II Trimester MSc (AI & ML)

Advanced Machine Learning

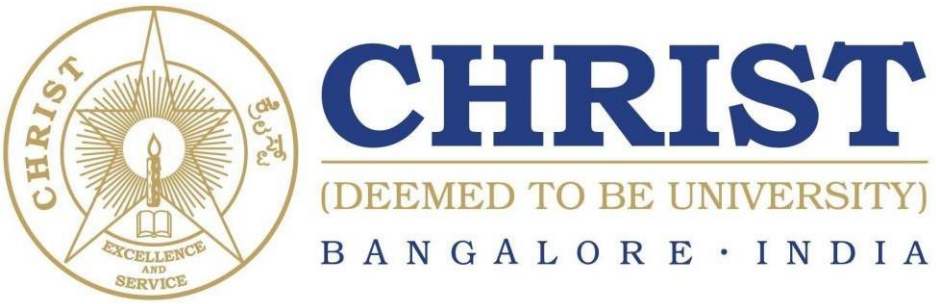
Department of Computer Science

CREDIT CARD FRAUD DETECTION

Submitted by

M SURENDRAN (2348527)
SANJAY S (2348553)
VICTOR JOSE I J (2348570)

January 2024



CERTIFICATE

This is to certify that the report titled ***Credit card fraud detection*** *is a bona fide record of work done by* ***M Surendran (2348527), Sanjay S (2348553), Victor Jose I J (2348570)*** *of CHRIST(Deemed to be University), Bangalore, in partial fulfillment of the requirements of II Trimester of Msc Artificial Intelligence and Machine Learning during the year 2023-24.*

Course Teacher

Valued-by: (Evaluator Name & Signature)
1.

2.

Date of Exam:

Table of Contents

1. Abstract:

This document paper presents a comprehensive study of credit card fraud detection using machine learning algorithms. The primary objective is to classify credit card transactions as fraudulent or genuine based on a highly imbalanced dataset. To address the challenges posed by data imbalance, we employ a combination of undersampling and oversampling techniques. The performance of several classification algorithms, including logistic regression, support vector machines, decision trees, random forests, and K-nearest neighbors, is evaluated using cross-validation and ROC-AUC scores. We also conduct statistical tests, such as analysis of variance (ANOVA), to select informative features and gain insights into the underlying patterns and relationships within the data. Our findings contribute to the knowledge base of credit card fraud detection and provide valuable insights for developing robust and effective fraud detection systems.

2. Introduction:

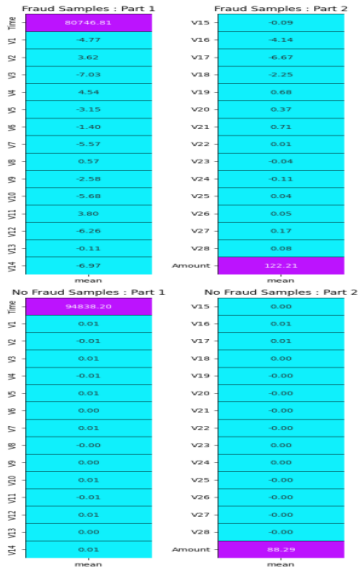
Credit card fraud has become a prevalent issue in the digital age, posing significant financial risks to both individuals and financial institutions. The vast amount of credit card transactions and the ease of online shopping have created ample opportunities for fraudsters to exploit vulnerabilities in payment systems. Hence, the development of automated and accurate fraud detection systems has become imperative. Machine learning algorithms have emerged as a powerful tool in this domain, offering the ability to learn from historical data and identify patterns characteristic of fraudulent transactions.

3. Data Pre-processing and Exploration:

3.1 Data understanding and exploration:

The dataset comprises 284,807 credit card transactions, of which only 492 are labeled as fraudulent, resulting in a highly imbalanced class distribution. We initially explore the data to gain insights into the underlying patterns and trends. We employ descriptive statistics, correlation analysis, and visualization techniques to identify potential indicators of fraudulent transactions.

HEAT MAP:



3.2 Data cleaning and handling missing values

We encounter missing values in some of the features, which can potentially impact the performance of machine learning algorithms. To address this issue, we impute missing values using appropriate statistical methods, ensuring that the integrity and reliability of the data are preserved.

3.3 Data integration and feature engineering:

Due to the highly imbalanced nature of the dataset, we employ data balancing techniques to mitigate the bias towards the majority class. We apply a combination of undersampling and oversampling methods to create a balanced dataset that represents both fraudulent and genuine transactions more equitably.

4. Algorithm Implementation:

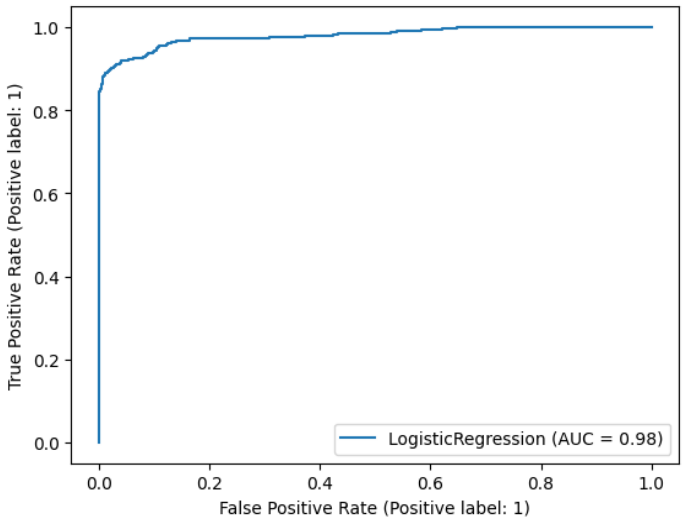
We implemented various machine learning algorithms to classify credit card transactions as fraudulent or genuine. These algorithms include:

4.1.1 Logistic Regression:

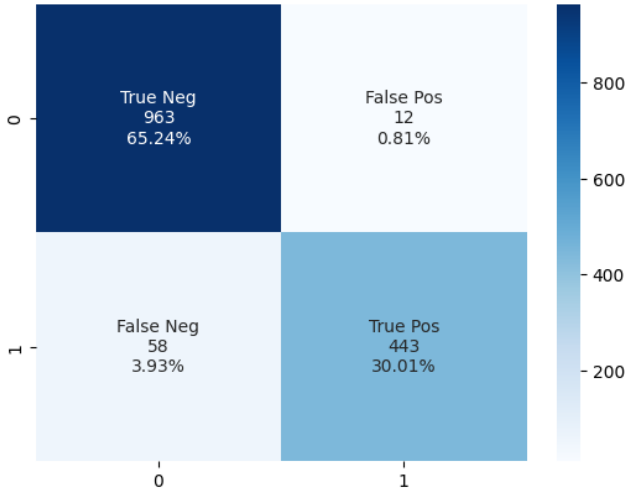
Logistic Regression is a widely used classification algorithm that is employed to model the relationship between independent variables and a binary target variable. This method is particularly valuable when dealing with problems where the outcome variable is categorical, specifically binary, meaning it has only two possible classes or states.

Mathematically, it can be expressed as $z=w0+w1x1+w2x2+…..+wnxn$ where w represents the weights assigned to each feature and x are the input variables.

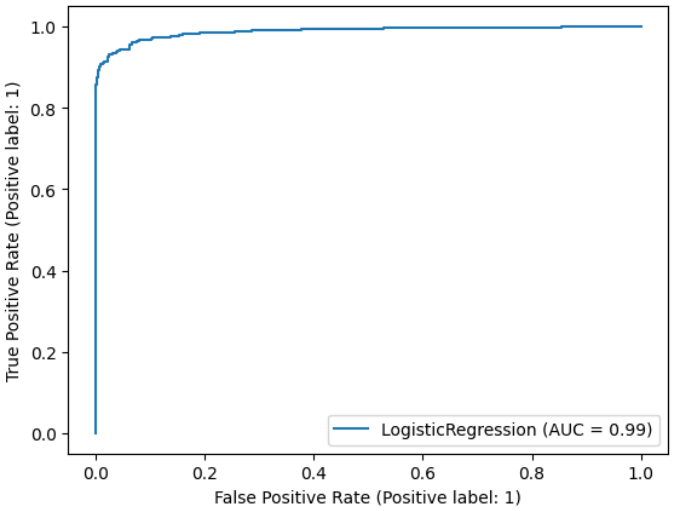
ROC CURVE:



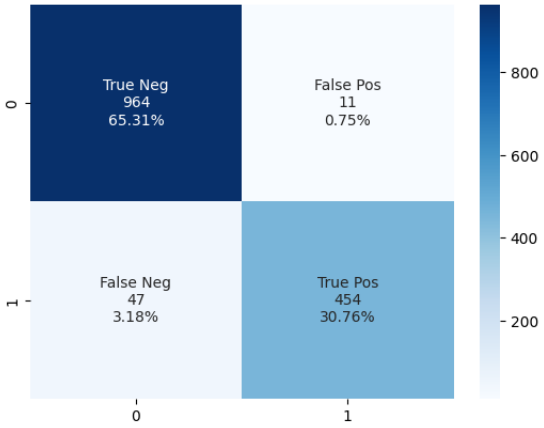
CONFUSION MATRIX:



ROC CURVE FOR ANOVA SCORE:

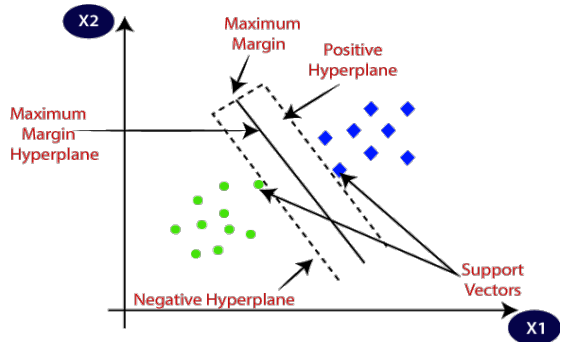


CONFUSION MATRIX FOR ANOVA SCORE:

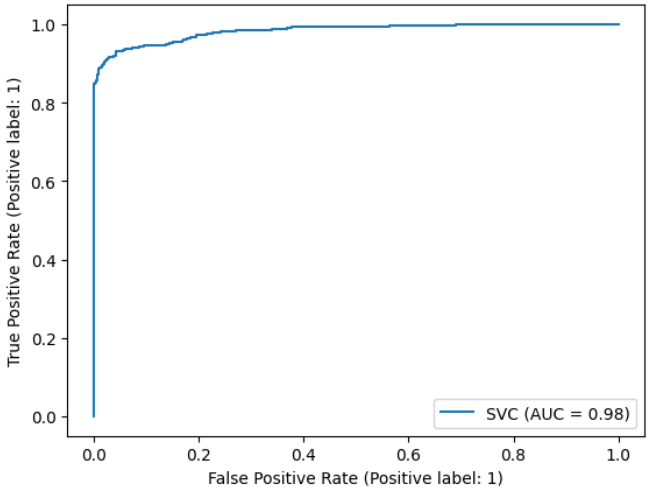


4.1.2 Support Vector Machine:

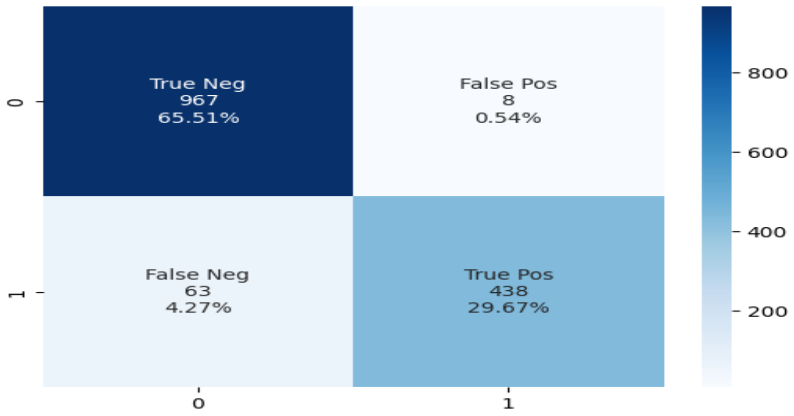
A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression tasks. It is particularly effective in high-dimensional spaces and is well-suited for situations where the data points are not linearly separable. SVM works by finding the optimal hyperplane that maximally separates data points belonging to different classes in the feature space.



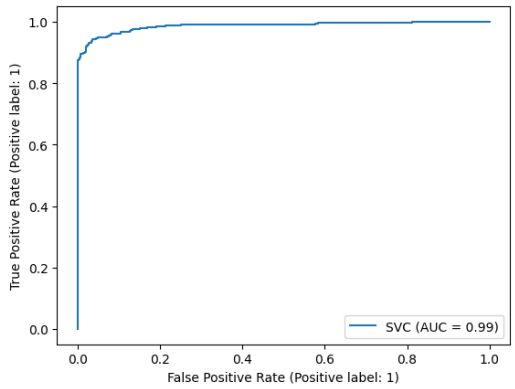
ROC CURVE:



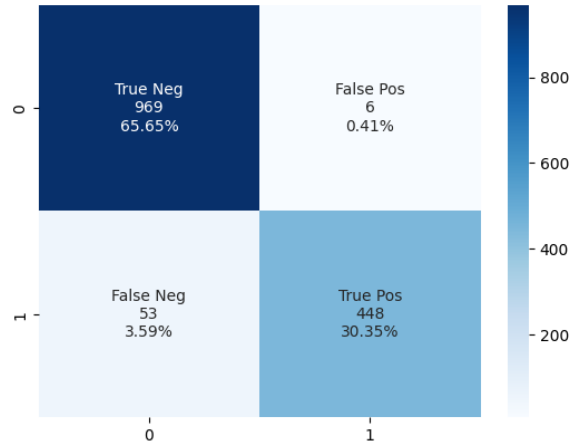
CONFUSION MATRIX:



ROC CURVE FOR ANOVA SCORE:

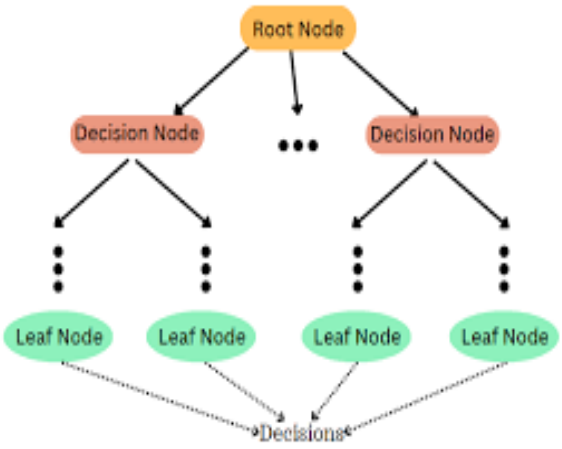


CONFUSION MATRIX FOR ANOVA SCORE:

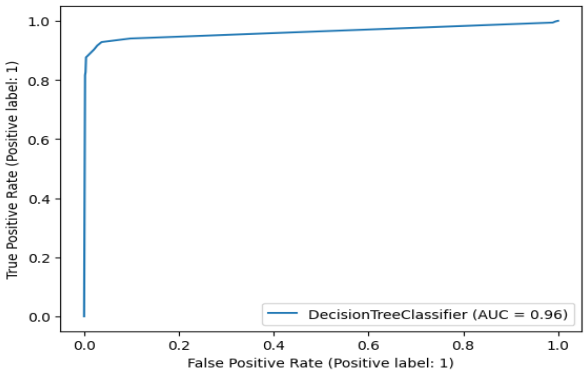


4.1.3 Decision Trees:

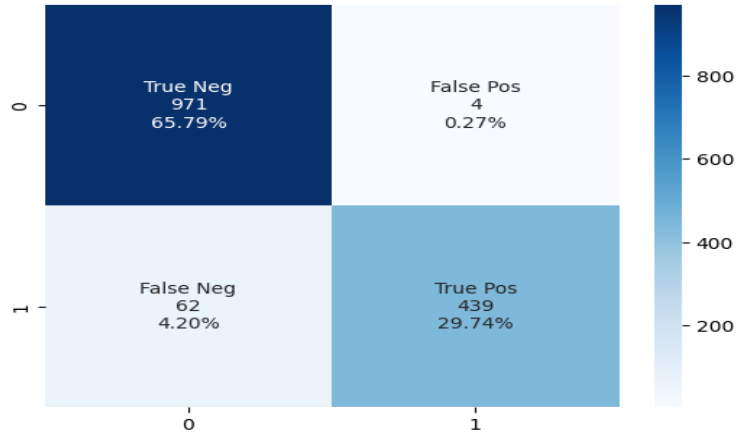
A Decision Tree is a versatile and widely used supervised machine learning algorithm for both classification and regression tasks. It models decisions or predictions based on a set of rules learned from the training data. Decision Trees are particularly appealing due to their simplicity, interpretability, and the ability to handle both categorical and numerical da



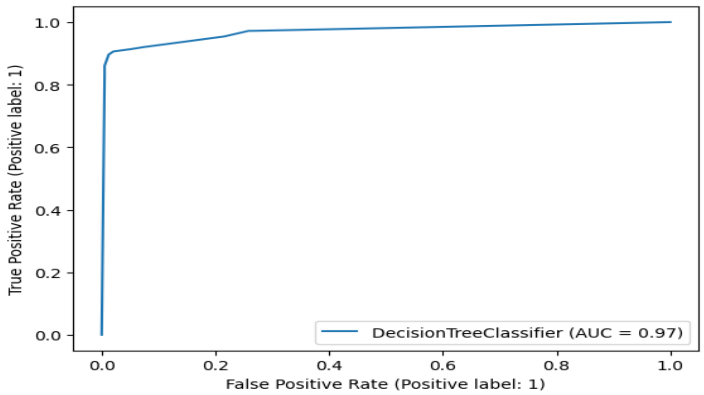
ROC CURVE:



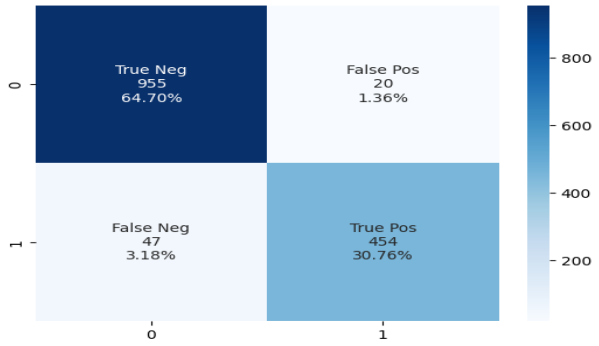
CONFUSION MATRIX:



ROC CURVE FOR ANOVA SCORE:

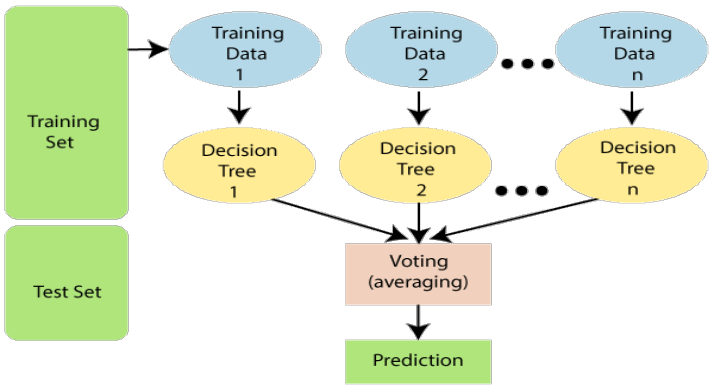


CONFUSION MATRIX FOR ANOVA SCORE:

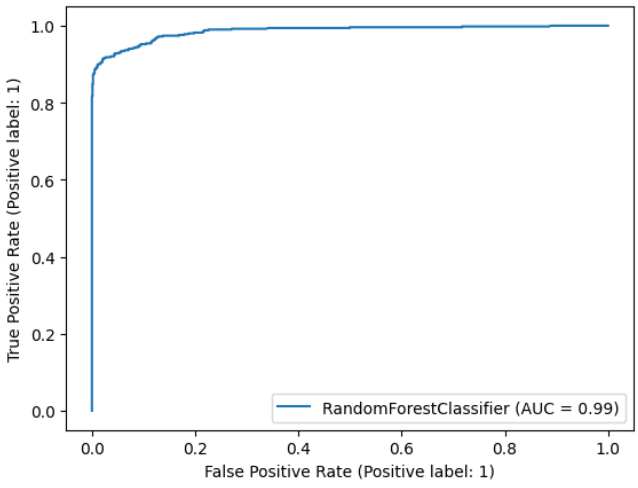


4.1.4 Random Forest:

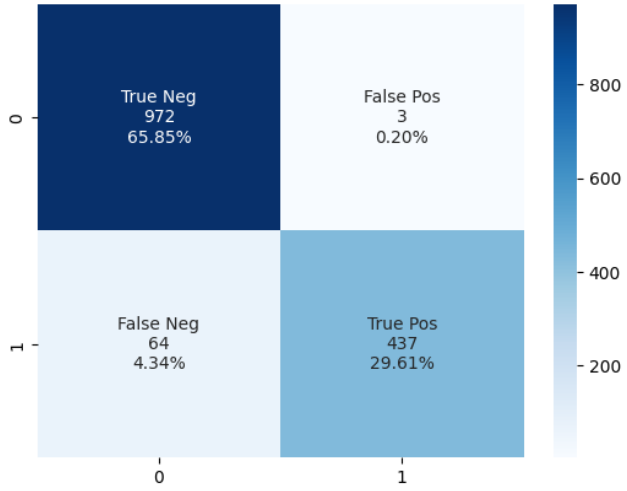
Random Forest is an ensemble learning algorithm that combines the power of multiple decision trees to enhance predictive accuracy and reduce overfitting. In Random Forest, a multitude of decision trees are trained on random subsets of the training data and features. Each tree in the forest independently makes predictions, and the final output is determined through a majority vote (classification) or averaging (regression). This ensemble approach promotes robustness and generalization by mitigating the risk of any individual tree capturing noise in the data. Additionally, Random Forest provides an estimate of feature importance, aiding in understanding the influential factors. The algorithm's ability to handle large datasets, high dimensionality, and various types of data (categorical and numerical) makes it versatile. Random Forest is particularly effective for complex tasks, such as image classification, and is resilient against overfitting due to the inherent diversity introduced by randomization during training. Its popularity stems from its simplicity, scalability, and consistently strong performance across diverse applications.



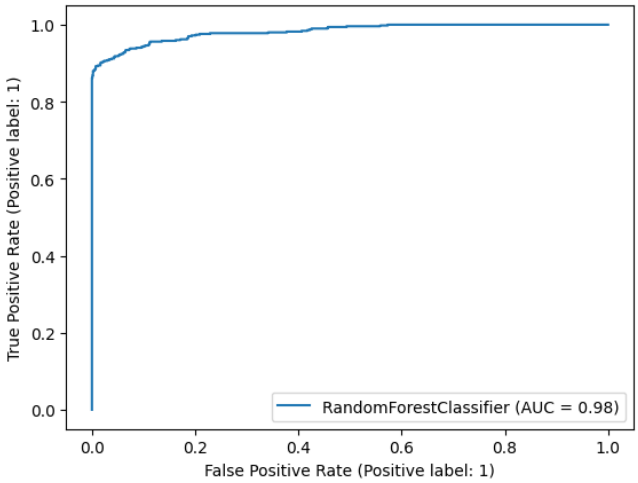
ROC CURVE:



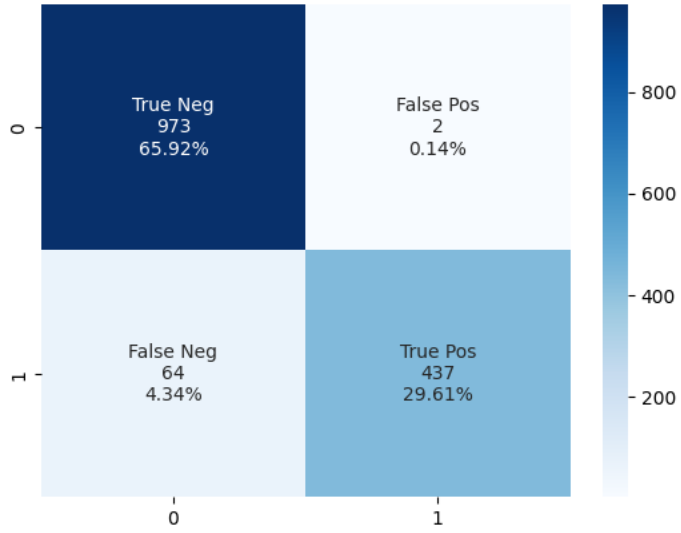
CONFUSION MATRIX:



ROC CURVE FOR ANOVA SCORE:

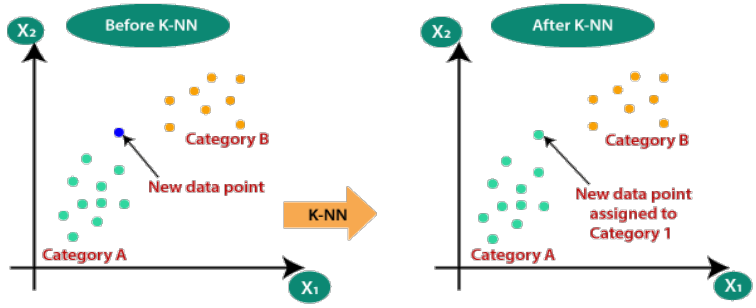


CONFUSION MATRIX FOR ANOVA MATRIX:

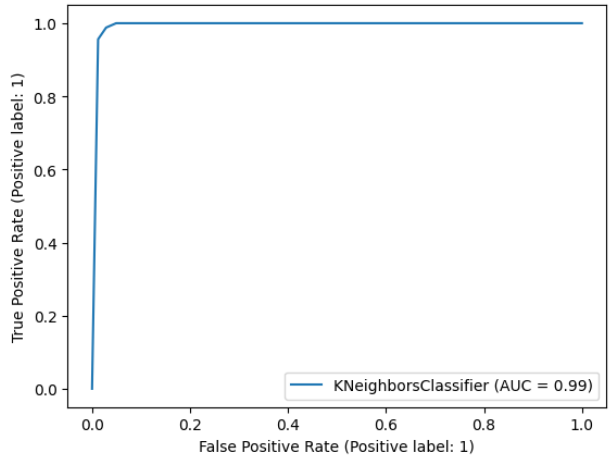


4.1.5 K-Nearest neighbors:

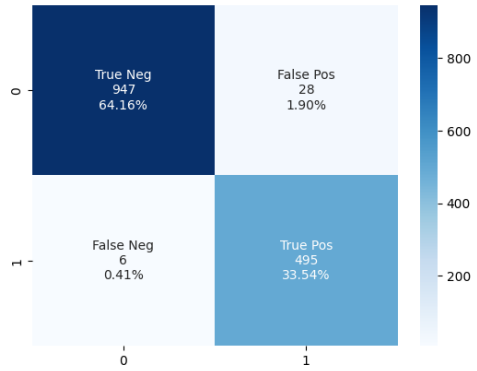
K-Nearest Neighbors (KNN) is a versatile supervised machine learning algorithm. It makes predictions based on the majority class (for classification) or average value (for regression) among the k-nearest data points in the feature space. The choice of k, the number of neighbors, impacts the model's sensitivity to noise. KNN employs a distance metric, such as Euclidean distance, to measure similarity between data points. It has no formal training phase; predictions are made by comparing the new instance to the labeled instances in the dataset. KNN's decision boundaries are flexible, adapting to data patterns, but it may be sensitive to outliers. Feature scaling is essential to ensure fair contribution of each feature in distance calculations. KNN performs well in low-dimensional spaces but can be affected by the curse of dimensionality in high-dimensional datasets. It is commonly used in various applications, such as image recognition, recommendation systems, and medical diagnosis, due to its simplicity and adaptability.



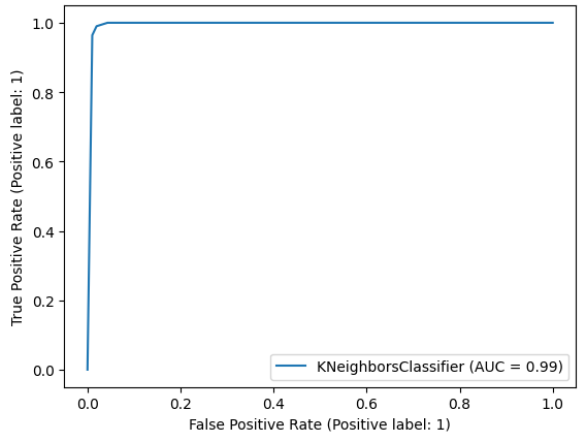
ROC CURVE :



CONFUSION MATRIX:



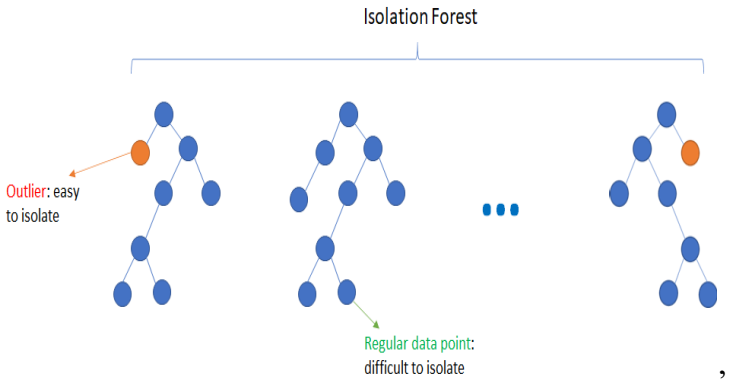
ROC CURVE FOR ANOVA SCORE:



4.1.6 Isolation forest:

Isolation Forest is an unsupervised anomaly detection algorithm based on random forest principles. It isolates anomalies efficiently by observing that they require fewer partitions in decision trees. The algorithm computes anomaly scores based on the average path length of instances across multiple trees in the forest. It is scalable and effective in high-dimensional spaces, requiring fewer trees for accurate detection. Isolation Forest is versatile, handling both numerical and categorical data. It finds applications in fraud detection, network security, and quality control. Key parameters include the number of trees and sub-sampling size. While

effective for global anomalies, it may struggle with contextual anomalies or clustered anomalies. Isolation Forest provides fast and efficient anomaly detection, making it



,

4.2 Correct parameter tuning:

To optimize algorithm performance, a systematic hyperparameter tuning process is employed. This involves experimenting with various hyperparameter combinations using cross-validation techniques. Cross-validation helps assess model generalization by splitting the dataset into training and validation sets multiple times. Different hyperparameter values are tested on these splits to evaluate model performance. The process aids in identifying optimal hyperparameters that maximize predictive accuracy. Utilizing cross-validation mitigates overfitting and ensures the model's robustness to diverse data subsets. The selected hyperparameter values are those consistently demonstrating superior performance across multiple validation folds. This systematic approach enhances the algorithm's efficacy by fine-tuning its configuration. The result is a more accurate and reliable model capable of achieving better generalization on unseen data.

4.3 Efficient Coding and Algorithm Execution:

The implementation of algorithms prioritizes efficiency through adept coding practices, ensuring swift and dependable execution. Utilizing optimized data structures and algorithms contributes to faster processing. Parallel processing techniques are applied when feasible, distributing tasks across multiple processors to concurrently handle computations. This

approach minimizes computational time, enhancing overall efficiency. It is particularly beneficial for algorithms with inherently parallelizable tasks, such as certain machine learning processes. The use of parallel processing also promotes scalability, enabling the algorithm to handle larger datasets or more complex computations efficiently. Careful consideration is given to resource allocation and load balancing during parallelization to maximize performance. The combination of efficient coding practices and parallel processing techniques contributes to the creation of high-performance algorithms capable of addressing computational challenges effectively.

5. Model Evaluation and Performance Analysis:

5.1 Evaluation metrics and performance assessment

We evaluate the performance of the implemented algorithms using various metrics, including accuracy, precision, recall, F1-score, and ROC-AUC score. These metrics provide a comprehensive assessment of the models' ability to correctly identify fraudulent transactions while minimizing false alarms.

Results Table for models based on Confusion matrix :

S.NO	**ML ALGORITHIM**	**CROSS VALIDATION SCORE**	**ROC AUC SCORE**	**F1 SCORE (FRAUD)**
1.	Logistic Regression	98.01%	92.35%	91%
2.	Support vector classifier	97.94%	92.10%	91%
3.	Decision Tree Classifier	96.67%	91.36%	90%
4	Random Forest Classifier	97.84%	91.71%	91%
5.	K-Nearest Neighbors(KNN)	99.34%	97.63%	97%

Results Table for models based on ANOVA Score:

S.NO	ML ALGORITHIM	CROSS VALIDATION SCORE	ROC AUC SCORE	F1 SCORE (FRAUD)
1.	Logistic Regression	98.45%	94.69%	94%
2.	Support vector classifier	98.32%	94.40%	94%
3.	Decision Tree Classifier	97.13%	93.69%	93%
4	Random Forest Classifier	98.20%	94.06%	94%
5.	K-Nearest Neighbors(KNN)	99.54%	98.47%	97%

5.2 Comparative analysis of different models:

Chart Based on Confusion Matrix:

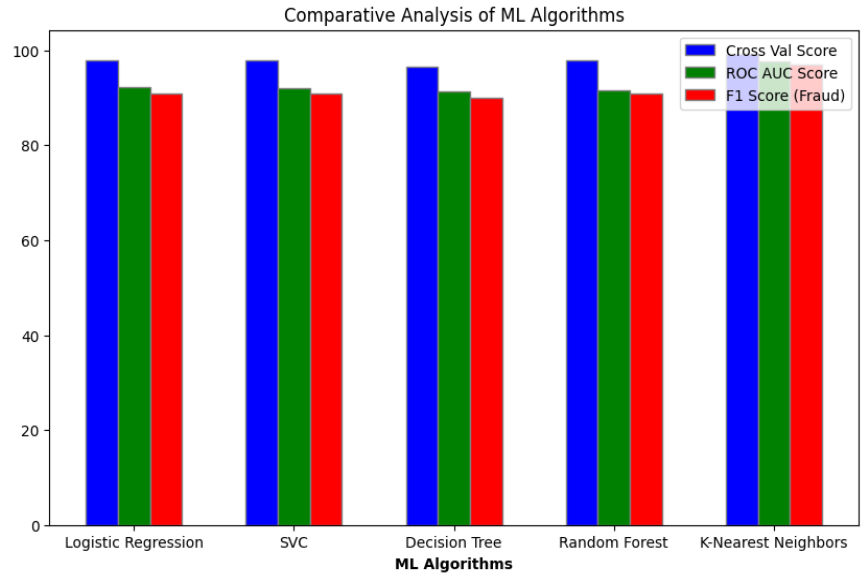
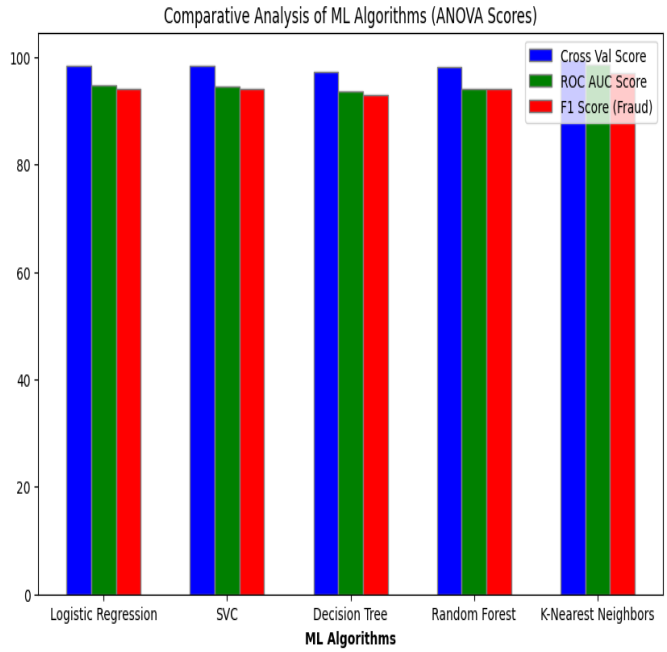


Chart Based on ANOVA Score:



- By this two charts, K-Nearest Neighbors is the best algorithim based on confusion matrix(Cross Validation Score=99.34%, ROC AUC Score=97.63,F1 Score=97% and based on ANOVA Score(Cross Validation Score=99.54%, ROC AUC Score=98.47%, F1 Score=97%)

6. CONCLUSION:

- This is a great dataset to learn about classification problem with unbalanced data.
- As the features are disguised, feature selection cannot be assisted based on the domain knowledge of the topic. Statistical tests hold the complete importance to select features for modeling.
- Due to the use of SMOTE analysis for balancing the data, the models trained on this synthetic data cannot be evaluated using accuracy. Hence, we resort to Cross Validation Score and ROC-AUC Score for model evaluation.
- In this K-Nearest Neighbors is the best algorithim based on confusion matrix(Cross Validation Score=99.34%, ROC AUC Score=97.63,F1 Score=97% and based on ANOVA Score(Cross Validation Score=99.54%, ROC AUC Score=98.47%, F1 Score=97%).

REFERENCES:

1. https://ieeexplore.ieee.org/abstract/document/8123782
2. https://ieeexplore.ieee.org/abstract/document/8717766
3. https://www.kaggle.com/code/sabanasimbutt/anomaly-detection-using-unsupervised-techniques
4. https://fraud-detection-handbook.github.io/fraud-detection-handbook/Chapter_3_GettingStarted/SimulatedDataset.html
5. https://ieeexplore.ieee.org/abstract/document/8824930
6. https://link.springer.com/chapter/10.1007/978-981-16-6407-6_3

Team Details:

Reg. no	Name	Summary of tasks performed
2348527	M SURENDRAN	Implemented Support Vector Classfier, Random Forest Classifier,Isolation Forest And Documentation.
2348553	SANJAY S	Implemented Logistic Regression , Decision Tree Classifier & Documentation.
2348570	VICTOR JOSE I J	Preprocessed the dataset implemented one model k-nearest neighbors, documentation .