Customer Transaction Prediction: A Machine Learning Approach

**Objective**

To build and evaluate a robust binary classification model for predicting whether a customer will perform a transaction, using a high-dimensional, imbalanced dataset of anonymized features.

**Challenges Faced and Solutions Implemented**

**1️⃣ Large Dataset Size & Computational Constraints**

* **Challenge:** Dataset too large, leading to memory issues and long training times (>1 hour for Random Forest).
* **Solution:** Stratified subsampling to reduce dataset size while preserving class distribution.

**2️⃣ Class Imbalance**

* **Challenge:** Positive class constituted only ~10% of the dataset.
* **Solution:** Applied **SMOTE (Synthetic Minority Oversampling Technique)** to balance the classes.

**3️⃣ High Dimensionality**

* **Challenge:** Dataset contained 200 features, many potentially irrelevant or redundant.
* **Solution:** Used **SelectKBest** to select the top 50 most relevant features.

**4️⃣ Feature Scaling**

* **Challenge:** Features had different scales, affecting distance-based models.
* **Solution:** Applied **RobustScaler** to normalize features.

**5️⃣ Model Training Time**

* **Challenge:** Default hyperparameters of some models (like Random Forest) were computationally expensive.
* **Solution:** Limited the number of estimators, tree depth, and used parallel processing.

**6️⃣ Cross-Validation Overhead**

* **Challenge:** 5-fold CV was too slow.
* **Solution:** Reduced to 3-fold for model comparison and 2-fold for hyperparameter tuning.

**7️⃣ Memory Management**

* **Challenge:** Multiple models and large datasets risked memory overflows.
* **Solution:** Used garbage collection after each model and monitored memory usage.

**8️⃣ Hyperparameter Tuning Complexity**

* **Challenge:** Exhaustive grid search was infeasible on large datasets.
* **Solution:** Reduced search space and CV folds.

**9️⃣ Anonymized Features**

* **Challenge:** No semantic meaning to feature names, limiting EDA.
* **Solution:** Relied on statistical analysis and correlation-based feature selection.

**🔟 Evaluation Metrics for Imbalanced Data**

* **Challenge:** Accuracy alone was misleading due to class imbalance.
* **Solution:** Evaluated using **F1-Score**, **ROC-AUC**, **precision**, and **recall**.

**Methodology**

* Data Balancing: SMOTE
* Feature Scaling: RobustScaler
* Feature Selection: SelectKBest (top 50)
* Model Training: Compared Logistic Regression, Random Forest, Gradient Boosting, SVM, KNN, Decision Tree, Naive Bayes, AdaBoost
* Best Model: Logistic Regression (after hyperparameter tuning)

**Model Comparison**

| **Model** | **Accuracy** | **F1-Score** | **ROC-AUC** |
| --- | --- | --- | --- |
| Logistic Regression | 0.7382 | 0.3549 | 0.8060 |
| Random Forest | 0.7161 | 0.2360 | 0.6480 |
| Gradient Boosting | 0.7006 | 0.2341 | 0.6414 |
| Support Vector Machine | 0.5896 | 0.1345 | 0.4507 |
| K-Nearest Neighbors | 0.3246 | 0.2034 | 0.6013 |
| Decision Tree | 0.5688 | 0.1988 | 0.5740 |
| Naive Bayes | 0.7602 | 0.2711 | 0.6791 |
| AdaBoost | 0.5867 | 0.2157 | 0.6087 |

**Best Model: Logistic Regression**

**Performance After Hyperparameter Tuning (Threshold = 0.6)**

| **Metric** | **Value** |
| --- | --- |
| **F1-Score** | 0.3881 |
| **Accuracy** | 0.8051 |
| **ROC-AUC** | ~0.81 |
| **Precision** | 0.2835 |
| **Recall** | 0.6152 |

Best parameters: {C: 0.1, penalty: 'l2', solver: 'saga'}

**Evaluation Plots**

* 📈 **Confusion Matrix at Threshold = 0.6:**
  + TN: 29,729, FP: 6,251, FN: 1,547, TP: 2,473
  + Shows improved precision-recall trade-off at adjusted threshold.
* 📈 **Precision-Recall Curve:**
  + Displays trade-off between precision and recall for different thresholds.
* 📈 **ROC Curve:**
  + AUC = ~0.81, indicating good discriminatory power.

**Key Achievements**

✅ Addressed class imbalance effectively.  
✅ Reduced feature space from 200 → 50 informative features.  
✅ Improved model performance using hyperparameter tuning & threshold adjustment.  
✅ Provided a thorough model comparison & detailed evaluation.  
✅ Delivered actionable insights with production-ready recommendations.

**Conclusion**

This project successfully built a robust Logistic Regression model to predict customer transactions, handling the challenges of high dimensionality, imbalanced classes, and computational constraints. The final model achieves an F1-Score of ~0.3881 and ROC-AUC of ~0.81, demonstrating strong ability to identify potential transactions while balancing false positives and negatives.