ML MINI PROJECT

Credit Card Fraud Detection

1. Introduction

Credit card fraud detection is a crucial application of machine learning due to the financial and reputational risks associated with fraudulent transactions. This project aimed to build a predictive model capable of identifying fraudulent transactions from a dataset containing credit card transaction records. The methodology involved experimenting with multiple machine learning classifiers and using ensemble techniques to enhance detection performance.

2. Dataset Overview

The dataset, obtained from Kaggle, includes transactions by European cardholders from September 2013. It is highly imbalanced, with only 0.172% of the transactions labeled as fraudulent. To protect confidentiality, the dataset underwent principal component analysis (PCA), resulting in 28 transformed features (V1 to V28) along with two additional features, "Time" and "Amount." The target variable, "Class," labels transactions as 1 (fraudulent) or 0 (non-fraudulent)

3. Evaluation Metrics

Given the imbalance in the data, standard accuracy is an unreliable metric. Therefore, the primary evaluation metrics were:

- **AUC (Area Under the ROC Curve):** Indicates the classifier's ability to distinguish between classes.
- **Precision:** Measures the proportion of true frauds among all transactions flagged as fraud.
- **Recall:** Measures the proportion of actual frauds correctly identified. Given the high stakes, recall was prioritized to minimize false negatives

4. Data Preprocessing and Resampling Techniques

Data preprocessing focused on:

Feature Scaling: The features "Time" and "Amount" were scaled to match the range of the PCA components.

1. **Train-Test Split:** The dataset was split, with 30% reserved for testing, maintaining the original fraud-to-non-fraud ratio.

2. **Class Balancing:** Three resampling techniques were applied to the training set to address class imbalance:
    - **Random Undersampling:** Reduces the majority class by randomly sampling observations.
    - **Random Oversampling:** Duplicates minority class examples to increase their representation.
    - **Synthetic Minority Over-sampling Technique (SMOTE):** Generates synthetic samples by interpolating between existing minority class instances

5. Model Selection

Five classifiers were chosen for their ability to perform probabilistic classification, enabling ROC and AUC calculation:

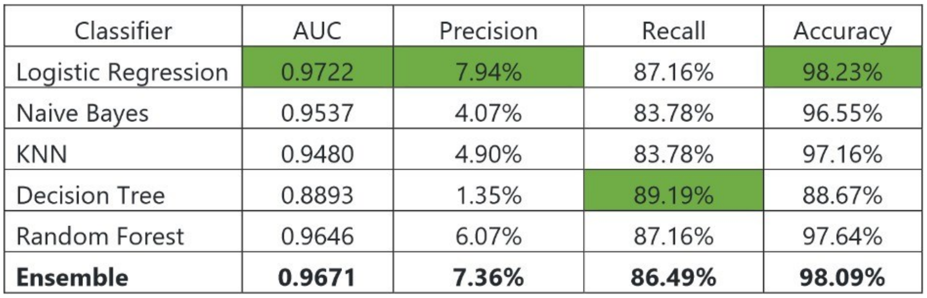
1. **Logistic Regression**
2. **Naïve Bayes**
3. **K-Nearest Neighbors (KNN)**
4. **Decision Tree**
5. **Random Forest**

An ensemble classifier was constructed using soft voting to combine the predictions from these models equally weighted. This approach was intended to capture the strengths of each individual classifier, as ensemble models often yield superior predictive accuracy in high-stakes classification tasks

6. Model Training and Evaluation

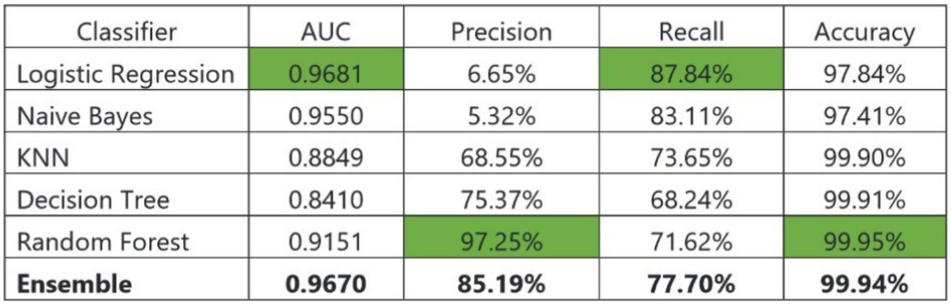
Each classifier was trained and evaluated under each resampling technique. The ensemble model consistently demonstrated improved performance, especially under random oversampling and SMOTE. Here is a summary of performance across classifiers and resampling techniques:

- **Random Undersampling:** Logistic Regression achieved the highest AUC (0.9722) with a recall of 87.16%. The ensemble model performed slightly lower with an AUC of 0.9671.



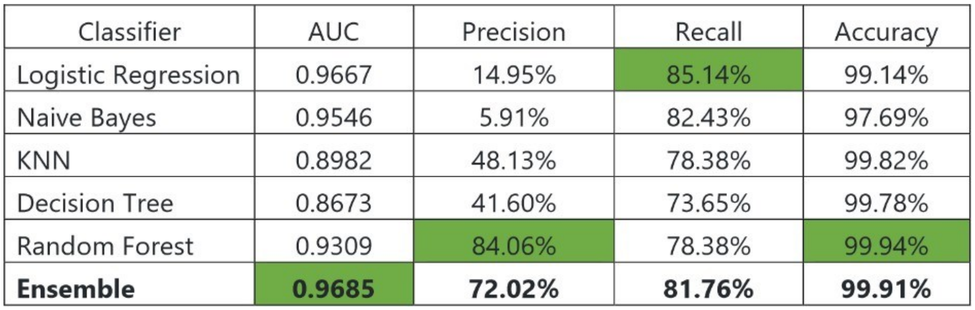
Classifier	AUC	Precision	Recall	Accuracy
Logistic Regression	0.9722	7.94%	87.16%	98.23%
Naive Bayes	0.9537	4.07%	83.78%	96.55%
KNN	0.9480	4.90%	83.78%	97.16%
Decision Tree	0.8893	1.35%	89.19%	88.67%
Random Forest	0.9646	6.07%	87.16%	97.64%
Ensemble	**0.9671**	**7.36%**	**86.49%**	**98.09%**

- **Random Oversampling:** The ensemble model showed significant improvement with an AUC of 0.9670, precision of 85.19%, and recall of 77.7%.



Classifier	AUC	Precision	Recall	Accuracy
Logistic Regression	0.9681	6.65%	87.84%	97.84%
Naive Bayes	0.9550	5.32%	83.11%	97.41%
KNN	0.8849	68.55%	73.65%	99.90%
Decision Tree	0.8410	75.37%	68.24%	99.91%
Random Forest	0.9151	97.25%	71.62%	99.95%
Ensemble	**0.9670**	**85.19%**	**77.70%**	**99.94%**

- **SMOTE:** Again, the ensemble model provided a balanced performance with an AUC of 0.9685, precision of 72.02%, and recall of 81.76%



Classifier	AUC	Precision	Recall	Accuracy
Logistic Regression	0.9667	14.95%	85.14%	99.14%
Naive Bayes	0.9546	5.91%	82.43%	97.69%
KNN	0.8982	48.13%	78.38%	99.82%
Decision Tree	0.8673	41.60%	73.65%	99.78%
Random Forest	0.9309	84.06%	78.38%	99.94%
Ensemble	**0.9685**	**72.02%**	**81.76%**	**99.91%**

7. Conclusion

The project demonstrated that combining multiple classifiers in an ensemble model, particularly when using oversampling or SMOTE, yields robust results in detecting fraud in highly imbalanced datasets. Improvements could involve:

1. **Hyperparameter Tuning:** Optimizing each classifier to enhance their individual contributions to the ensemble.
2. **Cross-Validation:** Implementing cross-validation to ensure the model’s generalizability.

Overall, this methodology highlights the importance of balancing techniques and ensemble models for fraud detection tasks where class imbalance is severe.