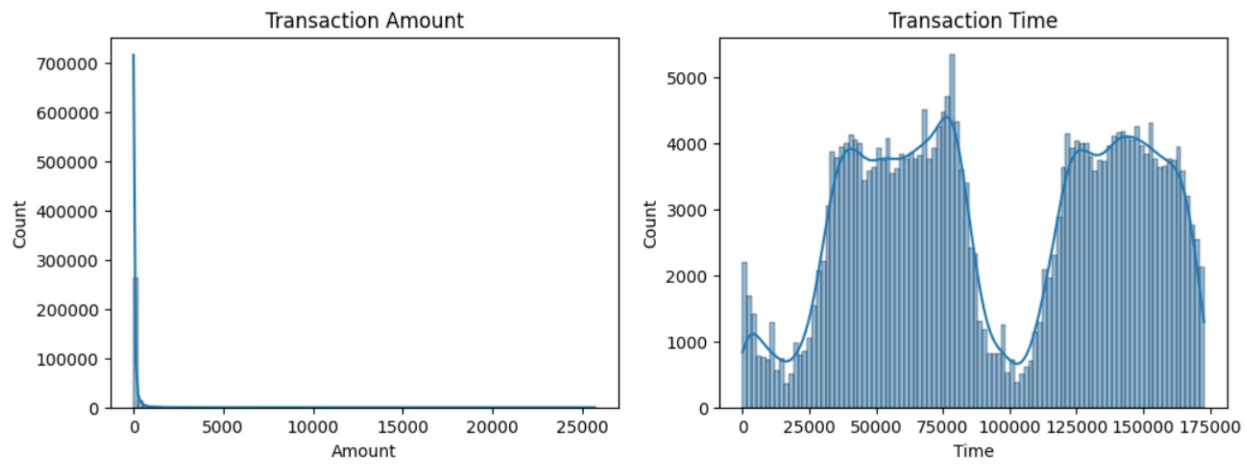
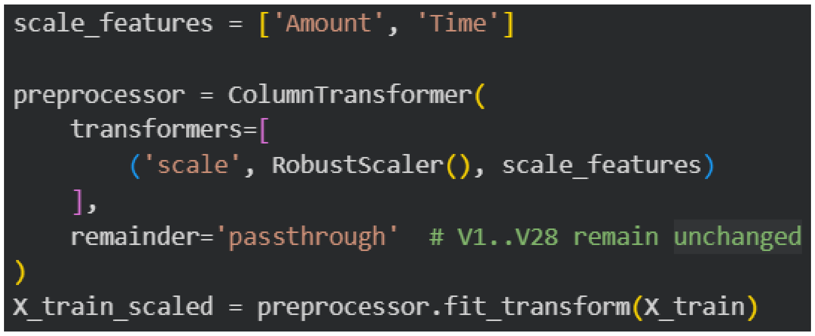


Figure 1

The data was split into training and testing sets using an 80/20 stratified split to preserve the original class proportions in both subsets. This ensures that the highly imbalanced nature of the dataset remains consistent across evaluation. To prepare features for model training, Amount and Time were scaled using RobustScaler (figure 2), chosen specifically for its resilience to outliers, while all PCA-transformed variables were left unchanged because they are already standardized. The resulting preprocessed feature matrix forms the foundation for subsequent feature selection, resampling, and model training steps.

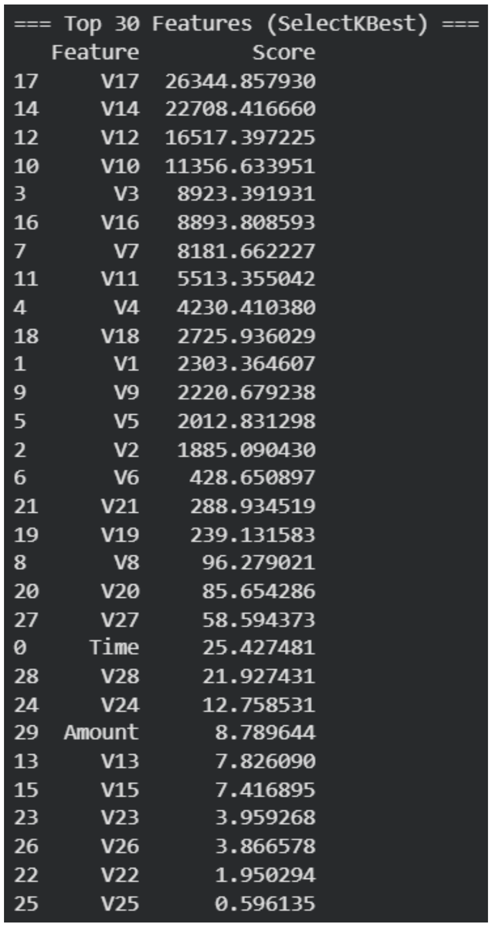


```
scale_features = ['Amount', 'Time']

preprocessor = ColumnTransformer(
    transformers=[
        ('scale', RobustScaler(), scale_features)
    ],
    remainder='passthrough'  # V1..V28 remain unchanged
)
X_train_scaled = preprocessor.fit_transform(X_train)
```

Figure 2

3. Feature Selection

To assess the importance of the available features and explore dimensionality reduction options, three complementary techniques were applied: SelectKBest, Recursive Feature Elimination (RFE), and Principal Component Analysis (PCA). These methods serve different purposes—statistical relevance, model-based selection, and variance-based dimensionality reduction—which provides a comprehensive view of feature significance.

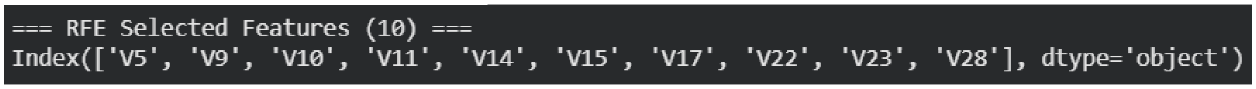


```
=== Top 30 Features (SelectKBest) ===
   Feature         Score
17     V17  26344.857930
14     V14  22708.416660
12     V12  16517.397225
10     V10  11356.633951
3       V3   8923.391931
16     V16   8893.808593
7       V7   8181.662227
11     V11   5513.355042
4       V4   4230.410380
18     V18   2725.936029
1       V1   2303.364607
9       V9   2220.679238
5       V5   2012.831298
2       V2   1885.090430
6       V6    428.650897
21     V21    288.934519
19     V19    239.131583
8       V8     96.279021
20     V20     85.654286
27     V27     58.594373
0     Time     25.427481
28     V28     21.927431
24     V24     12.758531
29  Amount      8.789644
13     V13      7.826090
15     V15      7.416895
23     V23      3.959268
26     V26      3.866578
22     V22      1.950294
25     V25      0.596135
```

Figure 3

SelectKBest (figure 3), using the ANOVA F-test, ranked all 30 features according to their discriminative power. The highest-scoring attributes were several PCA-transformed components such as V17, V14, V12, and V10, which showed exceptionally strong statistical separation between fraud and non-fraud classes. Even the lowest-ranked features maintained non-zero relevance scores, indicating that each contributes some degree of predictive information.

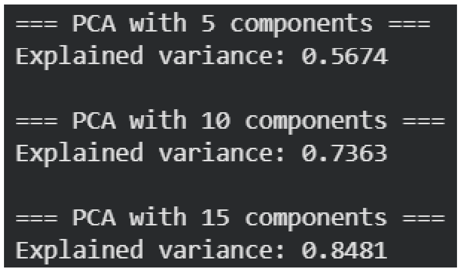
RFE (figure 4), using logistic regression as the estimator, reduced the feature space to 10 components by recursively removing the least important variables. This technique identified a compact subset, primarily concentrated around components such as V5, V9, V10, V11, V14, and V17.



```
=== RFE Selected Features (10) ===
Index(['V5', 'V9', 'V10', 'V11', 'V14', 'V15', 'V17', 'V22', 'V23', 'V28'], dtype='object')
```

Figure 4

PCA (figure 5) was evaluated to examine the potential for dimensionality reduction. However, retaining only 5, 10, or 15 components preserved just 56.7%, 73.6%, and 84.8% of the variance, respectively, which represents a substantial loss of information relative to the full feature set. Given that fraud detection is a rare-event problem, discarding variance may lead to losing subtle but critical patterns associated with fraudulent behavior.



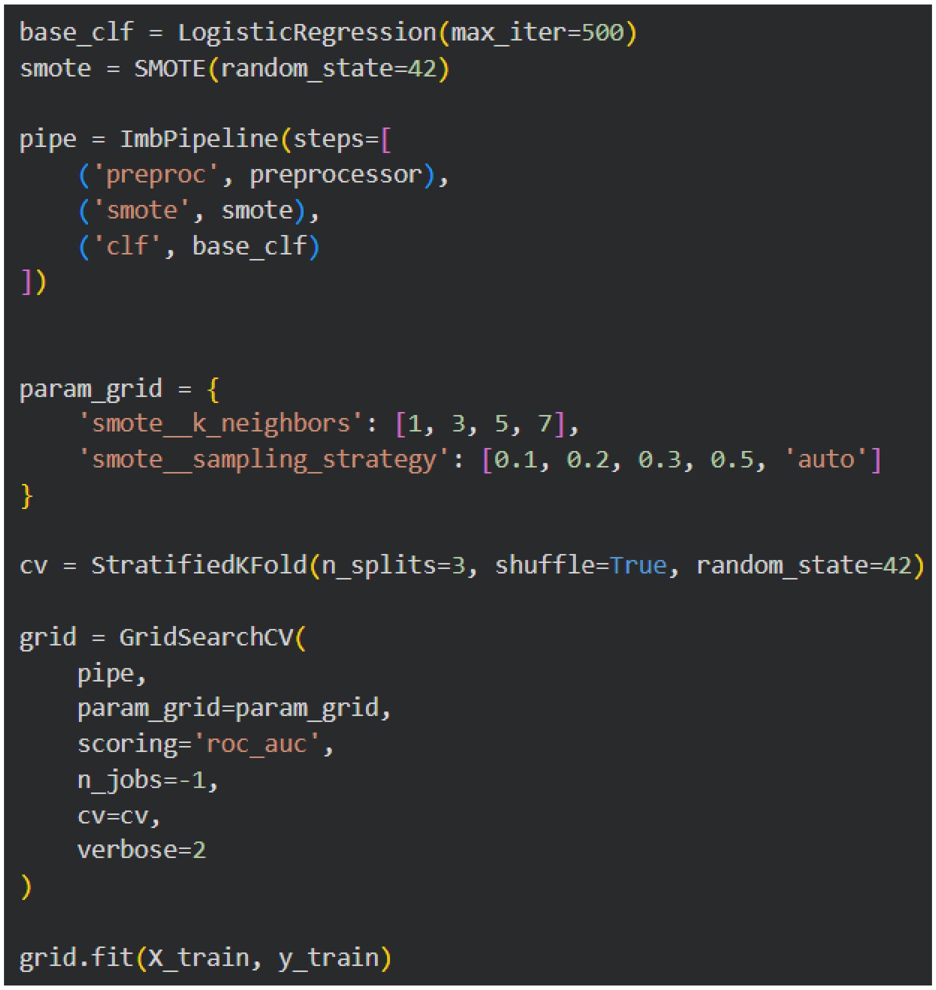
```
=== PCA with 5 components ===
Explained variance: 0.5674

=== PCA with 10 components ===
Explained variance: 0.7363

=== PCA with 15 components ===
Explained variance: 0.8481
```

Figure 5

Based on these findings and considering that many machine learning algorithms (including XGBoost and Isolation Forest) perform well with the full feature set, the decision was made to retain all 30 original features for subsequent modeling. This ensures that no potentially informative component is removed, preserving the dataset's full discriminatory capacity.

4. Handling Imbalanced Data

The dataset exhibits extreme class imbalance, with fraudulent transactions representing only 0.17% of all samples. Training a model directly on such skewed data would cause classifiers to favor the majority class, resulting in poor fraud detection performance. To address this issue, the Synthetic Minority Oversampling Technique (SMOTE) was used to generate synthetic fraud samples within the feature space of the minority class.



```
base_clf = LogisticRegression(max_iter=500)
smote = SMOTE(random_state=42)

pipe = ImbPipeline(steps=[
    ('preproc', preprocessor),
    ('smote', smote),
    ('clf', base_clf)
])


param_grid = {
    'smote__k_neighbors': [1, 3, 5, 7],
    'smote__sampling_strategy': [0.1, 0.2, 0.3, 0.5, 'auto']
}

cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)

grid = GridSearchCV(
    pipe,
    param_grid=param_grid,
    scoring='roc_auc',
    n_jobs=-1,
    cv=cv,
    verbose=2
)

grid.fit(X_train, y_train)
```

Figure 6

To ensure that oversampling was applied optimally, SMOTE parameters were tuned through a pipeline that included preprocessing, SMOTE, and a logistic regression baseline classifier (figure 6). Using a grid search with stratified cross-validation, the hyperparameters k_neighbors and sampling_strategy were evaluated according to ROC-AUC performance. The best configuration was identified as k_neighbors = 3 and sampling_strategy = 0.1, meaning that the minority class was expanded to 10% of the majority class size. This moderate oversampling level improved separability without excessively altering the original data distribution, achieving a cross-validated ROC-AUC of 0.9763, which confirmed the effectiveness of the chosen parameters.

To visualize the impact of SMOTE on the data structure, UMAP was applied to project high-dimensional features into a 2D space. Before SMOTE, minority samples appeared isolated and scattered across the embedding, reflecting their sparse density within the original distribution. After applying SMOTE, synthetic fraud samples filled the local minority neighborhoods, forming more coherent clusters while preserving the global structure of majority samples. This demonstrates that SMOTE successfully increases the representation of fraudulent patterns in the feature space, providing supervised learning models with a richer set of informative minority-class examples.

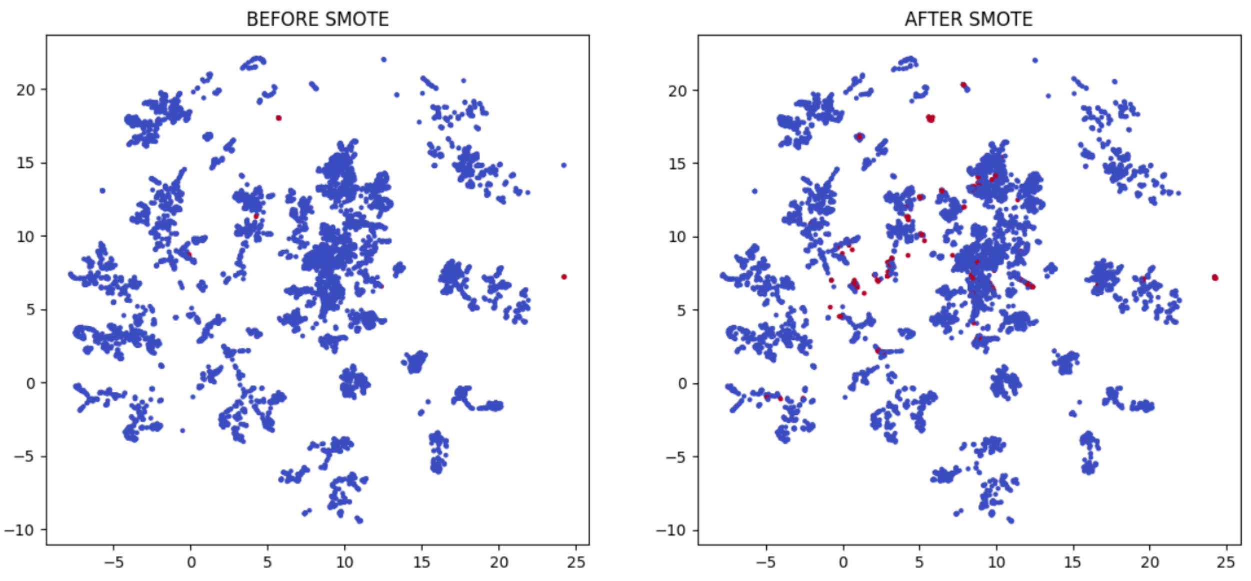
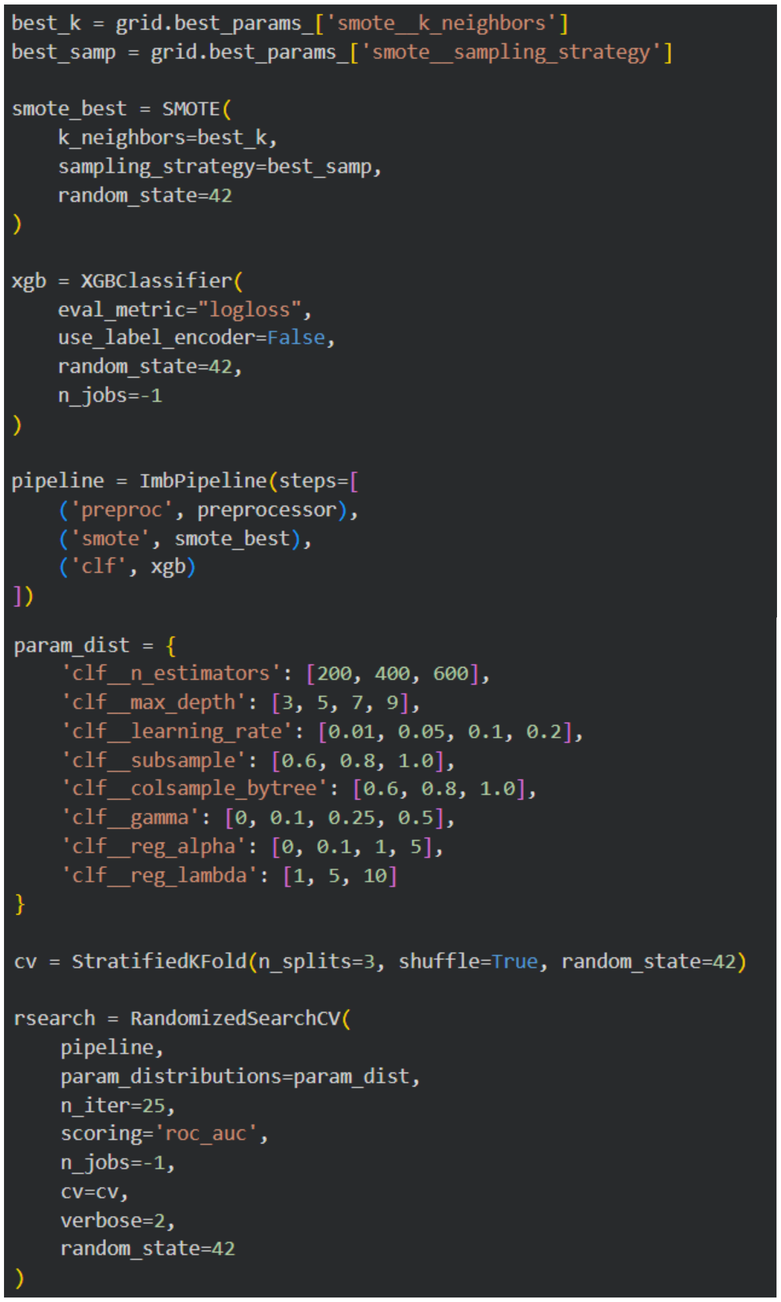


Figure 7

This enhanced training distribution is particularly important for models like XGBoost, which rely on sufficient class representation to learn discriminative patterns, and it forms the basis for the supervised experiments in subsequent sections.

5. Supervised Learning Model: XGBoost

To model the fraud detection problem under supervised learning, the XGBoost (Extreme Gradient Boosting) algorithm was selected due to its strong performance on tabular data, ability to model nonlinear relationships, and robustness to heterogeneous feature distributions. XGBoost is particularly well-suited for fraud detection because it can capture complex interactions between features while maintaining strong generalization through regularization and tree-based ensemble methods.



```
best_k = grid.best_params_['smote__k_neighbors']
best_samp = grid.best_params_['smote__sampling_strategy']

smote_best = SMOTE(
    k_neighbors=best_k,
    sampling_strategy=best_samp,
    random_state=42
)

xgb = XGBClassifier(
    eval_metric="logloss",
    use_label_encoder=False,
    random_state=42,
    n_jobs=-1
)

pipeline = ImbPipeline(steps=[
    ('preproc', preprocessor),
    ('smote', smote_best),
    ('clf', xgb)
])

param_dist = {
    'clf__n_estimators': [200, 400, 600],
    'clf__max_depth': [3, 5, 7, 9],
    'clf__learning_rate': [0.01, 0.05, 0.1, 0.2],
    'clf__subsample': [0.6, 0.8, 1.0],
    'clf__colsample_bytree': [0.6, 0.8, 1.0],
    'clf__gamma': [0, 0.1, 0.25, 0.5],
    'clf__reg_alpha': [0, 0.1, 1, 5],
    'clf__reg_lambda': [1, 5, 10]
}

cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)

rsearch = RandomizedSearchCV(
    pipeline,
    param_distributions=param_dist,
    n_iter=25,
    scoring='roc_auc',
    n_jobs=-1,
    cv=cv,
    verbose=2,
    random_state=42
)
```

Figure 8

The model was incorporated into a pipeline alongside preprocessing and the optimized SMOTE configuration (k_neighbors = 3, sampling_strategy = 0.1). To maximize predictive performance, an extensive hyperparameter search was performed using RandomizedSearchCV, evaluating configurations over tree depth, learning rate, number of estimators, sampling ratios, and regularization parameters. The search was conducted with stratified 3-fold cross-validation and ROC-AUC as the scoring metric, reflecting the importance of ranking and minority-class detection in imbalanced settings.

The optimization process identified an effective parameter combination, including max_depth = 5, a conservative learning_rate = 0.01, and n_estimators = 200, alongside subsampling and column-sampling ratios that help reduce overfitting. Regularization parameters (reg_alpha = 1, reg_lambda = 1) and a moderate gamma = 0.5 further contributed to controlling model complexity. With these settings, the tuned XGBoost model achieved a cross-validated ROC-AUC of 0.9854, indicating excellent separation between fraudulent and legitimate transactions.

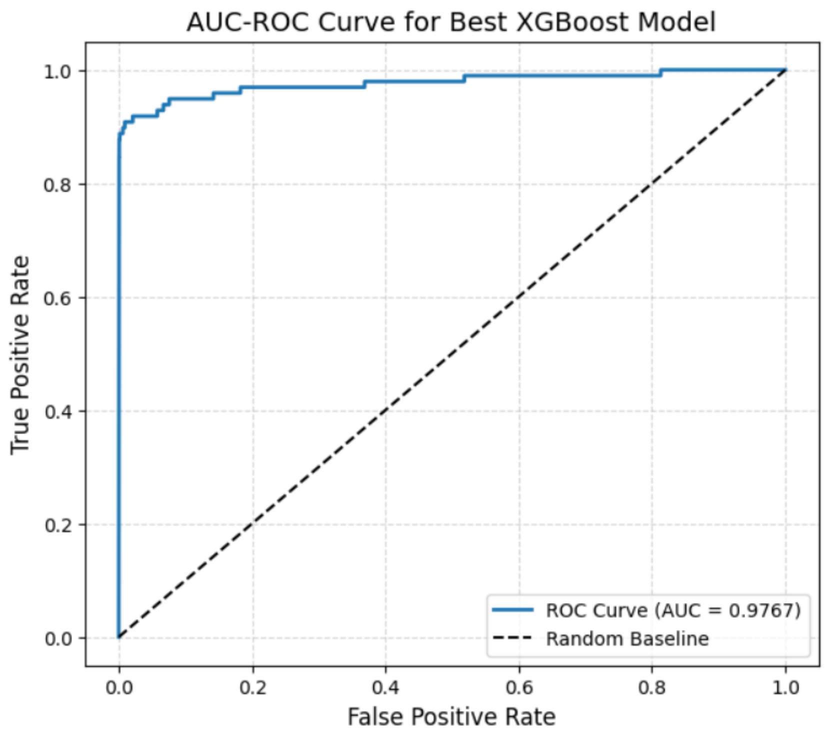


Figure 9

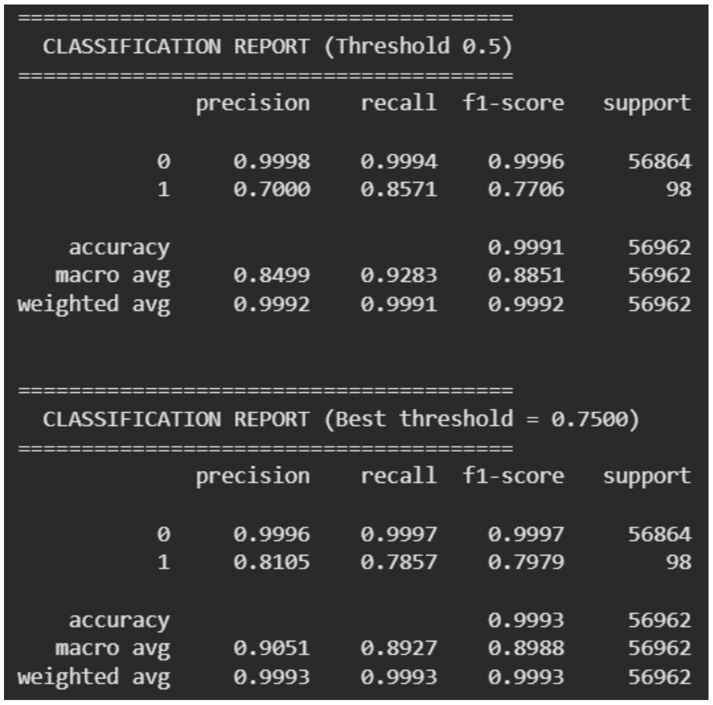
This strong performance demonstrates the capability of XGBoost, especially when combined with targeted resampling through SMOTE, to effectively learn patterns associated with rare fraudulent behavior. The resulting model serves as the main supervised benchmark for comparison against the unsupervised anomaly detection approach.

6. Threshold Optimization for XGBoost

Although XGBoost produces probability estimates for each transaction, classification performance, especially on imbalanced datasets, depends heavily on the decision threshold used to convert probabilities into class labels. The default threshold of 0.5 often does not maximize the metrics that are most relevant for fraud detection, such as recall, precision, or F1-score. To evaluate the effect of threshold choice, a range of values from 0.01 to 0.99 was tested on the model's predicted probabilities for the test set.

Across thresholds, the best F1-score was obtained at t = 0.75 (F1 = 0.798), the best precision at t = 0.82 (0.849), and the best recall at extremely low thresholds (t = 0.01) where the model flagged all fraud cases but also introduced many false positives. These trade-offs highlight the inherent tension between maximizing fraud detection (recall) and minimizing false alarms (precision).

Despite the F1-optimal value at 0.75, the threshold of 0.5 remained a competitive choice because it maintained higher recall, a critical property in fraud detection tasks where missing a fraudulent transaction is significantly more costly than investigating a legitimate one. At t = 0.5, the recall was 0.857, compared to 0.786 at the F1-optimal threshold, while still preserving strong overall accuracy. Precision, recall, and F1 differences between the two thresholds were found to be modest, but the increase in false negatives at t = 0.75 made the default threshold preferable.



```
========================================
  CLASSIFICATION REPORT (Threshold 0.5)
========================================
              precision    recall  f1-score   support

           0     0.9998    0.9994    0.9996     56864
           1     0.7000    0.8571    0.7706        98

    accuracy                         0.9991     56962
   macro avg     0.8499    0.9283    0.8851     56962
weighted avg     0.9992    0.9991    0.9992     56962


========================================
  CLASSIFICATION REPORT (Best threshold = 0.7500)
========================================
              precision    recall  f1-score   support

           0     0.9996    0.9997    0.9997     56864
           1     0.8105    0.7857    0.7979        98

    accuracy                         0.9993     56962
   macro avg     0.9051    0.8927    0.8988     56962
weighted avg     0.9993    0.9993    0.9993     56962
```

Figure 10

Confusion matrices (figure 11) confirmed this behavior. At t = 0.5, the model misclassified only 14 fraudulent cases, whereas t = 0.75 increased false negatives to 21. Since recall is a priority in fraud prevention domains, the threshold of 0.5 was selected for the final supervised model evaluation.

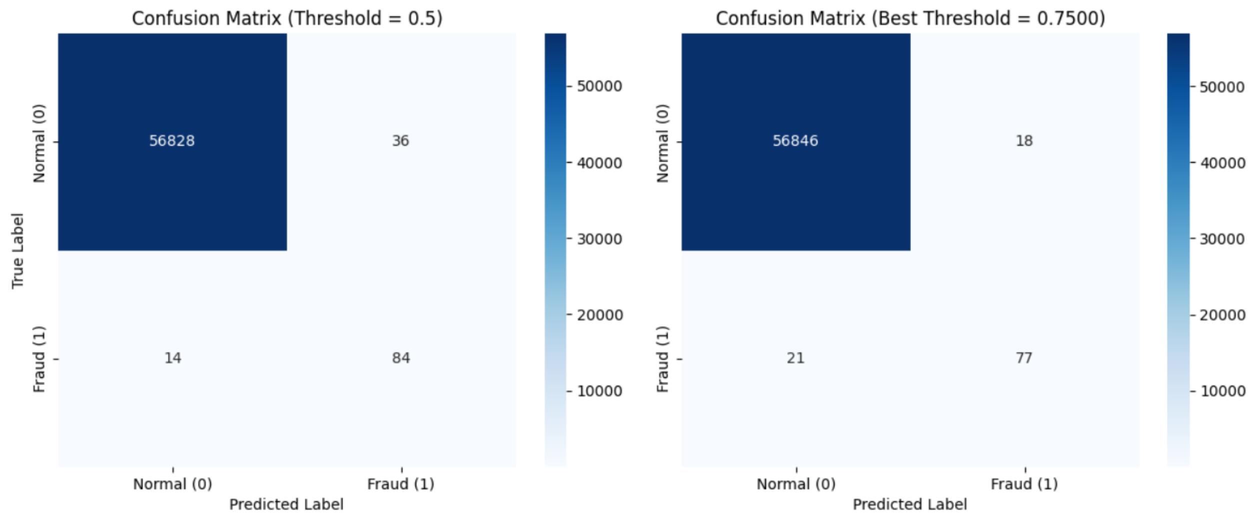


Figure 11

This threshold-tuned XGBoost model provides a strong baseline against which the unsupervised approach (Isolation Forest) can be compared.

7. Unsupervised Anomaly Detection: Isolation Forest

To complement the supervised XGBoost model, the Isolation Forest algorithm was employed as an unsupervised anomaly detection approach. Isolation Forest identifies anomalies by recursively partitioning the feature space; fraudulent transactions, being rare and different from normal patterns, require fewer splits to isolate and therefore receive lower anomaly scores. This method is appealing in fraud detection scenarios where labeled data may be limited or imbalanced.

The model was trained on the scaled feature set without SMOTE, as oversampling is not applicable in unsupervised anomaly detection. To identify an effective configuration, an extensive grid search was conducted over key hyperparameters including the number of trees (n_estimators), subsampling size (max_samples), contamination level (expected proportion of anomalies), and feature sampling ratio (max_features). Each configuration was evaluated on the test set using ROC-AUC computed from binary anomaly labels.

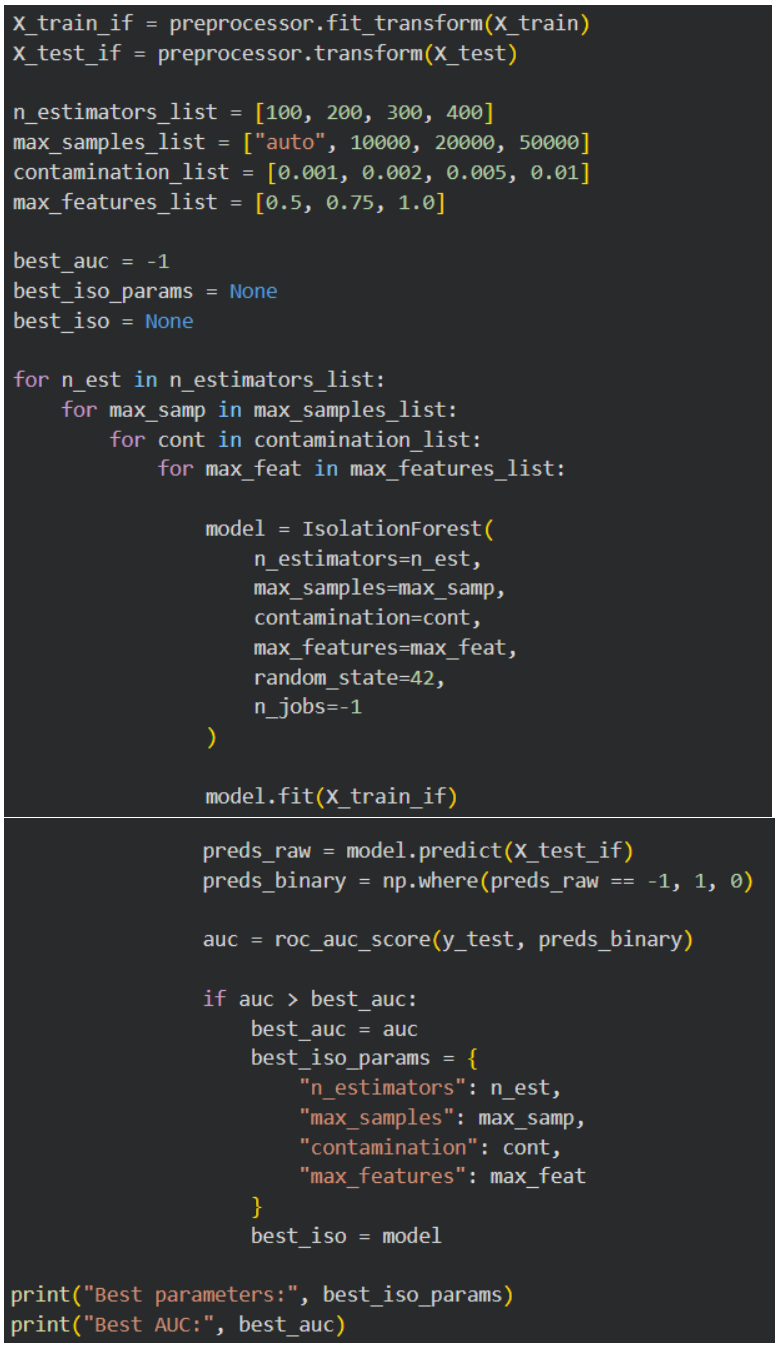

```python
X_train_if = preprocessor.fit_transform(X_train)
X_test_if = preprocessor.transform(X_test)

n_estimators_list = [100, 200, 300, 400]
max_samples_list = ["auto", 10000, 20000, 50000]
contamination_list = [0.001, 0.002, 0.005, 0.01]
max_features_list = [0.5, 0.75, 1.0]

best_auc = -1
best_iso_params = None
best_iso = None

for n_est in n_estimators_list:
    for max_samp in max_samples_list:
        for cont in contamination_list:
            for max_feat in max_features_list:

                model = IsolationForest(
                    n_estimators=n_est,
                    max_samples=max_samp,
                    contamination=cont,
                    max_features=max_feat,
                    random_state=42,
                    n_jobs=-1
                )

                model.fit(X_train_if)

                preds_raw = model.predict(X_test_if)
                preds_binary = np.where(preds_raw == -1, 1, 0)

                auc = roc_auc_score(y_test, preds_binary)

                if auc > best_auc:
                    best_auc = auc
                    best_iso_params = {
                        "n_estimators": n_est,
                        "max_samples": max_samp,
                        "contamination": cont,
                        "max_features": max_feat
                    }
                    best_iso = model

print("Best parameters:", best_iso_params)
print("Best AUC:", best_auc)
```

Figure 12

The best-performing model used 100 estimators, 50,000 subsampling samples, 1% contamination, and 0.5 max_features, producing a classification-style AUC of 0.8729. While substantially lower than the supervised XGBoost performance, this is typical for unsupervised methods given that they lack label information during training.

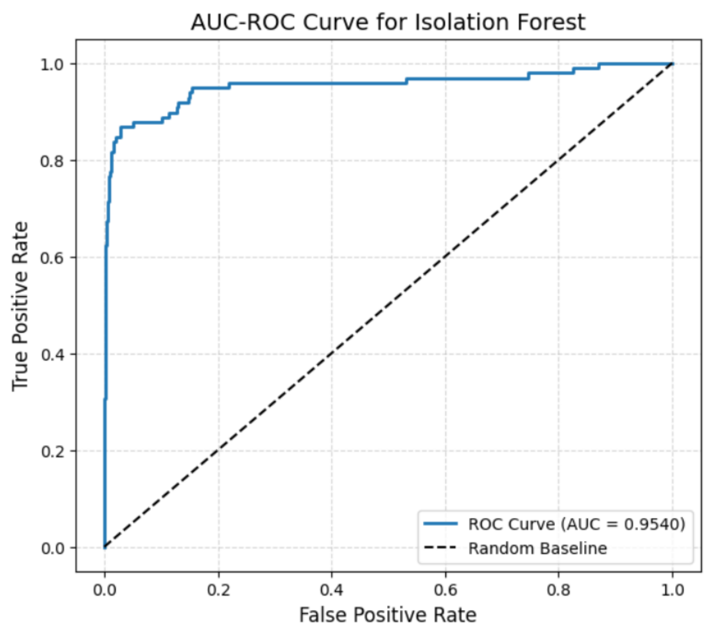


Figure 13

To obtain a more faithful evaluation, the model's continuous anomaly scores (via decision_function) were converted into fraud likelihoods, and a proper ROC curve was computed. This yielded a significantly higher ROC-AUC of 0.9540, indicating that while the model’s decision threshold is imperfect, its ranking ability remains strong. The ROC curve shows that Isolation Forest can effectively distinguish normal and fraudulent patterns, though its optimal operating threshold may differ from the default.

Overall, Isolation Forest provides a reasonable unsupervised baseline and demonstrates the ability to identify anomalous behavior without labeled data. However, its performance remains below the tuned supervised XGBoost model, highlighting the advantage of incorporating ground-truth labels in fraud detection tasks.

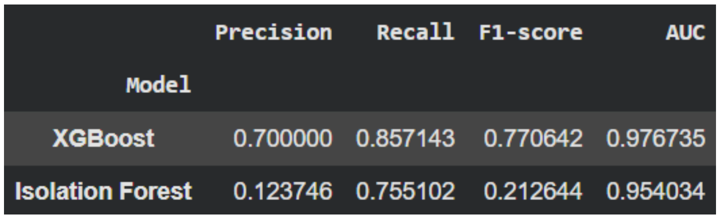
8. Model Comparison

Both XGBoost and Isolation Forest were evaluated on the same test set to compare supervised and unsupervised approaches for fraud detection. The results clearly highlight the advantage of supervised learning when labeled data is available.

The XGBoost model achieved substantially higher performance across all metrics, with a precision of 0.70, recall of 0.857, F1-score of 0.771, and ROC-AUC of 0.9767. The confusion matrix shows that XGBoost correctly identified 84 out of 98 fraud cases, misclassifying only 14, while maintaining an extremely low false-positive rate (only 36 legitimate transactions flagged incorrectly). This demonstrates its strong ability to detect fraudulent activity with minimal disruption to normal transactions.

In contrast, the Isolation Forest model, despite achieving a strong ROC-AUC of 0.9540, suffered from significantly lower classification effectiveness when using a thresholded prediction. It attained a precision of only 0.124 and an F1-score of 0.213, reflecting a high number of false positives. The confusion matrix indicates that Isolation Forest incorrectly flagged 524 legitimate transactions, greatly reducing precision, even though it captured 74 of the fraud cases. This outcome is expected for unsupervised anomaly detection, which must infer fraud patterns without labels and often over-identifies outliers to maximize recall.

Table 1



	Precision	Recall	F1-score	AUC
Model				
XGBoost	0.700000	0.857143	0.770642	0.976735
Isolation Forest	0.123746	0.755102	0.212644	0.954034

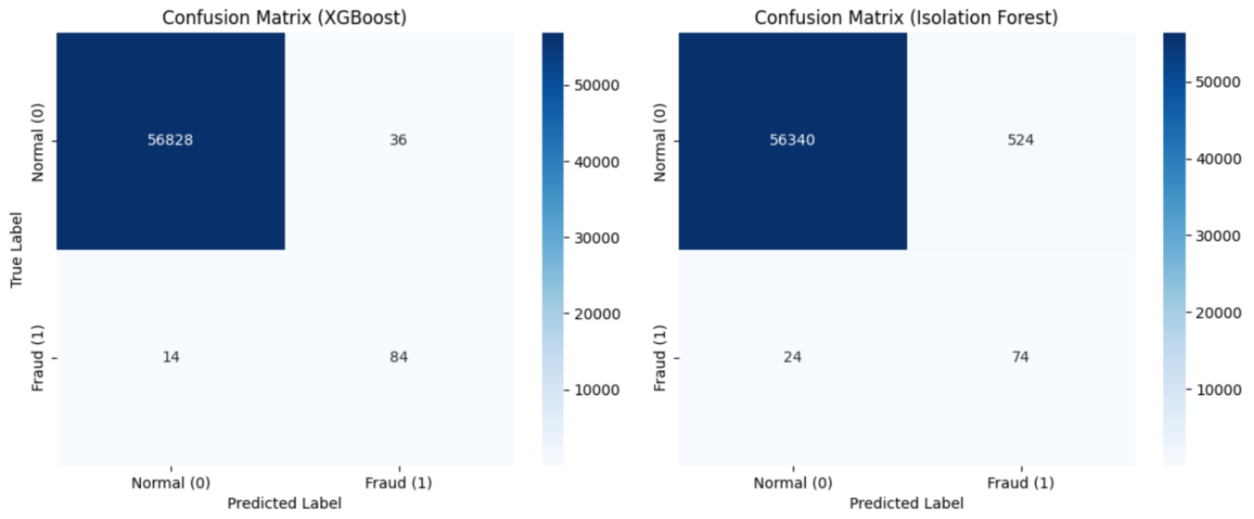


Figure 14

Overall, while Isolation Forest demonstrates good ranking ability through ROC-AUC, its practical classification performance is far inferior to XGBoost. The supervised model benefits from labeled fraud examples and the SMOTE-enhanced training distribution, enabling it to learn more precise and discriminative decision boundaries. These results show that supervised learning is clearly superior in this dataset, provided that even a small number of labeled fraud instances are available.