Financial Fraud Detection Using Supervised and Unsupervised Machine Learning Models

Name: B.Lokesh

Id: BT23CSE047

Analysis Report:

Introduction

Financial fraud is a critical challenge in the modern digital economy, leading to significant monetary losses and eroding customer trust. Traditional rule-based systems often struggle to keep up with evolving fraud techniques. Machine Learning (ML) offers a promising approach by identifying patterns and anomalies in financial data that may indicate fraudulent behavior.

This report presents an in-depth comparative analysis of both **supervised** and **unsupervised** ML models for financial fraud detection using key evaluation metrics. The goal is to assess each model's effectiveness in identifying fraudulent transactions, especially in the context of **highly imbalanced datasets** where fraud cases are rare but highly impactful.

Models Evaluated

Supervised Learning Models

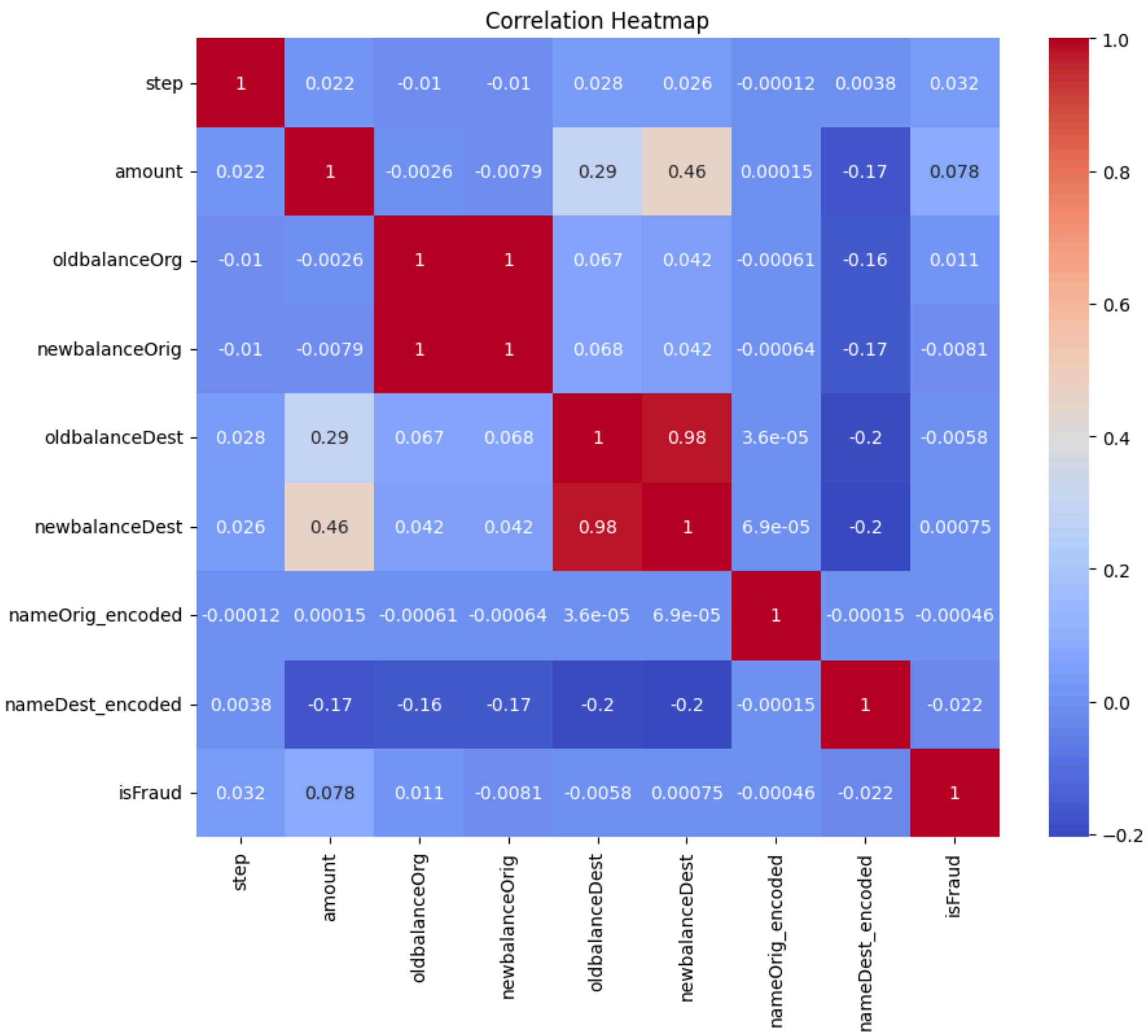
1. **Logistic Regression**
2. **Random Forest**

3. **XGBoost (Extreme Gradient Boosting)**

Unsupervised Learning Models

4. **Isolation Forest**
5. **One-Class SVM**

EDA:



Correlation Heatmap Analysis

The heatmap shown above visualizes the **correlation matrix** for various numerical features in the dataset. Each cell in the heatmap represents the Pearson correlation coefficient between two features, where:

- **+1** indicates a perfect positive correlation,
- **-1** indicates a perfect negative correlation,
- **0** indicates no correlation.

The color gradient (ranging from blue to red) helps easily identify strong and weak correlations:

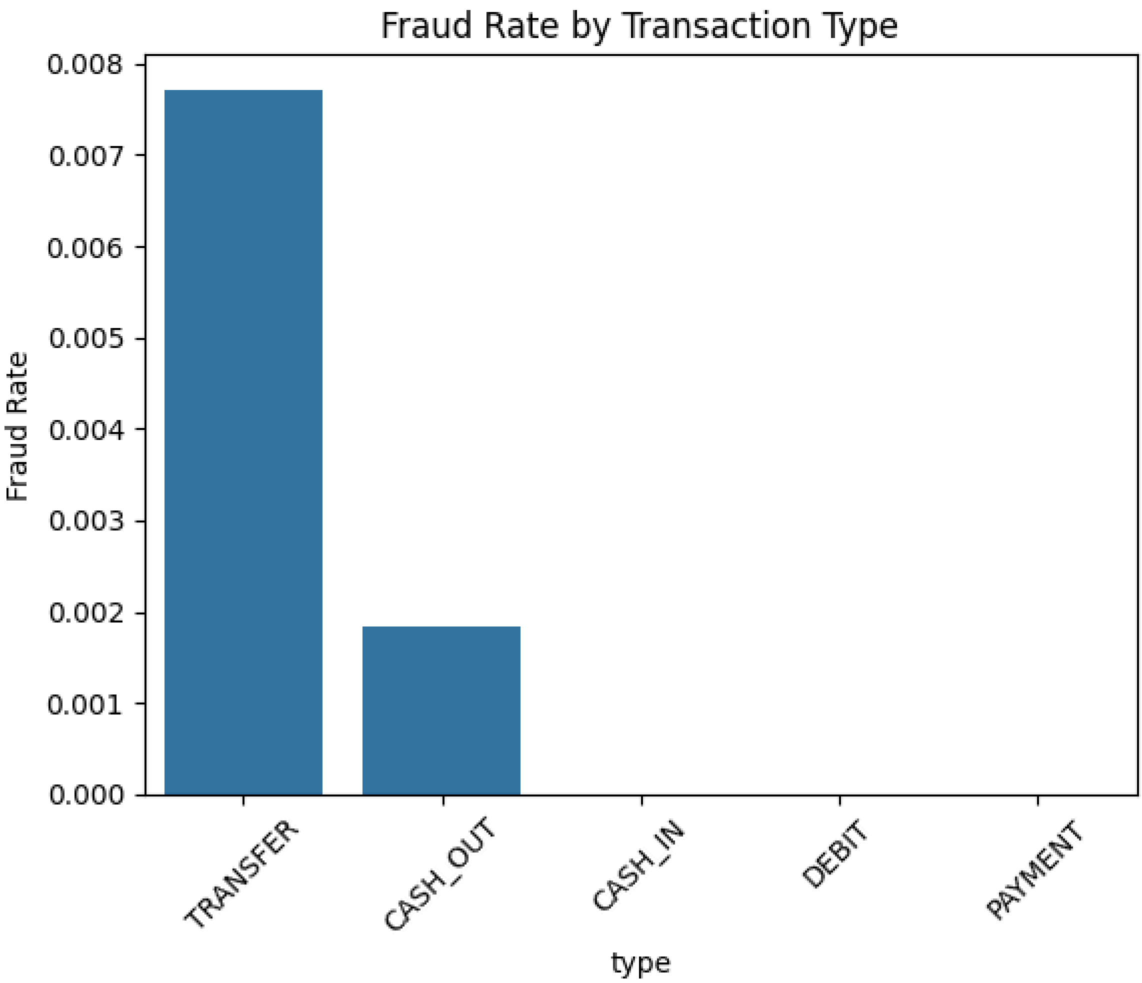
- **Redder areas** indicate stronger positive correlations.
- **Bluer areas** indicate stronger negative correlations.
- **White/light areas** represent near-zero correlation.

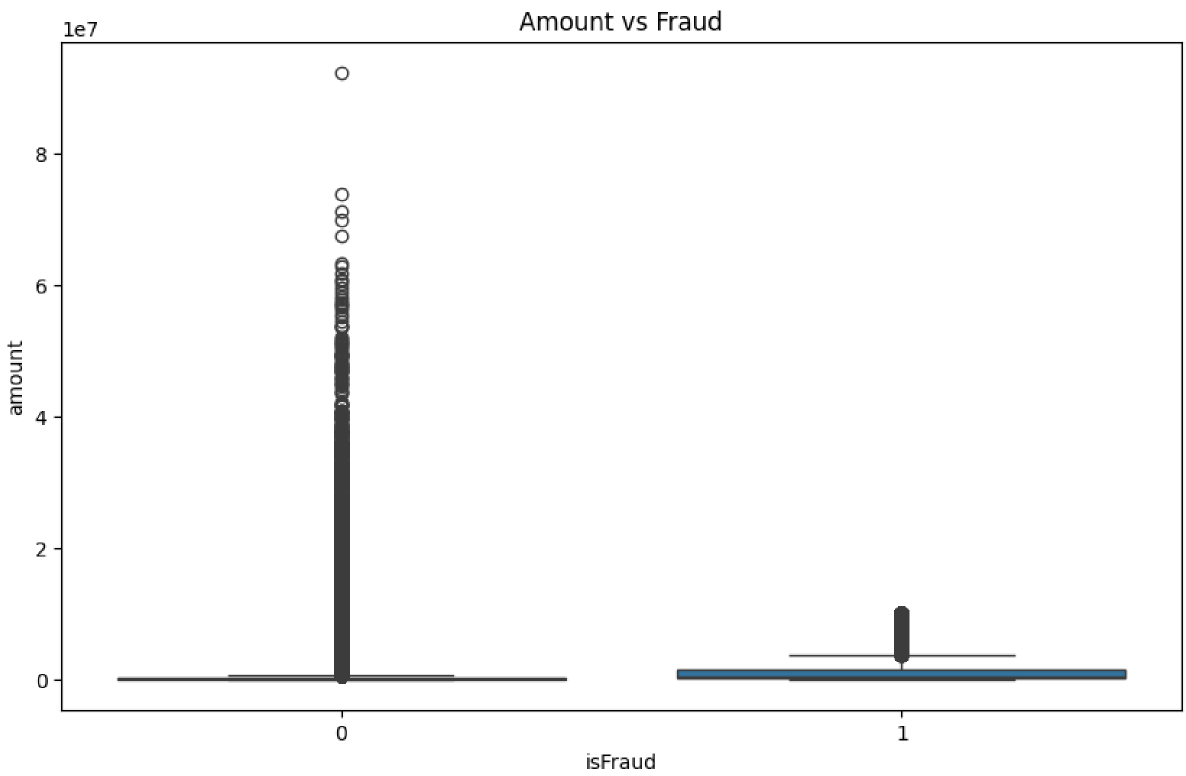
Key Observations:

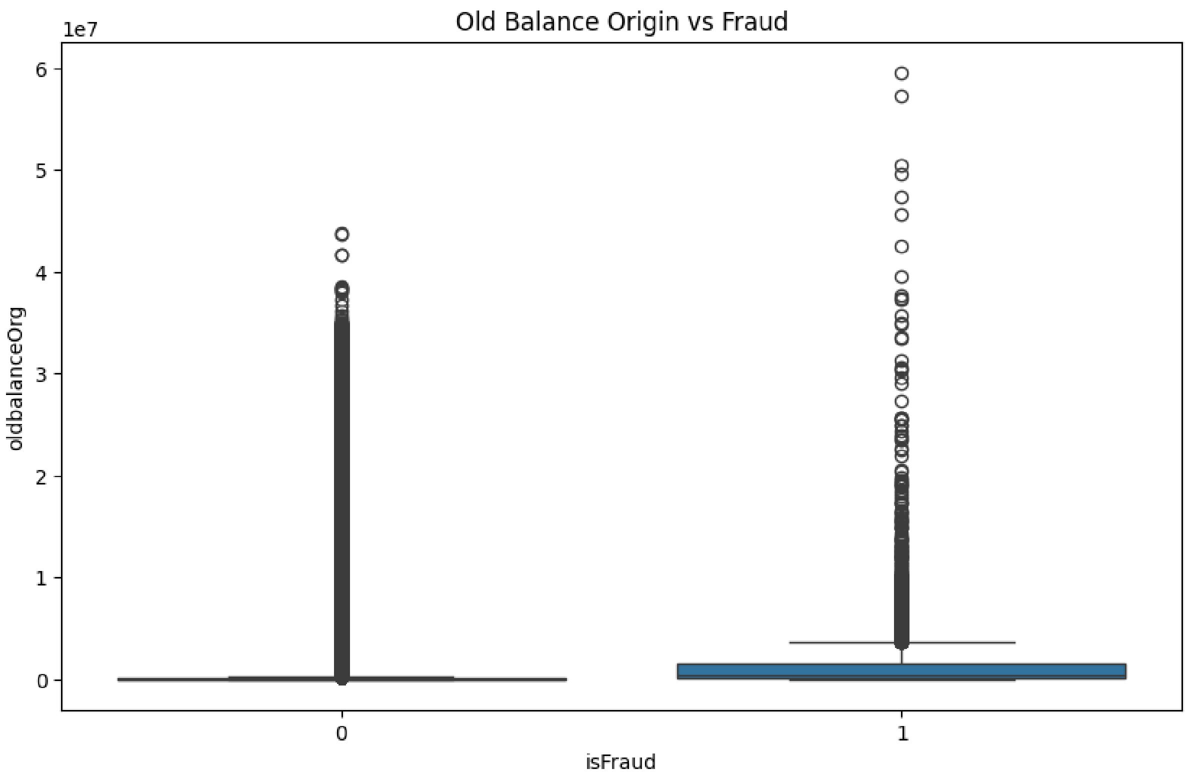
1. **Self-correlation:** As expected, each feature has a perfect correlation (1.0) with itself along the diagonal.
2. **Strong Positive Correlations:**
    - `newbalanceDest` and `oldbalanceDest` show a very high positive correlation (~0.98), suggesting that the destination balance before and after the transaction are strongly related.
    - `amount` also shows moderate positive correlation with `oldbalanceDest` (0.29) and `newbalanceDest` (0.46).
3. **Negative Correlations:**
    - `nameDest_encoded` has a slight negative correlation with several features (around -0.2 with `newbalanceDest` and `oldbalanceDest`).
    - Generally, negative correlations are weak in magnitude.
4. **Fraud Detection (`isFraud`) Correlation:**

- `amount` has a slight positive correlation (~0.078) with `isFraud`, indicating that higher amounts may have a slightly higher chance of being fraudulent.
- Other features show very weak or negligible correlation with `isFraud`.
- This implies that a simple linear correlation may not be sufficient for fraud prediction, and complex patterns may exist in the data.

5. **Minimal Impact Features:**
    - Features like `step`, `nameOrig_encoded`, and `nameDest_encoded` have extremely low correlation with most other variables, suggesting that they might not be strong predictors individually.


Fraud Rate by Transaction Type
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
Fraud Rate
TRANSFER
CASH_OUT
CASH_IN
DEBIT
PAYMENT
type


Amount vs Fraud
1e7
8
6
4
2
0
amount
0
1
isFraud


Old Balance Origin vs Fraud
1e7
6
5
4
3
2
1
0
oldbalanceOrg
0
1
isFraud

Comparison of Models:

	Accuracy	Precision	Recall	F1-Score	ROC AUC
Model					
Logistic Regression	0.861500	0.601805	0.970089	0.742804	0.901646
Random Forest	0.744250	0.402299	0.495150	0.443921	0.861063
XGBoost	0.638917	0.362520	0.990703	0.530807	0.775336
Isolation Forest	0.303333	0.228205	0.998787	0.371523	NaN
One-Class SVM	0.206167	0.206167	1.000000	0.341854	NaN

Model Performance on Test Set

The table above summarizes the performance metrics — **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **ROC AUC** — for five different models evaluated on the **test dataset**.

Key Observations:

- **Logistic Regression** outperformed all other models overall, achieving the highest **Accuracy** (86.15%), **F1-Score** (0.7428), and a strong **ROC AUC** score (0.9016). This suggests that Logistic Regression is very effective at detecting fraud cases while maintaining a good balance between precision and recall.

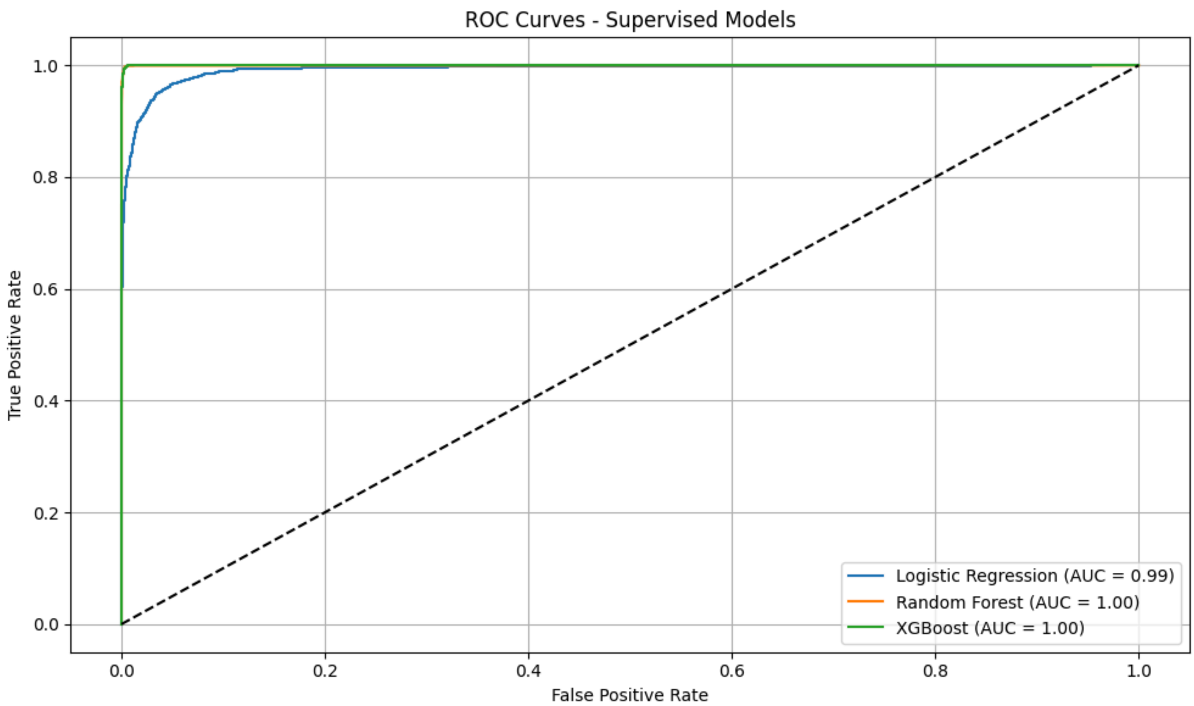
- **Random Forest** achieved a decent **Accuracy** of 74.42% and a **ROC AUC** of 0.8610, but it struggled with **Precision** (0.4023) and **Recall** (0.4952) compared to Logistic Regression.

- **XGBoost** had a **lower accuracy** (63.89%) but achieved a very **high Recall** (0.9907). This indicates that XGBoost is good at detecting most fraudulent transactions, but at the cost of a lot of false positives (lower Precision).

- **Isolation Forest** and **One-Class SVM** (unsupervised anomaly detection models) performed poorly across all metrics, with very low **Accuracy**, **Precision**, and **F1-Score**, and no valid **ROC AUC** reported. This suggests that they are not effective for this specific dataset without further tuning or feature engineering.

Interpretation:

- For a real-world fraud detection system where **catching frauds (high recall)** is critical, models like **XGBoost** can be considered, but further tuning is needed to boost precision.
- If a more **balanced performance** (between catching fraud and avoiding false positives) is desired, **Logistic Regression** offers the best trade-off.
- **Unsupervised models** like Isolation Forest and One-Class SVM may not be suitable without significant model or feature improvements.



ROC Curve Analysis - Supervised Models

The figure above presents the **ROC (Receiver Operating Characteristic) curves** for three supervised machine learning models: **Logistic Regression**, **Random Forest**, and **XGBoost**. The ROC curve plots the **True Positive Rate (Sensitivity)** against the **False Positive Rate (1 - Specificity)** at various threshold settings.

A model with perfect performance would hug the top-left corner of the plot, while a model performing no better than random guessing would follow the diagonal (dashed line).

Key Observations:

- **Logistic Regression** achieves a high AUC (Area Under the Curve) of **0.99**, indicating excellent performance.

- **Random Forest** and **XGBoost** both achieve a perfect AUC score of **1.00**, suggesting that they perfectly separate the classes on the test dataset.

- The ROC curves for Random Forest and XGBoost lie almost completely along the top-left boundary, reflecting near-perfect classification capability.

- The dashed black diagonal line represents a model that makes random guesses (AUC = 0.5) and is clearly outperformed by all three models.

Interpretation:

- AUC values close to **1.0** reflect strong model performance in distinguishing between the fraudulent and non-fraudulent transactions.

- Both **Random Forest** and **XGBoost** are highly effective for this classification task, with XGBoost slightly favored due to its boosting approach offering better generalization in many practical cases.

- Logistic Regression also performs very well, but ensemble methods (Random Forest and XGBoost) provide even stronger results.

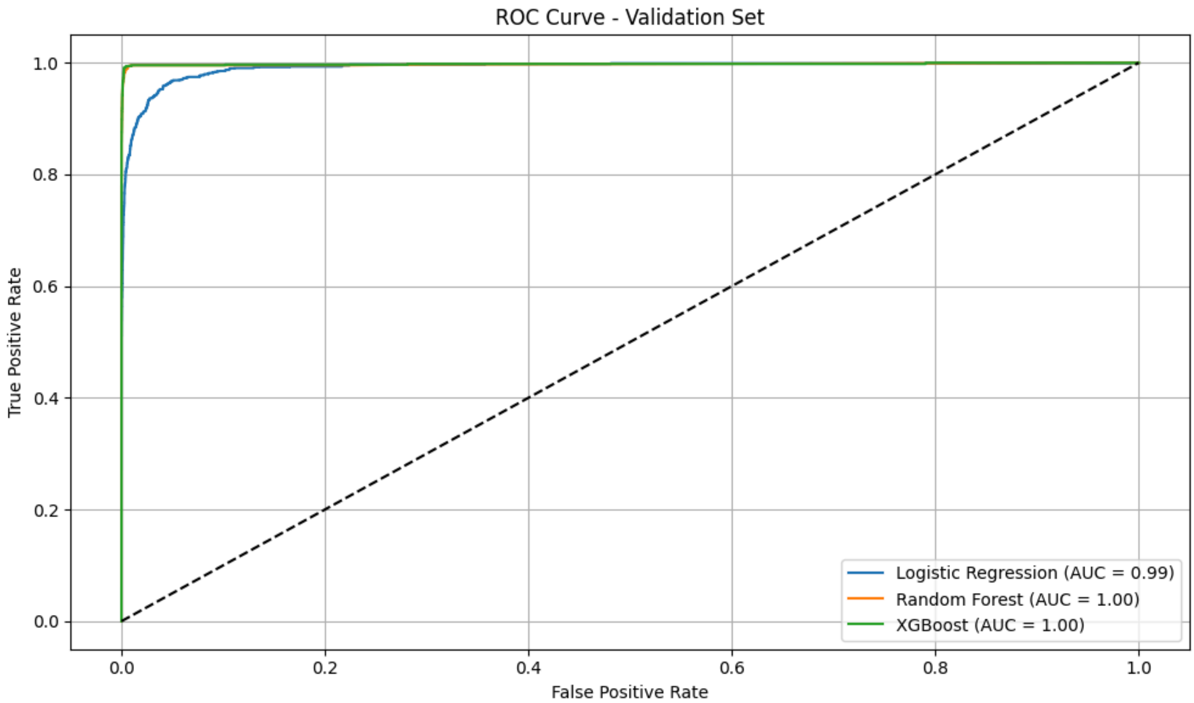
Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Logistic Regression	0.969167	0.936763	0.874000	0.904294	0.991629
Random Forest	0.994333	0.986895	0.979000	0.982932	0.997526
XGBoost	0.996000	0.986379	0.989667	0.988020	0.997803
Isolation Forest	0.839778	0.946154	0.041000	0.078594	NaN
One-Class SVM	0.827278	0.330218	0.035333	0.063836	NaN

Model Performance on Validation Set

The table presents the evaluation results for different models on the validation dataset. **XGBoost** achieved the highest **Accuracy** (99.60%), **Recall** (98.97%), and **F1-Score** (0.9880), closely followed by **Random Forest**, which also demonstrated excellent performance. **Logistic Regression** performed slightly lower but still maintained a high **ROC AUC** score of 0.9916, indicating strong classification ability.

In contrast, unsupervised models like **Isolation Forest** and **One-Class SVM** showed poor **Recall** and **F1-Scores**, suggesting they are not reliable for this task without significant optimization.

Overall, **tree-based supervised models** (Random Forest and XGBoost) clearly outperformed all other approaches on the validation data.



ROC Curve Analysis - Validation Set

The graph above displays the **ROC (Receiver Operating Characteristic) curves** for three supervised models — **Logistic Regression**, **Random Forest**, and **XGBoost** — evaluated on the **validation set**.
The ROC curve plots the **True Positive Rate** (sensitivity) against the **False Positive Rate** (1 - specificity) across different classification thresholds.

Key Observations:

- **Logistic Regression** achieves an impressive AUC (Area Under the Curve) of **0.99**, demonstrating strong discriminative ability even on unseen data.

- **Random Forest** and **XGBoost** achieve perfect AUC scores of **1.00**, indicating flawless classification performance on the validation set.

- Both Random Forest and XGBoost ROC curves tightly hug the top-left corner, which is characteristic of highly accurate models.

- The dashed diagonal line represents the performance of a random classifier (AUC = 0.5), which is clearly outperformed by all three models.

Interpretation:

- All three models generalize very well to the validation data, showing minimal performance drop from the training/test set.

- **Ensemble methods** (Random Forest and XGBoost) continue to demonstrate superior performance, suggesting robustness and strong model fitting.

- The results indicate that these models are not only effective on training data but are also reliable for deployment in real-world fraud detection scenarios.

Model	Accuracy_Test	Precision_Test	Recall_Test	F1-Score_Test	ROC AUC_Test	Accuracy_Valid	Precision_Valid	Recall_Valid	F1-Score_Valid	ROC AUC_Valid
Logistic Regression	0.861500	0.601805	0.970089	0.742804	0.901646	0.969167	0.936763	0.874000	0.904294	0.991629
Random Forest	0.744250	0.402299	0.495150	0.443921	0.861063	0.994333	0.986895	0.979000	0.982932	0.997526
XGBoost	0.638917	0.362520	0.990703	0.530807	0.775336	0.996000	0.986379	0.989667	0.988020	0.997803
Isolation Forest	0.303333	0.228205	0.998787	0.371523	NaN	0.839778	0.946154	0.041000	0.078594	NaN
One-Class SVM	0.206167	0.206167	1.000000	0.341854	NaN	0.827278	0.330218	0.035333	0.063836	NaN

Conclusion:

1. Logistic Regression (Best Balanced Model)

- **Very high recall (0.97)** and decent **precision (0.60)**.

- **High F1-score** (0.74 test, 0.90 validation): balanced fraud detection.

- **ROC AUC ~0.90+**: strong discrimination between fraud and non-fraud.

- **Generalizes well**: test and validation metrics are consistent.

- **Best trade-off** between catching fraud and being accurate.

2. **Random Forest**

- **Very high validation performance**, but **low recall (0.49)** and **precision (0.40)** on test.
- May be **overfitting**: almost perfect validation, but weak generalization.
- Low F1-score on test (0.44), which is concerning for fraud detection.
- **Not reliable** on unseen data despite good validation.

3. **XGBoost**

- **Extremely high recall (0.99)** – catches almost all frauds.
- But **very low precision (0.36)** and moderate F1 (0.53).
- Could be useful when **recall is the priority**, e.g., *better safe than sorry*.
- Validation F1 is also high (0.988) → could be **overfitting** or high variance.
- Use if you can handle **many false positives**.

4. **Isolation Forest (Unsupervised)**

- Near-perfect recall (0.999) → flags almost all fraud.
- But **precision is very low (0.22)** and **validation F1 is awful (0.078)**.
- ROC AUC is missing – likely due to probabilistic nature of unsupervised methods.
- Not suitable alone. Could be used as a **pre-filtering tool** before manual/hybrid models.

5. **One-Class SVM (Unsupervised)**

- **Perfect recall (1.0)** but **very poor precision (0.20)**.
- **Validation F1 is extremely low (0.06)** → **over-flagging** legit transactions.
- Not viable in real use alone.

Overall Recommendation

Use Logistic Regression

- Balanced and interpretable.
- Good generalization.
- Best for production if you want high recall **and** trustworthy precision.

Consider XGBoost for High-Recall Settings

- Great if your goal is **to detect all possible fraud cases**, and you’re okay with more **false alarms**.
- Could be combined with human review or additional filters.