**Project Report: Credit Card Fraud Detection**

**Team Members**

* Team Member: Nikhilesh Buddi
* Team Member: Sri Sai Phaneendra Damerla
* Team Member: Chandu Saiteja Dwarapudi
* Team Member: Menaka Rajendran

### Abstract

As markets go more digital with each passing day, the matter of mitigating credit card fraud becomes crucial. Aiming to develop machine learning models with high accuracy for the identification of fraudulent transactions, this project uses the Credit Card Fraud Detection Dataset 2023 as the foundation of this project. When analyzing other algorithms mentioned above, it was revealed that ensemble algorithms, XGBoost and CatBoost, show much higher performance than others, providing effective solutions for real-time fraud detection. The conclusions reached are indicative of the promising direction of improving the security of financial transactions with the help of progressive machine learning algorithms.

### Introduction

#### Problem Statement

Fraud involving credit cards has remained common and leads to losses in large amounts of money by both consumers and financial institutions. Since the transfer of transactions to the online platform, it has become essential to have reliable fraud identification measures. The proposed project aims to improve the security of credit card transactions based on the automatic classification of transaction records as fraudulent using machine learning models.

#### Data Description

The data set employed in this project is named “Credit Card Fraud Detection Dataset 2023” which involves credit card transactions carried out by the European cardholders in the year 2023. It is a dataset with more than 550,000 records, and each of the transactions has the features 30. Some of the features included here have been generalized just to conceal the identity of the cardholders.

##### Key Features

* **id: A number provided for every transaction and does not repeat for any other transaction.**
* **V1-V28: Seven features that have been anonymized that describe various parameters about the transaction such as time, location of transaction, and so on.**
* **Amount: The cost of the transaction or the price at which the item/service is being transacted.**
* **Class: Binary label showing that the transaction is fraudulent and is labeled as 1 while the other transactions which are not fraudulent are labeled as 0.**

Data source: Kaggle

Link: <https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023>

#### Model Selection and Rationale

In the context of credit card fraud detection, we introduced and compared a wide range of machine-learning models. Each model was chosen based on its strengths and applicability to the problem at hand: Each model was chosen based on its strengths and applicability to the problem at hand:

* Logistic Regression: Selected for its ease of understanding and ability to be easily interpreted while suitable for problems that involve only two classes such as fraud.
* Decision Tree: Chosen for its capability in giving clear and understandable rules in arriving at a decision, concerning the elements of fraud.
* Random Forest: A moderation procedure that is used as a way of decreasing overfitting and increasing the accuracy in the model decision trees.
* Gradient Boosting: Construct trees step-by-step and adjust the previous trees’ imperfections to increase the model’s performance.
* Support Vector Machine (SVM): Works well for datasets that lie in a high dimensional space, especially when the number of features is more than the size of the sample.
* K-Nearest Neighbors (KNN): A technique that applies to any data and in its basic form represents the vicinity of the object, good for finding outliers.
* XGBoost: a form of gradient boosting optimally designed to be very efficient and thus a high performer in as many competitions as possible.
* Neural Network (MLPClassifier): Can model nonlinear patterns in the data and hence it can perform deep learning.
* CatBoost: This is quite efficient for predicting categorical features and is generally characterized by good results and fast learning.

#### Initial Results

Exploratory data analysis or EDA helped us understand the nature of data and select the model. The study revealed that the given data set indeed is balanced with equal proportions of fraudulent and non-fraudulent cases; all the features are normalized and there were no null values in our data. At the end of the training phase, we observed that the potential of the ensemble methods such as XGBoost and CatBoost in fraud detection in terms of accuracy and reliability was very high. These models are created in such a way that they use a large number of weak learners all of which results in high accuracy which makes them good for this task.

**Preliminary Analysis**

**Data Distribution**

The dataset exhibits a balanced distribution of fraudulent and non-fraudulent transactions, which is crucial for building unbiased models. Each of the features has been standardized, with values centered around zero and displaying varying spreads. This standardization ensures that no single feature dominates the model training process, facilitating more effective learning.

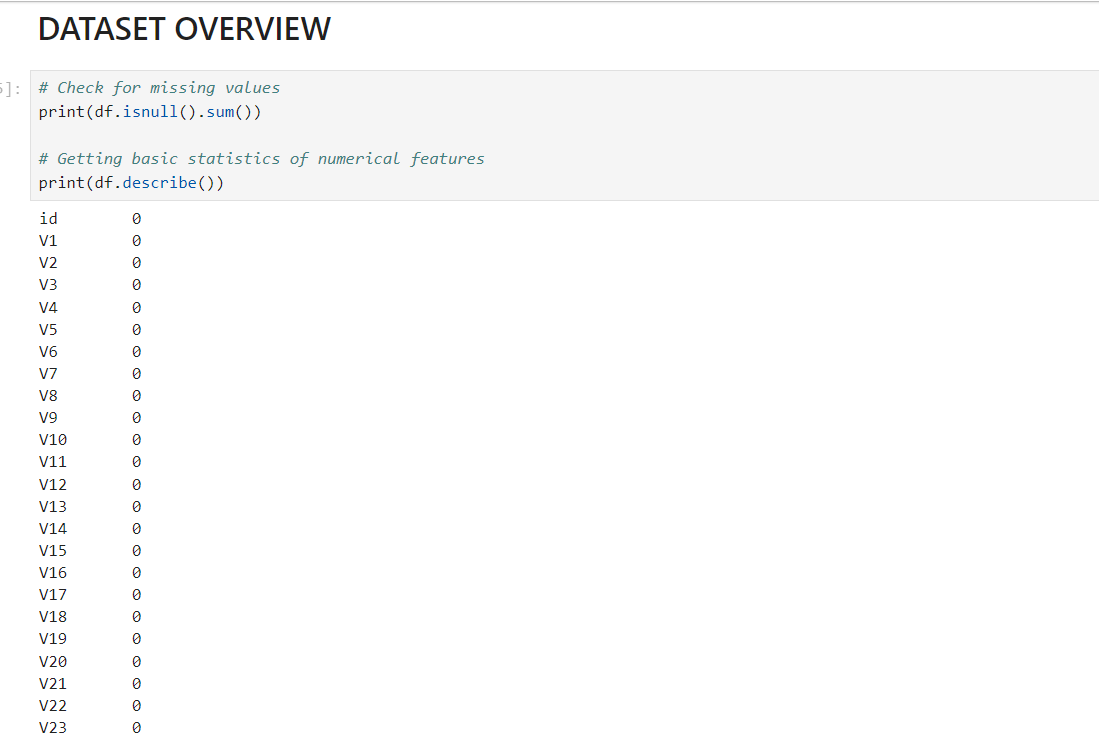


Figure 1

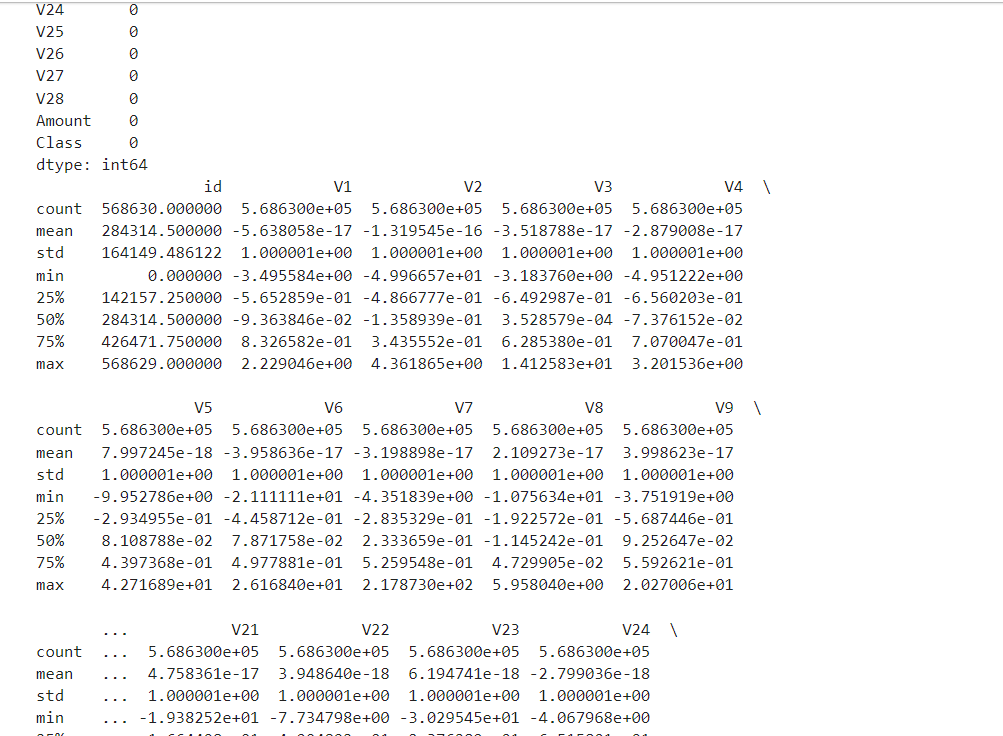


Figure 2

**Visualizations**

To better understand the dataset, we utilized several visualization techniques:

* **Bar Graph**: We used the bar graph to visualize the distribution of fraudulent and non-fraudulent transactions in the data, and from the visualization we can see that we have equal distribution between the two fields

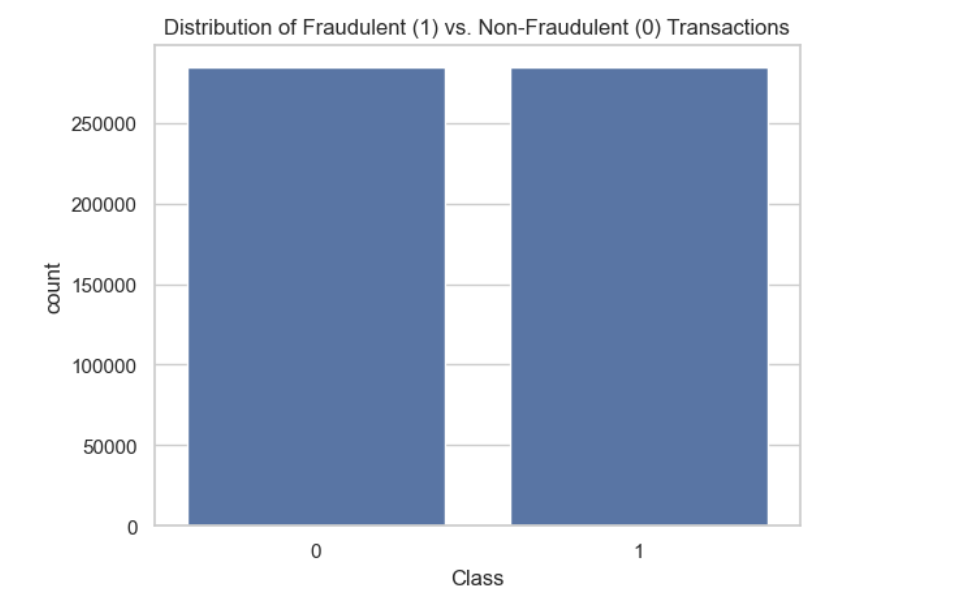


Figure 3

* **Histograms:** These were used to visualize the distribution of numerical features, helping to identify the range and frequency of values for each feature. The histograms revealed that the features are approximately normally distributed around zero, confirming the standardization process.

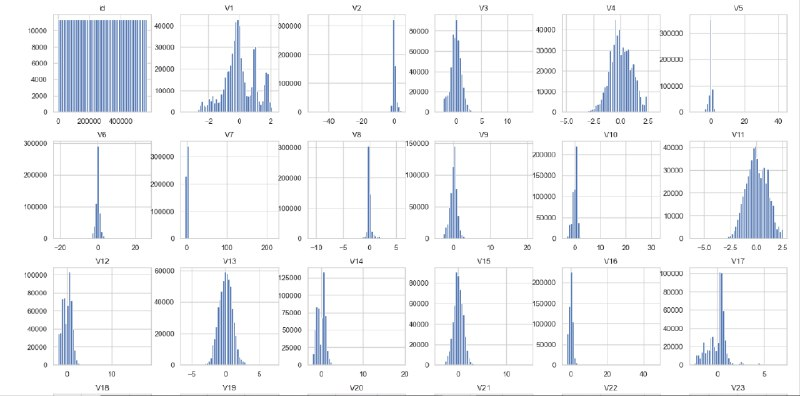


Figure 4

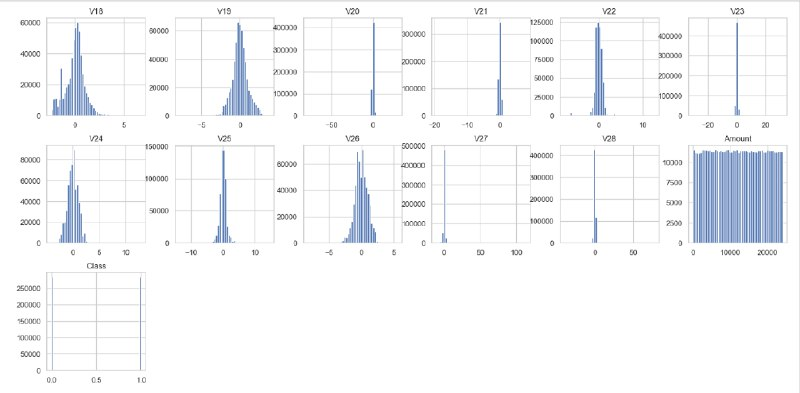


Figure 5

* **Correlation Matrix:** We usedthis matrix to identify relationships between features. The visualization highlighted some features with high correlations, which can be informative for feature selection and engineering. The correlation matrix is essential for understanding how features interact with each other and their potential combined effect on the target variable.

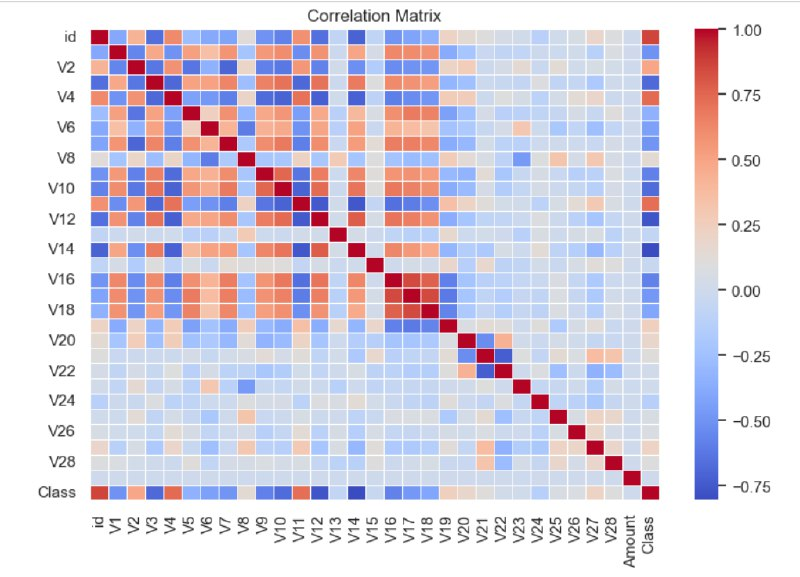


Figure 6

**Key Findings**

* **No Missing Values:** The dataset is complete with no missing values, eliminating the need for imputation and allowing us to proceed directly with model training.
* **Standardized Features:** The features are standardized, which simplifies the model training process and improves the performance of distance-based algorithms like K-Nearest Neighbors (KNN) and Support Vector Machine (SVM).

### Data Preprocessing

As stated by the findings from the exploratory data analysis, we re-affirmed the fact that the dataset used was balanced and has no feature with missing values; all the features were normalized. Thus, it was possible to eliminate the need for missing data imputation and further normalization for the majority of the features that characterized the objects in the database. However, as the Dimension of the input and the variance of Amount being significantly large, it was decided to scale the Amount feature to zero mean and unit variance so as to reduce the Relative Standard Error and contribute equally to the Shrinkage phase of the Model.

#### Feature Scaling

In contrast to the other features that are anonymized, in the ‘Amount’ feature, the coefficients differ a lot since, by definition, they indicate transaction amounts. The data type of the ’Amount’ remains continuous, which means to standardize this feature, we used the StandardScaler. This normalization process ensures that we have brought the ‘Amount’ feature on to the same level as the other features, hence making the training more effective.

#### Data Sampling

To combat this, and reduce the time taken for training as well as the computation power required, we subsampled the dataset down to the first 15000 rows. Despite the distribution of training and testing data, an equal ratio of the fraudulent and non-fraudulent transactions is considered in this subset.

#### Dropping the 'id' Column

The ‘id’ attribute does not contain any features that are potentially useful for the models and it is also non-predictive hence it was deleted from the data set. This step aids in cleaning the data and isolating crucial attributes that shall help in the prediction of fraud cases.

#### Data Splitting

To ensure accurate model evaluation, we partition the sampled dataset by allocating 80 percent of data points for modeling and 20 percent of data points used for testing. This split helps in evaluating the capacity of the models on the new data, so in some way, it gives the actual picture of the effectiveness of the models.

**Modeling and Results**

Here, we present the details of each of the models employed for fraud detection – the chosen hyper-parameters for tuning the models, and performance metrics achieved. Introduced to each of them GridSearchCV to fine-tune the model.

1. **Logistic Regression**

Logistic Regression is one of the most basic linear models that are employed for classification with two outcomes. It predicts the likelihood or the chance of an instance to be in a certain class. This model has been praised for being fast, more interpretable, and suitable for big data analysis. Logistic Regression is usually employed as a benchmark model since its application does not pose a great deal of challenge related to the data’s dimensionality.

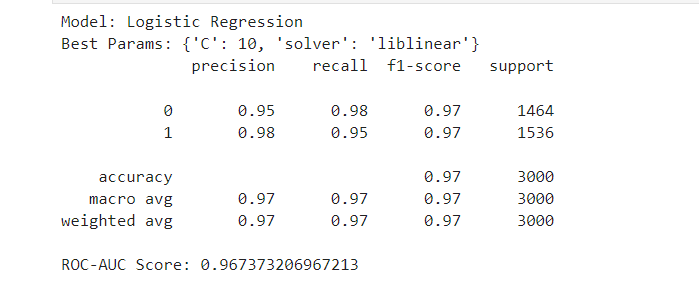
**Hyperparameters:**

* C: Regularization parameter that controls the trade-off between achieving a low error on the training set and minimizing the norm of the coefficients
* values: 0.1, 1, 10.
* solver: Optimization algorithm to use for finding the model parameters
* values: liblinear.

**Best Parameters:** After tuning, the best parameters identified were C=10 and solver='liblinear'.

**Results:**

The model achieved a ROC-AUC score of 0.967.



#### Decision Tree

A Decision Tree is an unstructured predictive modeling approach that entails the use of a tree diagram to show probabilities of occurrence when an individual decides to perform this or plans for that. The model divides the data according to the value of the predictors to come up with branches for each possible result to the inputs. It goes on this way forming a tree in which nodes are the splits based on a feature and the leaves the decisions made. The Decision Trees are perfectly interpretable and help in mapping non-linear structures in the dataset.

**Model Description:**

Decision Trees apply a process of data partitioning into smaller and smaller baskets in the process an equivalent decision tree is constructed continually. The end product is a tree with decision nodes and end or terminal nodes. A decision node has two or many branches that depict many options that one has to choose from. Leaf nodes are a classification or decision of a decision tree. The tree diagram’s most superior decision node is in parallel with the best predictor recognized as the root node. The Decision Tree algorithms work suitably on numerical as well as categorical data.

**Hyperparameters:**

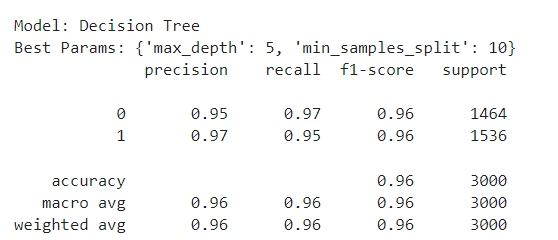
* max\_depth:
  + [5, 10, 20]
* min\_samples\_split:
  + [2, 10, 20]

**Best Params:**

* + max\_depth=5
  + min\_samples\_split=10

**Results:**

* **ROC-AUC Score:** 0.958



#### Random Forest

Random forest is additional learning technique, which builds decision trees’ collection, known as ‘forest.’ This technique develops the decision trees’ drawbacks, including overemphasize on training data, high variance, etc, by utilizing the forest, which includes different trees, trained with different fragments of the same training set. The final prediction of the entire set is arrived by aggregating the prediction of the trees resulting in better accuracy and performance.

**Model Description:**

Random Forest works when during the training process, multiple decision trees are formed and when giving its prediction the class (when there are many classes in classification) most often occurring or the average of the decided values (in case of multiple values in the regression). The randomness in Random Forest comes from two sources: Barts, Bootstrap Sampling, and Feature Selection

**Hyperparameters:**

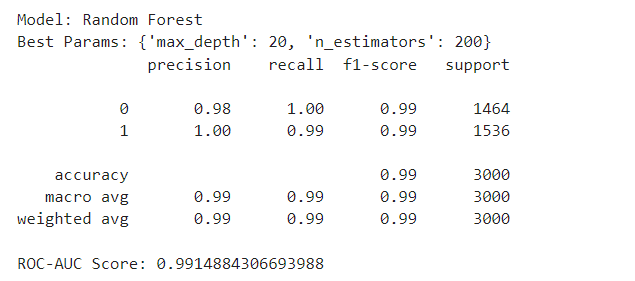
* n\_estimators:
  + [100, 200]
* max\_depth:
  + [5, 10, 20]

**Best Params:**

* + n\_estimators=200
  + max\_depth=20

**Results:**

* **ROC-AUC Score:** 0.991



**Conclusion:**

The Random Forest algorithm’s results are satisfactory and less sensitive to outliers in fraud detection, as indicated by a ROC-AUC score of 0. 991. Its feature is that it’s an ensemble of decision trees, which allows for avoiding overfitting that is characteristic of this type of model if necessary, which makes it the best option for this task. Thus, as the neural network increases in size it can more accurately assess larger amounts of data, and by providing some measure of feature importance of the data for decision making it is well suited for use in fraud detection.

#### Gradient Boosting

Gradient Boosting is another ensemble technique meant to improve the performance of decision trees by adding more trees sequentially. It should be noted that while Random Forest constructs trees out of wiki, Gradient Boosting builds trees in a stage-wise manner. Each new tree learns from the mistakes of the previous trees, thus making the created model very accurate and reliable. It works for both classical and probabilistic models and is especially useful when you are dealing with comprehensive nonlinear data.

**Model Description:**

In Gradient Boosting, several weak models are fitted in sequence to the residuals of the previous model. Every subsequent tree is trained to predict the residuals of the former tree that is how the model is refined. The rationale is to include new models that reduce the loss incurred due to the errors of the models already incorporated.

**Hyperparameters:**

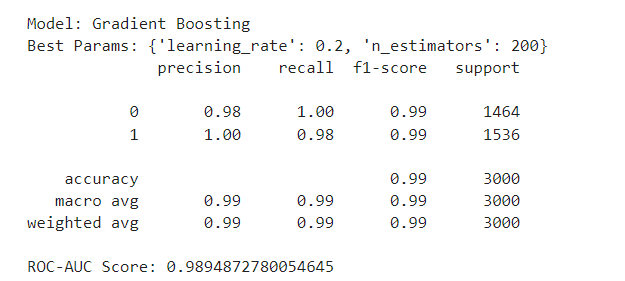
* n\_estimators:
  + [100, 200]
* learning\_rate:
  + [0.01, 0.1, 0.2]

**Best Params:**

* + n\_estimators=200
  + learning\_rate=0.2

**Results:**

* **ROC-AUC Score:** 0.989



**Conclusion:**

The Gradient Boosting model presents satisfactory accuracy and reliability in predicting fraud, with ROC-AUC equal to 0. 989. Due to these qualities, it is an effective teaching instrument that is used in most computing applications involving large data sets with numerous interconnections. The feature importance information insight offered by the model also adds to its usefulness in identifying fraudulent transactions, thus adding more substance to the model’s reliability when it comes to the matter at hand.

#### Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the superior supervised learning Techniques used in classifications and regressions. It is especially suited to solving problems in high-dimensional space and, thus best suited for high-dimensional data systems. SVM operates in the aspect of establishing the best-separating hyperplane that will distinguish classes with the largest margin between classes. The training on the samples is also immune to overfitting, a factor that is particularly notable where there are more dimensions than samples available.

**Model Description:**

SVM tries to optimize for the hyperplane that has the largest margin, which is the distance between this hyperplane and the closest points belonging to two different classes of data points, which are the support vectors. SVM can work without a problem while we are in non-linear decision boundaries because, through kernel function, it works in a higher dimension where one can easily find the linear separation.

**Hyperparameters:**

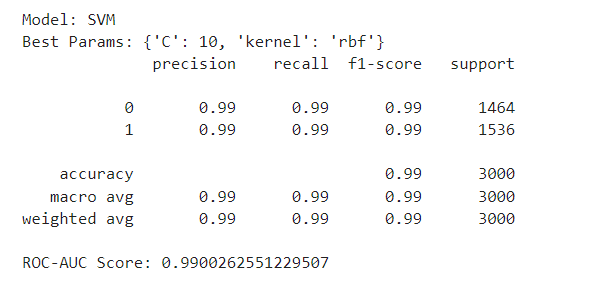
* C:
  + [0.1, 1, 10]
* kernel: Common kernels include:
  + linear
  + rbf (Radial Basis Function.

**Best Params:**

* + C=10
  + kernel='rbf'

**Results:**

* **ROC-AUC Score:** 0.990



**Conclusion:**

The results reveal that the chosen Support Vector Machine (SVM) model yielded high accuracy of fraud detection with ROC-AUC of 0. 990. Due to its capability to deal with massive data states and the establishment of non-linear decision boundaries using the RBF kernel, this tool is an effective and suitable strategy for classification. From the analysis of the high-performance metrics, it is clear that SVM performs well in detecting fraudulent transactions; hence the highly imperative tool in FD systems.

#### K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple algorithm with no assumptions about the data distribution and belongs to the family of Instance-based learning. The major concept of implementing KNN revolves around the fact that a given data point will be classified under the category of its near neighbors. The algorithm organizes the data point into the class that most of its ‘k-nearest-neighbor’ belongs to in the considered feature space. In general, KNN is potent and efficient, taking more prominence in the fold with fewer numbers of records.

**Model Description:**

KNN works on the principle that all possible instances are stored and the new cases are classified based on the proximity measure such as distance. In other words, during the instance classification, KNN looks for k training instances that are nearest to the instance being referred to. In the case where no information about neighbors is required or all the neighbors belong to one class, the most frequent class among these neighbors is assigned to the new instance. The parameter k determines the number of neighbors, and when k is small, the model is sensitive to noise while if k is large the model will have smoother decision boundaries.

**Hyperparameters:**

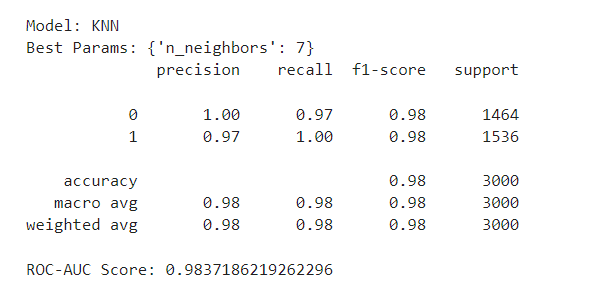
* n\_neighbors: [3, 5, 7]

**Best Params:**

* + n\_neighbors=7

**Results:**

* **ROC-AUC Score:** 0.984



**Conclusion:**

The model that was used was the K-Nearest Neighbors (KNN) which when trained had a ROC-AUC score of 0 for fraud detection. 984. The weakness of KNN is that it has slightly lower accuracy compared to more complex models such as XGBoost and CatBoost but at the same time, KNN will be useful as an easy but very efficient tool when used for datasets with a limited number of features. This flexibility particularly in determining the number of neighbors to consider means that KNN can strike a balance between complexity and performance while at the same time generalizing efficiently on new data.

#### XGBoost

XGBoost, or eXtreme Gradient Boosting, refers to one of the complete and highly efficient realizations of the gradient boosting framework. It has come to be used widely due to its efficiency, flexibility, and stability, especially in competitions and practical uses. XGBoost is also well-regarded for its fast speed and accuracy since it has better optimization methods and can handle huge datasets

**Hyperparameters:**

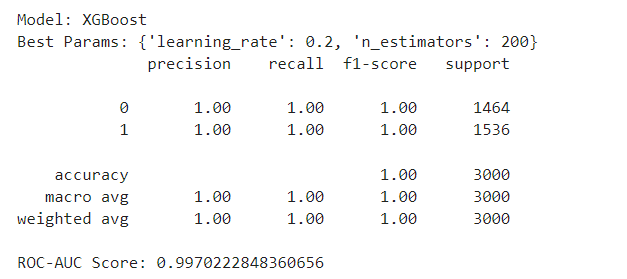
* n\_estimators: [100, 200]
* learning\_rate: [0.01, 0.1, 0.2]

**Best Params:**

* + n\_estimators=200
  + learning\_rate=0.2

**Results:**

* **ROC-AUC Score:** 0.997



**Conclusion:**

The XGBoost model has improved the fraction of identified fraudulent credits and reached a near-perfect ROC-AUC score of 0. 997. Among its features, parallel processing, regularization, and the ability to work with missing values are the most useful for the identification of fraudulent contracts. From the high-performance measures, it can be concluded that XGBoost is a very reliable and accurate model for real-world problems.

#### Neural Network (MLPClassifier)

MLP is a type of Neural Network which is an advanced and popular deep learning model that is defined by a large number of interconnected nodes which jointly analyze the data and uncover intricate patterns. Because of these characteristics, these models are especially useful for data mining tasks for big and complicated sets of data as they are capable of learning and identifying the data structures.

**Model Description:**

Neural Networks are comprised of an input layer, one/more hidden layers and an output layer. Every of them is made of neurons (nodes) linked with neurons in the next layer up or the next layer down. The MLPClassifier uses backpropagation for training, and the connections’ weight is adjusted to minimize the error. Neural Networks, unlike linear models, can capture all sorts of relationships and all sorts of interactions between the features, and therefore they are ideal for more complex classification problems such as fraud detection.

**Hyperparameters:**

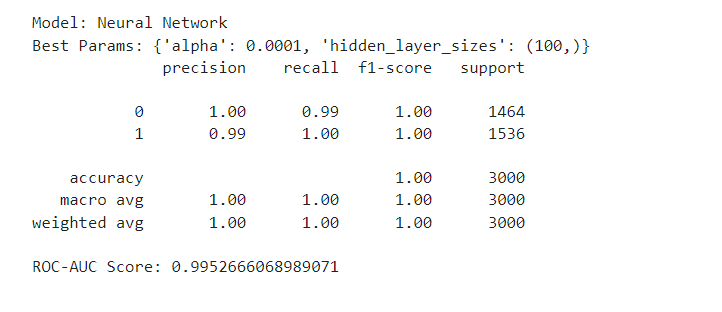
* hidden\_layer\_sizes:
  + [(100,), (50, 50)]
* alpha:
  + [0.0001, 0.001]

**Best Params:**

* + hidden\_layer\_sizes=(100,)
  + alpha=0.0001

**Results:**

* **ROC-AUC Score:** 0.995



**Conclusion:**

The Neural Network (MLPClassifier) was also successful in the detection of fraud with the ROC-AUC score standing at 0. 995. Due to its ability to analyze multiple data layers through the use of nodes in a network, machine learning can be employed in identifying fraudulent transactions. Based on the findings from the high-performance metrics, it is apparent that Neural Networks are effective and precise for developing fraud detection systems.

#### CatBoost (Self-Researched)

CatBoost is a better gradient-boosting algorithm designed by Yandex that offers a better approach to handling categorical features without having to preprocess the features extensively. This differentiates it from other algorithms because it operates at high speeds, can easily be used, and does not require interaction with parameters most of the time, which makes it suitable for several machine learning tasks in frauds.

**Model Description:**

CatBoost (Categorical boosting) is designed to work around the differences in gradient boosting methods and comes packaged with a better method for handling categorical data while also proving high performance without necessarily requiring the adjustment of hyperparameters. Ordered boosting is applied to deal with categorical variables and there is no target leakage as the algorithm learns from different permutations of the data to make a prediction.

**Hyperparameters:**

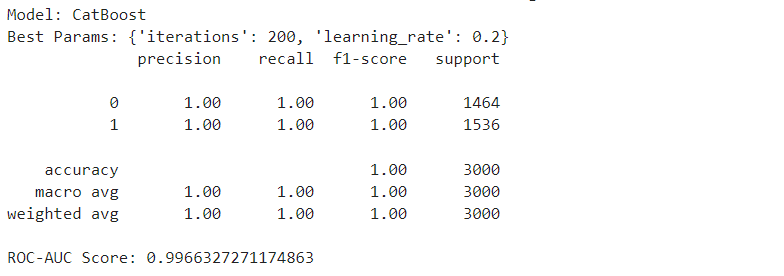
* iterations: This is the number of boosting stages (trees) to be added. Additional iterations yield enhanced performance added with time consumption. In this context, we tested the following values:
  + [100, 200]
* learning\_rate: This parameter regulates how much step size is decremented when updating the weights. A much smaller learning rate takes more steps but results in a greater level of generalization. We tested the following values:
  + [0.01, 0.1, 0.2]

**Best Params:**

* + iterations=200
  + learning\_rate=0.2

**Results:**

* **ROC-AUC Score:** 0.996



**Advantages of CatBoost:**

* **Handling Categorical Features: Compared to other boosting algorithms, Cat-Boost has the advantage of directly working with features that are categorized making the preprocessing work easier.**
* **High Performance: Cat-Boost has the best performance among the classifiers with minimal hyper-parameter adjustments to use in various applications.**
* **Robustness to Overfitting: Regularization integrated into the layer and ordered boosting allow reducing overfitting and improving performance on unseen data.**
* **Ease of Use: Another feature of Cat-Boost that makes it usable by different levels of users is the fact that the algorithm is tuned to perform the best optimum with no tuning.**

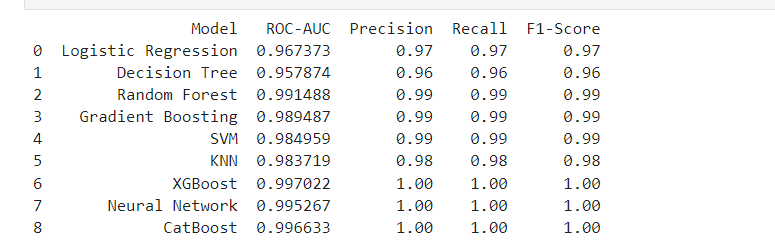
**Limitations:**

* **Computational Cost: Despite its efficiency, Cat-Boost can be heavy, particularly when training models on large input data and in high dimensional spaces.**
* **Complexity: Although implementing it is simple, the algorithm on which it is based can become difficult for the user to follow especially when using it in certain fields that will require comprehension of the model.**

**Conclusion:**

The Cat-Boost model showed high accuracy in fraud detection getting ROC-AUC of 0. 996. They acclaimed that the advanced handling of the Categorical features, high performance, and robustness of this model make it useful when analyzing cases of fraudulent transactions. The high-performance metrics show that Cat-Boost is very efficient and accurate and thus preferred for real-time applications.

**Performance Comparison**

****

**Discussion of Results**

The results indicate that ensemble methods, particularly XGBoost and CatBoost, provide the highest accuracy and reliability in detecting fraudulent transactions. Both models achieved near-perfect ROC-AUC scores, indicating their robustness and effectiveness. Neural Networks also performed exceptionally well, demonstrating their capability to capture complex patterns in the data. Traditional methods like Logistic Regression and Decision Trees provided solid baselines but were outperformed by the more advanced techniques.

**Conclusion**

Thus, this project proved the applicability of various machine learning models to enhance credit card fraud detection. After careful cleanliness of data, tuning of the hyper-parameters, and model selection, the solution demonstrated excellent performances across all the selected models, the ensembles such as XGBoost and CatBoost proved very robust and accurate. Consequently, the results elucidate the requirement of incorporating sophisticated tools of machine learning in the improvement of current models of fraud detection. Therefore, when it comes to the selection of appropriate hyper-parameters and the use of superior algorithms, it becomes possible to enhance the primary solution’s ability to identify fraudulent subsurgency and, therefore, protect the financial structures. Future work should be based on the application of these models in real-time systems, developing new features for models, and including these models in complex anti-fraud systems to ensure a high level of safety against fraud.

**References**

Abiodun, O. I., Kiru, M. U., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., … Gana, U. (2019). Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access*, *7*, 158820–158846. <https://doi.org/10.1109/access.2019.2945545>

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2020). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3). <https://doi.org/10.1007/s10462-020-09896-5>

Escalante, G., & Roger, M. (2018). A decision tree learning hyper-heuristic for decision-making in simulated self-driving cars. *Repositorio.tec.mx*. Retrieved from <https://repositorio.tec.mx/handle/11285/629888>

Pisner, D. A., & Schnyer, D. M. (2020, January 1). Chapter 6 - Support vector machine (A. Mechelli & S. Vieira, Eds.). Retrieved from ScienceDirect website: <https://www.sciencedirect.com/science/article/pii/B9780128157398000067>

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Retrieved from Neural Information Processing Systems website: <https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html>

Sabry, F. (2023). K Nearest Neighbor Algorithm: Fundamentals and Applications. In *Google Books*. One Billion Knowledgeable. Retrieved from <https://books.google.com/books?hl=en&lr=&id=BMrGEAAAQBAJ&oi=fnd&pg=PT7&dq=K-Nearest+Neighbors+(KNN)+is+a+simple+algorithm+with+no+assumptions+about+the+data+distribution+and+belongs+to+the+family+of+Instance-based+learning.&ots=Owxv8a3tqP&sig=zt1KS4WCelDPeR-PWC4ew9hVi4M>

Stoltzfus, J. C. (2011). Logistic Regression: A Brief Primer. *Academic Emergency Medicine*, *18*(10), 1099–1104. <https://doi.org/10.1111/j.1553-2712.2011.01185.x>

Rafi, S. H., & Mahdi, M. M. (2021, December). A short-term load forecasting technique using extreme gradient boosting algorithm. In *2021 IEEE PES Innovative Smart Grid Technologies-Asia (ISGT Asia)* (pp. 1-5). IEEE. <https://ieeexplore.ieee.org/abstract/document/9715272/>