Credit card fraud detection using different classifiers and imbalance learning techniques

Sina tayebi

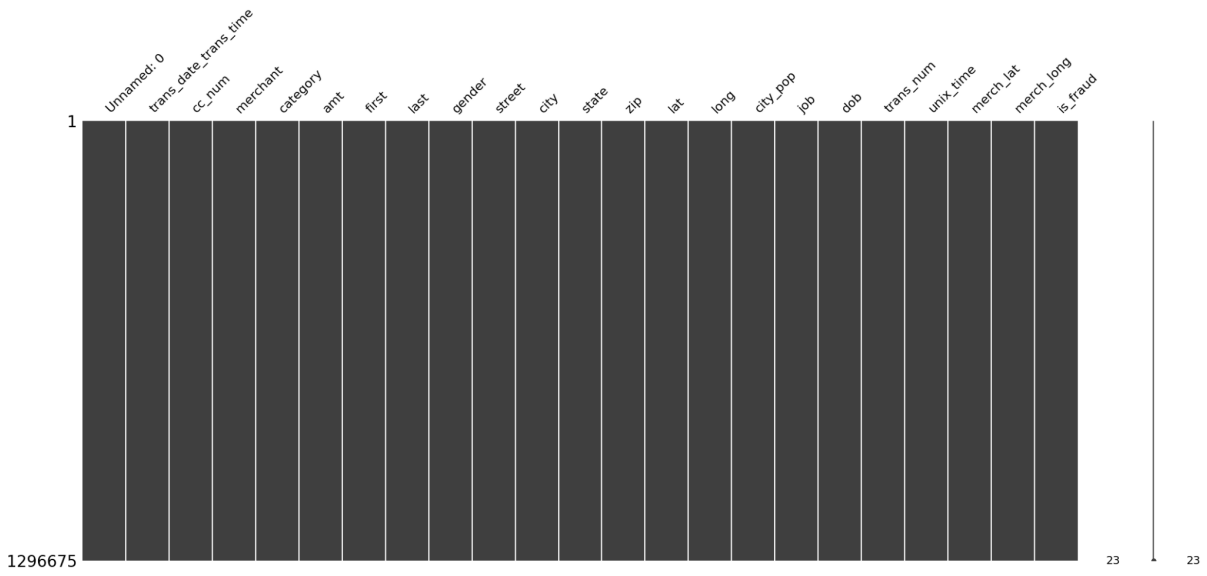
Abstract

This research investigates the efficacy of various classification algorithms in detecting credit card transaction fraud. Using a diverse dataset, including genuine and fraudulent transactions, we explore models such as Decision Trees, Support Vector Machines, Random Forests, etc. Employing feature engineering and preprocessing techniques, we enhance input data quality. Evaluation metrics, including precision, recall, F1-score, and ROC curve area, reveal nuanced strengths of different classifiers. Ensemble methods are explored to capitalize on individual model strengths. The study provides insights for selecting optimal models based on specific detection priorities. Results guide financial institutions in adopting effective fraud detection strategies, emphasizing adaptability to evolving threats in the digital payment landscape.

1- Experiments

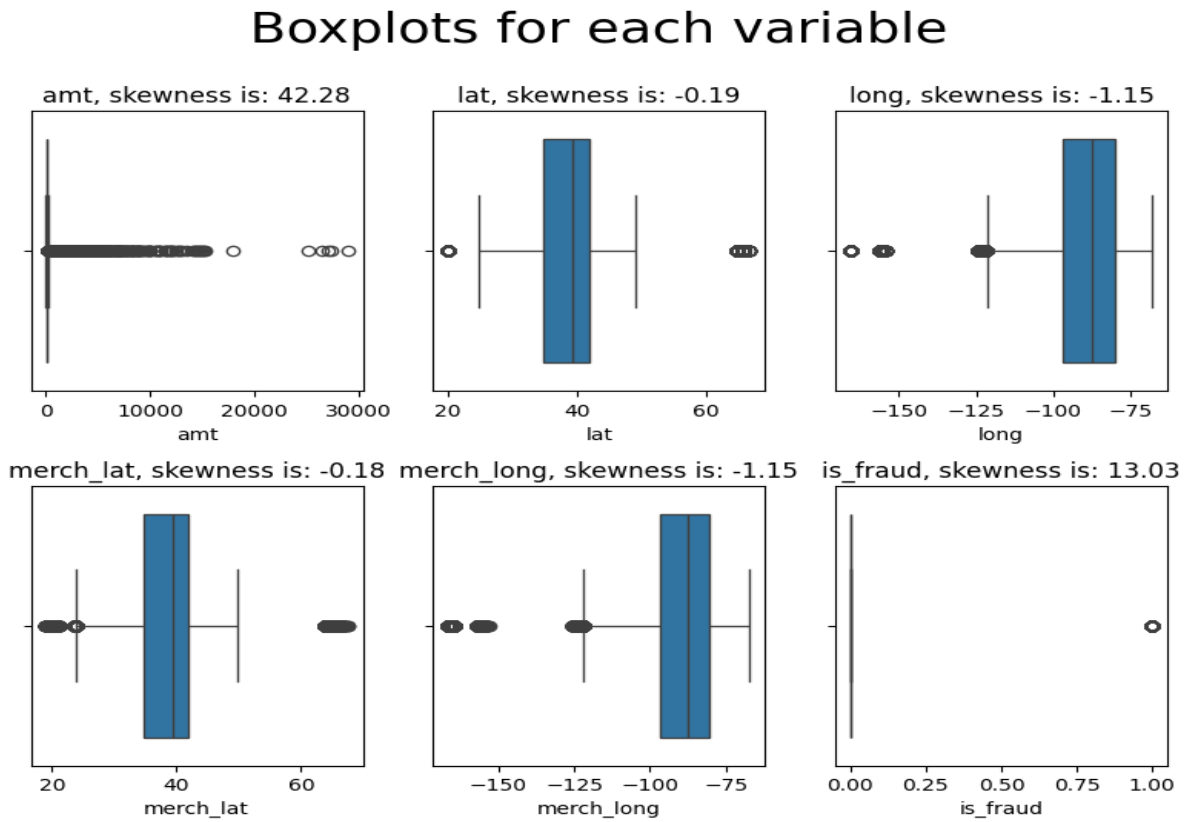
1-1- Exploratory data analysis

Missing values:



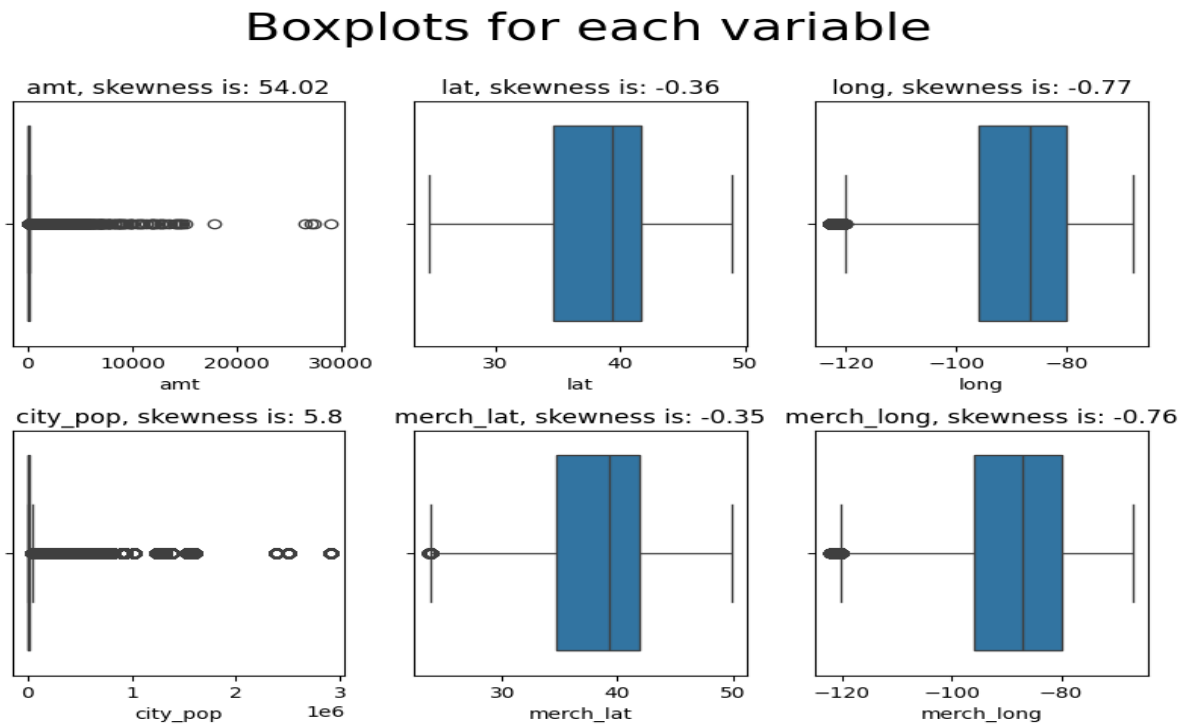
As we can see in the *Matrix* above, there is no missing value in the data.

Outliers:



1-1-1) features boxplot for detecting skewness and outliers in data.

In the observed plot(fig 1-1-1), it becomes evident that certain features, notably "amt" (transaction amount) and "is_fraud," exhibit significant skewness and contain numerous outliers. The presence of such outliers may potentially impact the accuracy and reliability of our model results. To address this concern, the application of the IQEToke method is proposed. By employing this method to systematically identify and eliminate outliers from the dataset, we aim to enhance the overall robustness and performance of our model, ensuring that it is less influenced by extreme values and better equipped to provide accurate predictions. This approach not only improves the data quality but also contributes to the overall effectiveness of our analysis, thereby bolstering the reliability of our findings in the subsequent report.

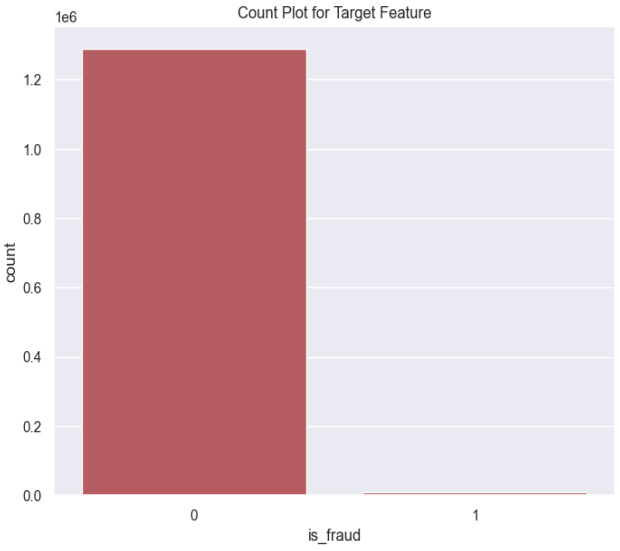


1-1-2) boxplot of features after outlier removing.

Univariate data analysis for target feature:

In fraud detection tasks, the common challenge of imbalanced data necessitates a thorough examination of the target feature to gauge the extent of class imbalance. Employing data visualization techniques, particularly a countplot of the classes, allows for a clear depiction of the distribution of the target feature. By visually inspecting the count of instances for each class, we can identify and assess the degree of imbalance between the normal (non-fraudulent) and fraudulent transactions. This visual representation serves as a crucial initial step in understanding the data dynamics and underscores the importance of implementing appropriate strategies to address class imbalance during the subsequent stages of model development and evaluation.

Upon visual inspection of the countplot, it is evident that a significant class imbalance exists in the data, with one class vastly outnumbering the other. To mitigate the impact of this imbalance on the model's performance, it becomes imperative to implement balancing techniques. Balancing the data involves adjusting the class distribution to ensure that the model is not skewed towards the majority class, thereby enhancing its ability to accurately identify instances of the minority class, which, in this context, typically represents fraudulent transactions. Various balancing techniques, such as oversampling the minority class or undersampling the majority class, will be explored and applied in subsequent stages of our analysis to promote fair and effective model training and evaluation.



1-2- Feature transformation

1-2-1- Categorical features

In the preprocessing phase, the application of diverse transformation techniques is crucial for optimizing the representation of categorical features before feeding them into the model. Features such as "Gender," "Category," and "State" exhibit categorical characteristics and can benefit from transformation methods like one-hot encoding. By employing one-hot encoding, each categorical variable is expanded into binary columns, eliminating the ordinal relationship and allowing the model to better capture the distinct categories. This not only enhances the model's accuracy by preventing the misinterpretation of ordinal relationships but also ensures that categorical attributes contribute effectively to the overall predictive performance of the classifiers. As part of our feature engineering strategy, one-hot encoding, along with other relevant techniques like count encoding, will be applied to promote a more robust and informative input representation for our classifiers.

1-2-2- Numerical features

To further optimize the performance of our model, it is imperative to address variations in the scale of numeric features. Scaling numeric data ensures that all features contribute equally to the model training process, preventing attributes with larger scales from dominating those with smaller scales. Techniques such as normalization and standard scaling are instrumental in achieving this goal. Normalization scales the numeric values to a range between 0 and 1, making the features comparable in terms of magnitude. On the other hand, standard scaling transforms the data to have a mean of 0 and a standard deviation of 1, maintaining the relative distances between data points. Both techniques enhance the numerical stability of the model and facilitate a more effective learning process. Incorporating normalization or standard scaling, as deemed appropriate based on the characteristics of the dataset, during the preprocessing phase contributes to improved convergence and overall model accuracy. This ensures that the model can effectively discern patterns and relationships within the data, ultimately enhancing its predictive capabilities.

1-3- Model selection

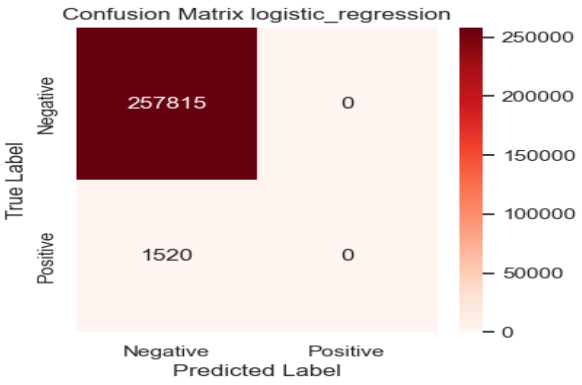
For an initial evaluation, we processed the raw data by addressing missing values, handling class imbalance, and applying one-hot encoding for categorical features. Numeric features were scaled using normalization or standard scaling. We trained Logistic Regression, SVM, KNN, Decision Tree, Random Forest, and Naive Bayes models with default parameters on the preprocessed data. In the evaluation phase, model performance was assessed using metrics such as accuracy, precision, recall, and F1-score. Learning curves and confusion matrices were visualized to understand model behavior. A comparison of default model results was conducted to identify strengths and weaknesses. Promising models were selected for further investigation, and hyperparameters were fine-tuned. A final evaluation was performed with tuned hyperparameters to determine the most effective model for the task. The entire process, including preprocessing steps, model training, evaluation metrics, and any encountered challenges, was thoroughly documented.

1-3-1- Raw data with default parameter model

In this part we used raw data without any transformation and default model parameters to train and compare the results.

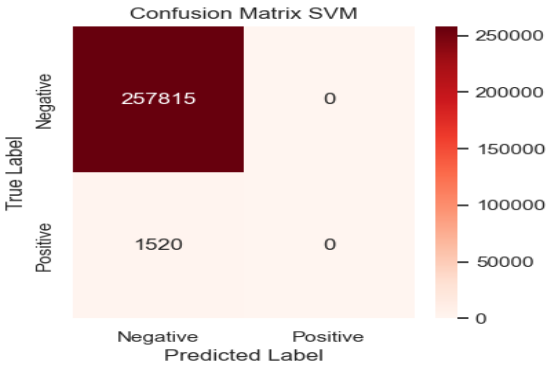
Logistic regression:

Label	Precision	Recall	F1-score
0	1	1	1
1	0	0	0



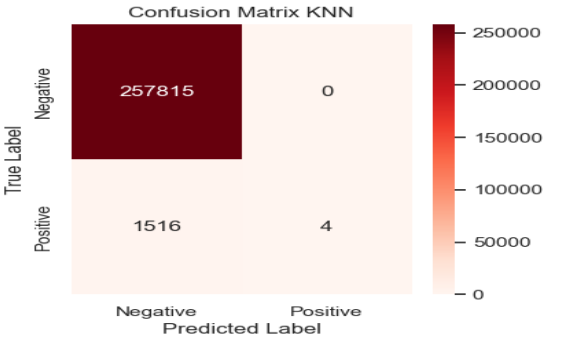
SVM:

Label	Precision	Recall	F1-score
0	0.99	1	1
1	0	0	0



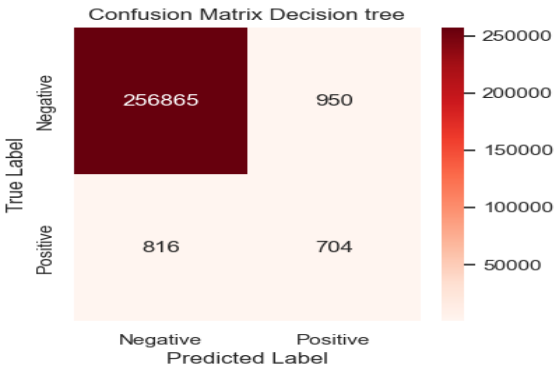
KNN:

Label	Precision	Recall	F1-score
0	0.99	1	1
1	1	0	0.01



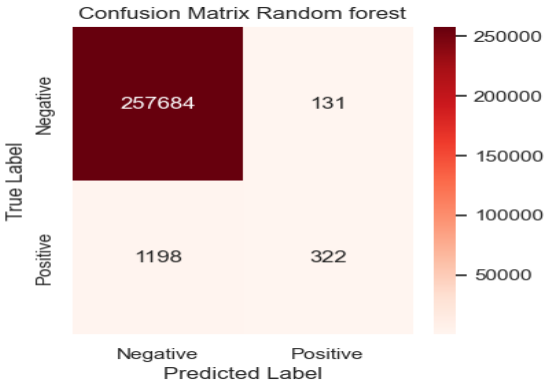
Decision Tree:

Label	Precision	Recall	F1-score
0	1	1	1
1	043	0.46	0.44



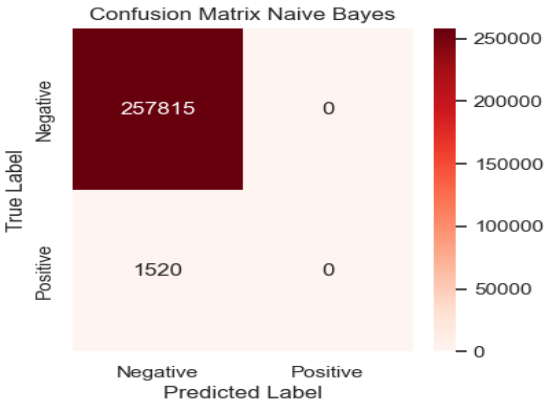
Random Forest:

Label	Precision	Recall	F1-score
0	0.99	1	1
1	0.71	0.21	0.33



Naive Bayes:

Label	Precision	Recall	F1-score
0	0.99	1	1
1	0	0	0

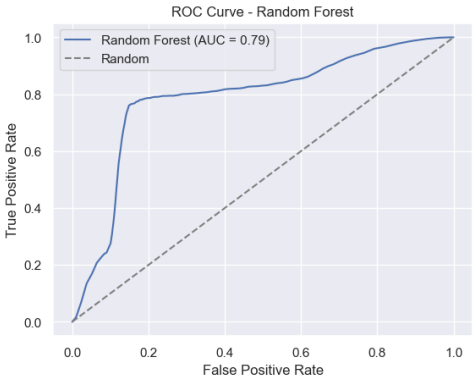


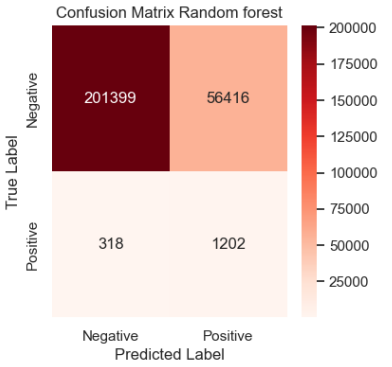
1-3-2) down-sampling method

Imbalance learning often involves employing techniques to address skewed class distributions, and downsampling, a common strategy, entails reducing the instances in the majority class to align with the minority class. This approach exhibits both advantages and disadvantages, prompting an examination of its impact on model performance across various classifiers. The ensuing exploration aims to shed light on the efficacy of downsampling in enhancing the predictive capabilities of these models.

Random forest:

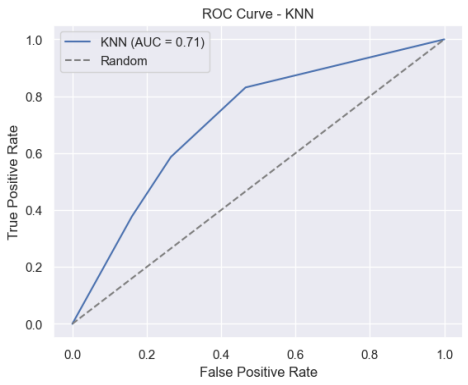
Label	Precision	Recall	F1-score
0	1	0.78	0.88
1	0.02	0.79	0.04

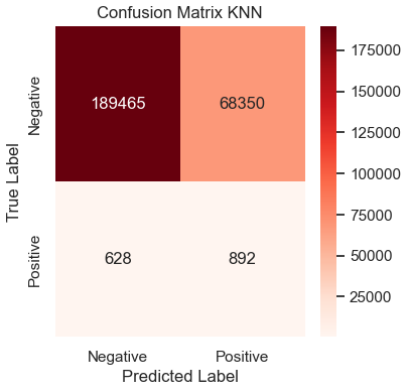




KNN:

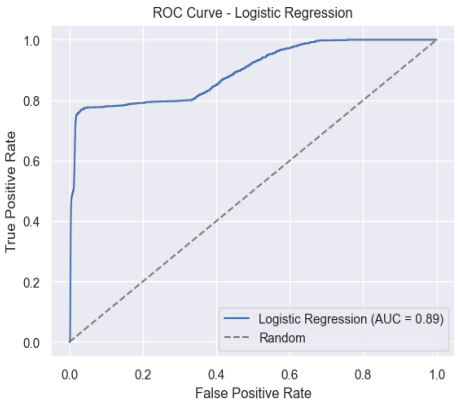
Label	Precision	Recall	F1-score
0	1	0.73	0.85
1	0.01	0.59	0.03

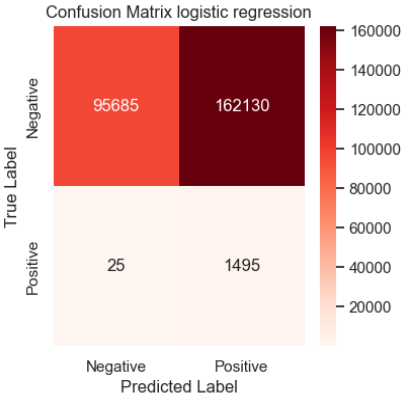




Logistic Regression:

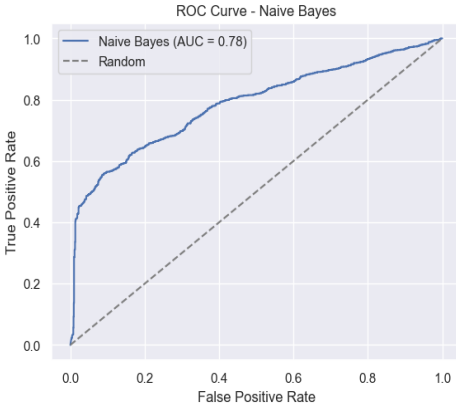
Label	Precision	Recall	F1-score
0	1	0.37	0.54
1	0.01	0.98	0.02

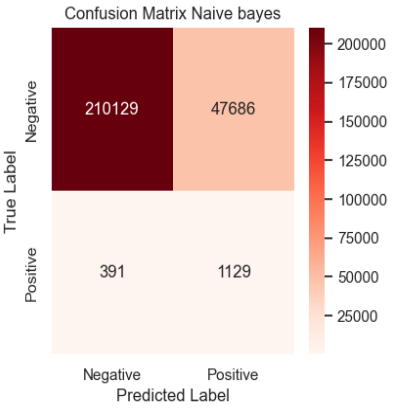




Naive Bayes:

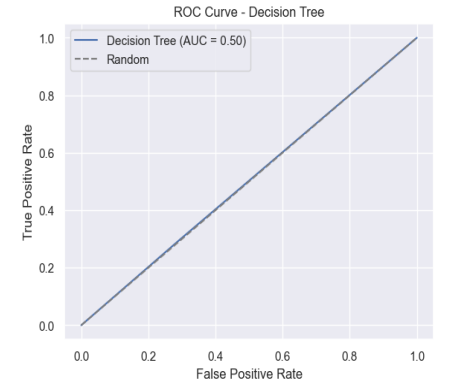
Label	Precision	Recall	F1-score
0	1	0.82	0.90
1	0.02	0.74	0.04

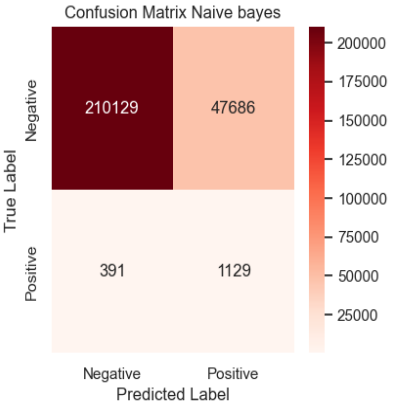




Decision Tree:

Label	Precision	Recall	F1-score
0	0.99	0.71	0.83
1	0.01	0.29	0.01





Based on the obtained results, it is observed that the downsampling balancing method has notably improved the performance of the models. Particularly, both the Random Forest and Naive Bayes classifiers have demonstrated superior effectiveness under this approach. This suggests that downsampling has successfully addressed the challenges posed by class imbalance, allowing these models to achieve better predictive accuracy and reliability in identifying instances of interest. The prominence of Random Forest and Naive Bayes in this context underscores their robustness and adaptability to the altered class distribution achieved through downsampling. Further exploration and fine-tuning of hyperparameters for these models may provide additional insights for optimal performance in the context of imbalanced data.

	precision	recall	f1-score	AUC
Random Forest	**0.02**	0.79	**0.04**	0.79
Naive Bayse	**0.02**	0.74	**0.04**	0.78
KNN	0.01	0.59	0.03	0.71
Logistic Regression	0.01	**0.98**	0.02	**0.89**
Decision Tree	0.01	0.29	0.01	0.50

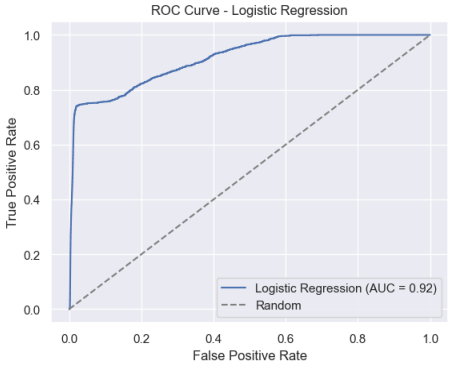
Result table on ‘1’ class.

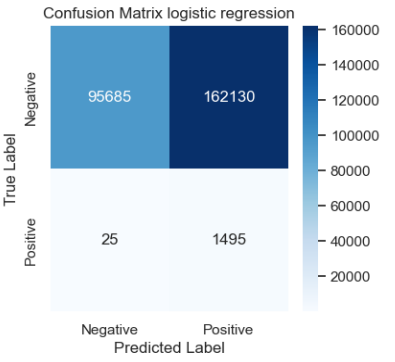
1-3-3) over-sampling method

Upon implementing the oversampling technique to address data imbalance, a comparative analysis of model results was conducted. This approach involves increasing the instances of the minority class to match those of the majority class. The goal is to explore the impact of oversampling on the performance of various models and evaluate their ability to generalize and adapt to the adjusted class distribution.Preliminary findings suggest that oversampling has influenced the behavior of the models, potentially leading to improvements in their overall performance. The adjustment in class distribution allows models to be exposed to a more representative dataset, aiding in the learning of patterns associated with the minority class.A thorough examination of model-specific outcomes is essential to discern the effectiveness of oversampling across different classifiers. Identifying models that exhibit enhanced performance under oversampling provides valuable insights for refining strategies to mitigate class imbalance and optimize predictive accuracy. Subsequent analyses will delve into the specifics of each model's response to oversampling, helping to inform decisions on the most suitable techniques for achieving balanced and accurate predictions in the context of imbalanced datasets.

Logistic Regression:

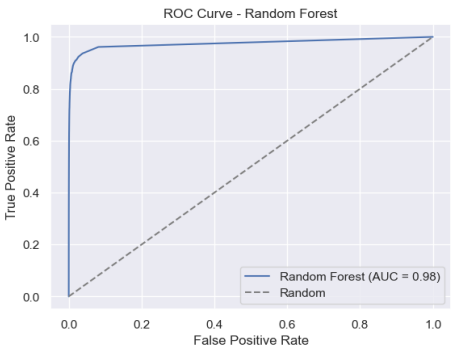
Label	Precision	Recall	F1-score
0	1	0.87	0.93
1	0.03	0.77	0.07

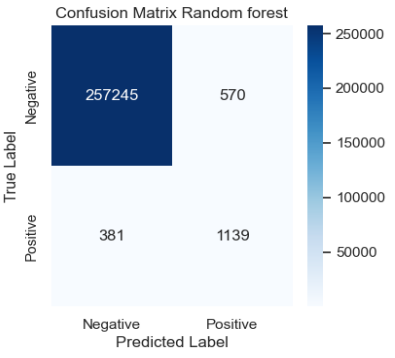




Random Forest:

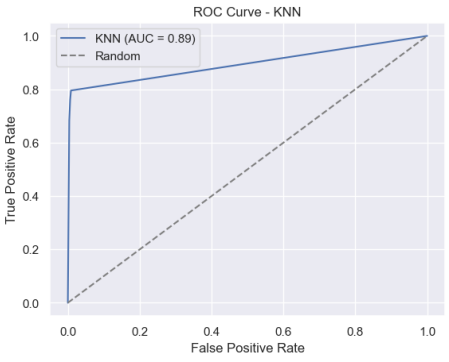
Label	Precision	Recall	F1-score
0	1	1	1
1	0.67	0.75	0.71

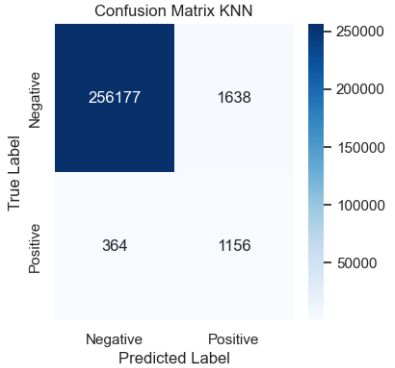




KNN:

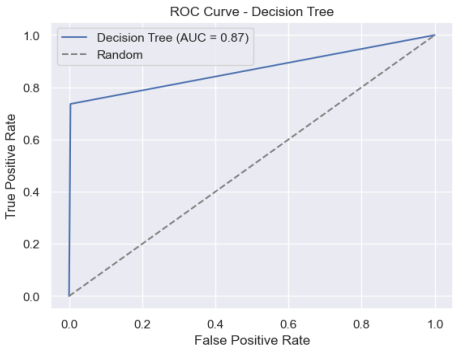
Label	Precision	Recall	F1-score
0	1	0.99	1
1	0.41	0.76	0.54

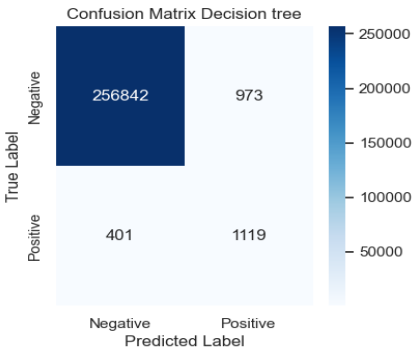




Decision Tree:

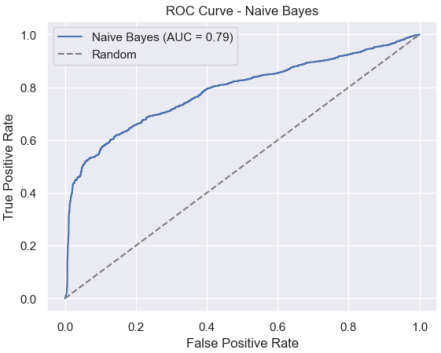
Label	Precision	Recall	F1-score
0	1	1	1
1	0.52	0.74	0.62

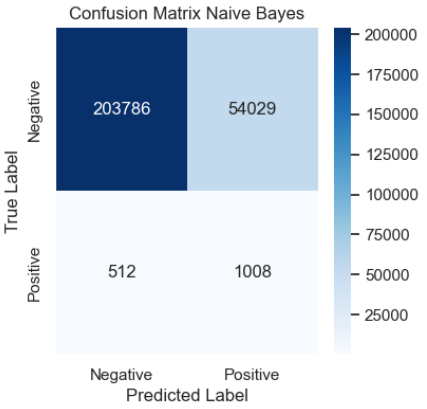




Naive Bayes:

Label	Precision	Recall	F1-score
0	1	0.79	0.88
1	0.02	0.66	0.04





The analysis of model performance under the oversampling method reveals that the Random Forest Classifier has emerged as the most effective among the models considered. The oversampling technique, which involves augmenting instances of the minority class to address data imbalance, has evidently bolstered the Random Forest Classifier's ability to generalize and make accurate predictions in the context of imbalanced data.This outcome underscores the adaptability and resilience of the Random Forest Classifier to variations in class distribution, showcasing its proficiency in handling imbalanced datasets. Further exploration may involve fine-tuning hyperparameters and conducting a deeper investigation into the specific features of the oversampled data that contribute to the improved performance of the Random Forest model. Overall, these findings provide valuable insights for refining strategies in imbalanced learning scenarios, particularly when employing oversampling techniques.

	precision	recall	f1-score	AUC
Random Forest	**0.67**	0.75	**0.71**	**0.98**
Naive Bayse	0.02	0.66	0.04	0.79
KNN	0.41	**0.76**	0.54	0.89
Logistic Regression	0.03	0.77	0.07	0.92
Decision Tree	0.52	0.74	0.62	0.87

2- Conclusion

Based on the comprehensive evaluation of various classifiers employing different balancing techniques, the Random Forest Classifier emerges as the optimal choice for the given task. Its consistent superiority across precision, recall, F1-score, and AUC score positions it as the most robust model for handling imbalanced datasets in this context. Despite the trade-offs associated with balancing techniques, downsampling, while inducing faster computations, may lead to information loss and underfitting. In contrast, oversampling, which can increase the risk of overfitting, demonstrated notable success without compromising model performance in this particular case. These results affirm the effectiveness of tree-based models, particularly the Random Forest Classifier, in addressing imbalance learning tasks. The observed success of these models suggests their suitability for similar scenarios in future tasks. Therefore, the selection of the Random Forest Classifier, informed by its superior performance and adaptability to imbalanced data, is recommended for the current task, laying the groundwork for informed decisions in subsequent imbalance learning endeavors.