**Fraud Detection Model Analysis Report**

**1️⃣ Problem Statement**

Fraud detection is a **highly imbalanced classification problem**, where the number of fraudulent transactions (positive class) is extremely low compared to legitimate transactions (negative class). In our dataset:

* Total test set fraud cases: **9 positive examples**
* Majority of the dataset: non-fraudulent transactions

Challenges in this scenario include:

1. **Class imbalance:** Traditional models tend to predict the majority class, ignoring rare fraud cases.
2. **Metric sensitivity:** Standard accuracy is misleading; models can achieve high accuracy by predicting only the majority class.
3. **Trade-off between recall and precision:** A model may identify most frauds (high recall) but also misclassify many non-fraud cases as fraud (low precision).

The goal of this study is to **identify models and resampling strategies** that reliably detect fraud while maintaining a practical balance between recall and precision.

**2️⃣ Methodology**

Two approaches were employed to tune and evaluate models:

1. **No-Optuna Approach (Randomized Search / Grid Search CV):**
   * Models trained with standard hyperparameters or tuned using conventional grid/random search.
   * Resampling methods applied: **SMOTE, Tomek Links, SMOTE + Tomek** to handle class imbalance.
2. **Optuna Hyperparameter Tuning:**
   * Automated optimization of model hyperparameters using Optuna.
   * Same resampling techniques applied.
   * Goal: Improve F1-score, which balances precision and recall.

Models evaluated:

* **Tree-based:** RandomForestClassifier, DecisionTreeClassifier, GradientBoostingClassifier, XGBoostClassifier
* **Linear:** LogisticRegression

Metrics analyzed:

* **Recall:** Fraction of actual frauds correctly detected
* **Precision:** Fraction of predicted frauds that are correct
* **F1-score:** Harmonic mean of precision and recall
* **ROC-AUC / PR-AUC:** General classifier performance

**3️⃣ Key Results**

**3.1 No-Optuna Results (Random/Grid Search)**

| **Metric** | **Observation** |
| --- | --- |
| **Best Recall** | RandomForestClassifier\_\_tomek → 0.889, DecisionTreeClassifier\_\_baseline → 1.0 |
| **Best Precision** | Very low across the board (~0.05–0.08) |
| **Best F1-score** | RandomForestClassifier\_\_smote & smote\_tomek → ~0.143 |
| **Other Models** | GradientBoosting, XGBoost, LogisticRegression mostly fail to detect fraud → F1 = 0 |

**Implications:**

* Resampling techniques improve recall but **precision remains extremely low**, leading to many false positives.
* Models with very high recall (e.g., DecisionTree baseline) are impractical, as almost every transaction is predicted as fraud.
* RandomForest with SMOTE/SMOTE+Tomek provides the **most balanced trade-off**: moderate recall (~0.44) and the highest F1 (~0.14).

**3.2 Optuna Hyperparameter Tuning Results**

| **Metric** | **Observation** |
| --- | --- |
| **F1-score** | Top models: RandomForestClassifier\_\_smote & smote\_tomek (~0.143) |
| **Other Models** | Minimal improvement compared to No-Optuna; many models fail to detect fraud or have F1 ~0.09–0.10 |

**Implications:**

* Automated hyperparameter tuning via Optuna **did not significantly improve F1-score**.
* Limitation is **data imbalance and extremely few positive examples**, not hyperparameter selection.
* Precision remains low due to the rarity of fraud cases → even tuned models struggle to predict positives reliably.

**4️⃣ Model Comparison and Selection**

| **Model** | **Resampling** | **Recall** | **Precision** | **F1-score** | **Conclusion** |
| --- | --- | --- | --- | --- | --- |
| RandomForest | SMOTE / SMOTE+Tomek | 0.44 | 0.085 | 0.143 | Best practical model. Detects some frauds with reasonable balance. |
| DecisionTree | Tomek / baseline | 0.667–1.0 | 0.07–0.05 | 0.098–0.132 | Predicts too many false positives; impractical despite high recall. |
| GradientBoosting / XGBoost | Various | 0–0.77 | 0–0.056 | 0–0.105 | Mostly fail to detect fraud; not recommended without additional balancing or feature engineering. |
| LogisticRegression | SMOTE / Tomek | 0–0.77 | 0–0.045 | 0–0.084 | Fails to reliably detect positive fraud cases. |

**Insights:**

* RandomForest consistently outperforms others on small, imbalanced datasets.
* Tomek or SMOTE+Tomek slightly improve recall for tree-based models but F1 remains low.
* GradientBoosting/XGBoost struggle in severe imbalance; may require **class-weight adjustments or cost-sensitive learning**.

**5️⃣ Why RandomForest + SMOTE/SMOTE+Tomek is the Best Practical Model**

1. **Balanced performance:**
   * Achieves moderate recall (~0.44), ensuring some frauds are detected.
   * Highest achievable F1 (~0.14) among all tested models.
2. **Practical trade-off:**
   * Models with perfect recall (e.g., DecisionTree baseline) are impractical due to near-zero precision.
   * RandomForest avoids over-predicting fraud while still identifying actual positives.
3. **Robust to imbalance:**
   * Ensemble nature of RandomForest reduces variance, handling imbalanced data better than single decision trees.
4. **Resampling efficacy:**
   * SMOTE / SMOTE+Tomek helps generate synthetic fraud examples and clean borderline samples, improving model sensitivity to rare events.
5. **Hyperparameter tuning:**
   * Optuna or conventional search had limited effect → confirms **dataset characteristics (imbalance) dominate model performance**, not hyperparameters.

**6️⃣ Recommendations & Future Steps**

To further improve performance:

1. **Feature Engineering:** Create more discriminative features to separate fraud from non-fraud.
2. **Cost-sensitive Learning:** Penalize false negatives more than false positives.
3. **Ensemble Strategies:** Combine RandomForest, DecisionTree, and XGBoost predictions.
4. **Threshold Optimization:** Adjust decision thresholds to maximize F1 or F2 scores.
5. **Anomaly Detection Techniques:** Consider Isolation Forest, One-Class SVM for rare-event detection.

**✅ Conclusion**

For **severely imbalanced fraud detection datasets**:

* **RandomForest + SMOTE / SMOTE+Tomek** is the most **reliable baseline**.
* Achieves a **practical balance** between detecting fraud and avoiding excessive false positives.
* Hyperparameter tuning improves little; focus should shift to **resampling, feature engineering, cost-sensitive methods, and anomaly detection strategies**.