Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Credit card is the most popular mode of payment. As the number of credit card users is rising world-wide, the identity theft is increased, and frauds are also increasing. In the virtual card purchase, only the card information is required such as card number, expiration date, secure code, etc. Such purchases are normally done on the Internet or over telephone. To commit fraud in these types of purchases, a person simply needs to know the card details. The mode of payment for online purchase is mostly done by credit card. The details of credit card should be kept private.

To secure credit card privacy, the details should not be leaked. Different ways to steal credit card details are phishing websites, steal/lost credit cards, counterfeit credit cards, theft of card details, intercepted cards etc. For security purpose, the above things should be avoided. In online fraud, the transaction is made remotely and only the card's details are needed. The simple way to detect this type of fraud is to analyze the spending patterns on every card and to figure out any variation to the "usual" spending patterns. Fraud detection by analyzing the existing data purchase of cardholder is the best way to reduce the rate of successful credit card frauds.

As the data sets are not available and also the results are not disclosed to the public. The fraud cases should be detected from the available data sets knownas the logged data and user behavior. At present, fraud detection has been implemented by a number of methods such as data mining, statistics, and artificial intelligence.

1.1 PROBLEM STATEMENT

We aim to develop and implement an effective credit card fraud detection system using machine learning methodologies. The objective is to create a model capable of accurately identifying fraudulent transactions within a vast dataset, thereby minimizing financial losses for individuals and financial institutions, while ensuring the security and trust of credit card users. After the completion of the project, we will be able to detect whether the transaction is fraud or not.

1.2 OBJECTIVES

The main aim of this project is the detection of credit card fraudulent transactions, as it's important to figure out the fraudulent transactions so that customers don't get charged for the purchase of products that they didn't buy. The detection of the credit card fraudulent transactions will be performed with multiple Machine Learning techniques then comparison will be made between the outcomes and results of each technique to find the best and most suited model in the detection of the credit card transactions that are fraudulent . In addition, exploring previous literature and different techniques used to distinguish the fraud within a dataset.

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE REVIEW

Jain's team used several ML techniques to distinguish credit card fraud, three of them are SVM, ANN and KNN. Then to compare the outcome of each model, they calculated the true positive (TP), false negative (FN), false positive (FP), and true negative (TN) generated. ANN scored 99.71% accuracy, 99.68% precision, and 0.12% false alarm rate. SVM accuracy is 94.65%, 85.45% for the precision, and 5.2% false alarm rate. and finally, the accuracy of KNN is 97.15%, precision is 96.84% and the false alarm rate is 2.88% (Jain et al., 2019).

Gupta's team worked on implementing an automated model that uses various ML techniques to detect fraudulent instances that are related economically to users but is specializing more in credit card transactions, according to Gupta and his team Out of all the techniques that they used Naïve Bayes had an outstanding performance in distinguishing fraudulent transactions as the accuracy of it was 80.4% and the area under the curve is 96.3% (Gupta et al., 2021).

Adepoju and his team used all of the ML methods that are used in this paper, Logistic Regression , (SVM) Support Vector Machine, Naive Bayes, and (KNN) K-Nearest Neighbor, those methods were used on distorted credit card fraud data. The accuracies scored by all the models were 99.07% for Logistic Regression, Naïve Bayes scored 95.98%, 96.91% for K-nearest neighbor, and the last model (SVM) Support Vector Machine scored 97.53% (Adepoju et al., 2019).

Safa and Ganga investigated how well Logistic Regression, (KNN) K-nearest neighbor, and Naïve Bayes work on exceptionally distorted credit card dataset, they implanted their work on Python where the best method was selected using evaluation. The accuracies result of their model for Naïve Bayes is 83%, 97.69% for Logistic regression and in last

place K-nearest neighbor with 54.86% (Safa & Ganga, 2019).

The team of Tanouz proposed working on various ML based classification algorithms, likeNaïve Bayes, Logistic Regression, Random Forest, and Decision Tree in handling datasets that are strongly imbalanced, in addition their research will have the calculations of five measures the first is accuracy, the second is precision, the third is recall, the fourth is confusion matrix, and the last one is Roc-auc score. 95.16% is the score of both Logistic Regression and Naïve Bayes, 96.77% is the score for random forest, for the last model Decision Tree scored 91.12% (Tanouz et al., 2021).

Najdat and his team's approach in detecting fraudulent transactions is (BiLSTM) BiLSTMMaxPooling-BiGRU- MaxPooling, this approach is established upon bidirectional Long short-term memory in addition to (BiGRU) bidirectional Gated recurrent unit. In addition,the group decided to go for six ML classifiers, which are Voting, Adaboost, Random Forest, Decision Tree, Naïve bayes, and Logistic Regression. K-nearest neighbor scored an accuracy of 99.13%, and logistic regression scored 96.27%, Decision tree scored 96.40% and Naïve bayes scored 96.98% (Najadat et al., 2020).

The paper of Saheed and his group focuses on detection of Credit Card Fraud with the use of (GA) Genetic Algorithm as a feature selection technique. In feature selection the data is splitted in two parts first priority features and second priority features, and the ML techniques that the group used are The Naïve Bayes (NB), Random Forest (RF) and (SVM) Support Vector Machine. Naïve bayes scored 94.3%, SVM scored 96.3%, and Random Forest scored 96.40% which is the highest accuracy (Saheed et al., 2020).

The paper of Kiran and his team presents Naïve Bayes (NB) improved (KNN) K-Nearest Neighbor method for Fraud Detection of Credit Card which is (NBKNN) in short format.The outcome of the experiment illustrates the difference in the process of each classifier on the same dataset. Naïve bayes performed better than K-nearest neighbor as it scored an accuracy of 95% while KNN scored 90% (Kiran et al., 2018).

The paper of Bhanusri and his team implemented multiple ML techniques on an unbalanced dataset. The ML methods used are logistic regression, naïve bayes, and random forest to explain the relation of fraud and credit card. Their

conclusion of the project presents the best classifier by training and testing supervised techniques in term of their work. The logistic regression model scored 99.8% accuracy, random forest scored 100% and 90.8% is scored by naïve bayes.

The work of Itoo and his group uses three different ML methods the first is logistic regression, the second is Naïve bayes and the last one is K-nearest neighbors. Itoo and his group recorded the work and comparative analysis, their work is implemented on python. Logistic regression accuracy is 91.2%, Naïve bayes accuracy is 85.4% and Knearest neighbor is last with an accuracy of 66.9% (Itoo et al., 2020).

The team of Varmedja used multiple machine learning algorithms in their paper such as Logistic Regression, Multilayer Perception, Random Forest, and Naïve Bayes. As the dataset was quite very unbalanced Varmedja and his team SMOTE technique to oversample, feature selection, in addition to sectioning the data into a training section and a testing data section. The best scoring model during the experiment is Random Forest with 99.96%, with not many difference the model in second place is Multilayer Perceptron with 99.93%, in third place is Naïve bayes with 99.23% and in last place is Logistic regression with 97.46% (Varmedja et al., 2019).

The model used by Alenzi and Aljehane to detect fraud in credit cards was Logistic Regression, their model scored 97.2% in accuracy, 97% sensitivity and 2.8% Error Rate. A comparison was performed between their model and two other classifier which are 5 Voting Classifier and KNN. VC scored 90% in accuracy, 88% sensitivity and 10% error rate, as for KNN where k = 1:10, the accuracy of the model was 93%, the sensitivity 94% and 7% for the error rate (Alenzi & Aljehane, 2020).

2.2 LITERATURE REVIEW CONCLUSION

Throughout the search I found that there were many models created by other researchers which have proven that people have been trying to solve the credit card fraud problem. I found that Najdat Team used an approach that is established upon bidirectional long/short-term memory in building their model, other researchers have tried different data splitting ratios to generate different accuracies. The team of Sahin and Duman used different Support Vector Machine methods which are (SVM) Support Vector Machine with RBF, Polynomial, Sigmoid, and Linear Kernel. The lowest accuracy of the four models that will be studied in this research, is 54.86% for KNN and 36.40% for logistic Regression which were scored by Awoyemi and his team, as for Naïve Bayes the lowest accuracy was scored by Gupta and his team which is 80.4% and finally, SVM the lowest score was 94.65% and it was scored by Jain's team. To determine the best model out of the four models that will be studied through the research, the average of the best three accuracies of each model will be calculated, the average of the accuracy of KNN is 98.72%, the average of logistic regression is 98.11%, 98.85% for Naïve bayes and 96.16% for Support Vector Machine. So, for the best performing credit card fraud detecting model within the Literature review is the Logistic Regression model.

CHAPTER 3
METHODOLOGY

We aim to develop a system that will help the company reduce their financial losses and also help customers so they don't get charged for the purchase of products that they didn't buy. After the completion of project we will be able to detect fraud transactions, by finding the most effective machine learning Algorithm for finding fraud transactions. Five Algorithms Logistic Regression, Support Vector Machines, Decision Trees, K-Nearest Neighbor and Random Forest were compared with each other in under sampling and oversampling technique of handling imbalance data.

3.1 DATASET DESCRIPTION

The dataset contains transactions made by a cardholder in a duration in 2 days i.e., two days in the month of September 2013. Where there are total 284,807 transactions among which there are 492 i.e., 0.172% transactions are fraudulent transactions. This dataset is highly unbalanced. Since providing transaction details of a customer is considered to issue related to confidentiality, therefore most of the features in the dataset are transformed using principal component analysis (PCA). V1, V2, V3,..., V28 are PCA applied features and rest i.e., 'time', 'amount' and 'class' are non-PCA applied features as shown in table 1.

Table 1: Attributes of European Dataset

S.No.	Feature	Description
1	Time	Time in seconds to specify the elapses between the current transaction and first transaction.
2	Amount	Transaction amount
3	Class	0 - not fraud 1 – fraud

3.2 HANDLING IMBALANCE DATASET

The dataset used in this project is massive imbalance which can cause problems for model learning. In this project we use the two techniques to handle the sample imbalance Under sampling and oversampling.

3.2.1 UNDER SAMPLING

We start with the very simple approach called under sampling, just randomly draw the same number of samples from large number of pieces as equal to small number of samples. After this, we will able to generate new dataset called new_data and then we train the model. After using under sampling proportion of legitimate and fraudulent transactions was 50 percent and 50 percent and now new dataset contains 984 samples.

3.2.2 OVER SAMPLING

Oversampling means increasing the number of positive samples, making the positive and negative samples equal. Here, we will select 20,000 samples from negative samples (class 0) and increase the samples of the positive class to 20,000 from 492 by generating new instances using the SMOTE technique. Now our new dataset will be generated, which will contain 40,000 records, of which 20,000 are fraud and the remaining 20,000 are legitimate.

3.3 ALGORITHMS

3.3.1 LOGISTIC REGRESSION

Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict the probability of an observation belonging to one of two classes. Despite its name, it's primarily used for classification rather than regression.

Instead of predicting a discrete outcome directly, logistic regression calculates the probability of an observation belonging to a particular class using a logistic (sigmoid) function shown in fig.1. The model is trained by iteratively adjusting the coefficients to minimize the difference between predicted probabilities and actual class labels using techniques like maximum likelihood estimation or gradient descent.

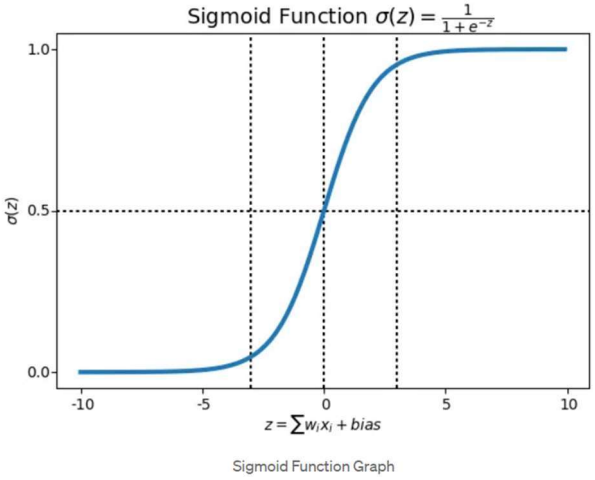


Fig.3.3.1 Sigmoid Function

3.3.2 SUPPORT VECTOR MACHINE

A Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. Its primary focus is on classification.

SVM is primarily used for classification, especially in scenarios where there's a clear margin of separation between classes. SVM aims to find the hyperplane that maximizes the margin, the distance between the hyperplane and the nearest data points of each class (these points are called support vectors). This margin allows for better generalization and helps in reducing overfitting.Initially designed for linear classification, SVM uses a linear hyperplane to separate classes. However, through the kernel trick, it can handle nonlinear relationships by mapping data into higher-dimensional spaces where classes become separable by hyperplanes.

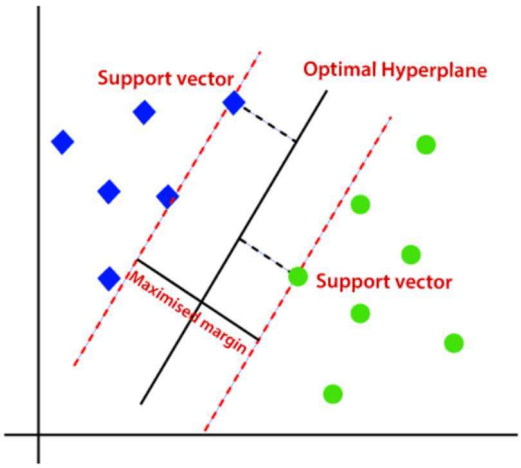


Fig. 3.3.2 Support Vector Machine

3.3.3 DECISION TREE

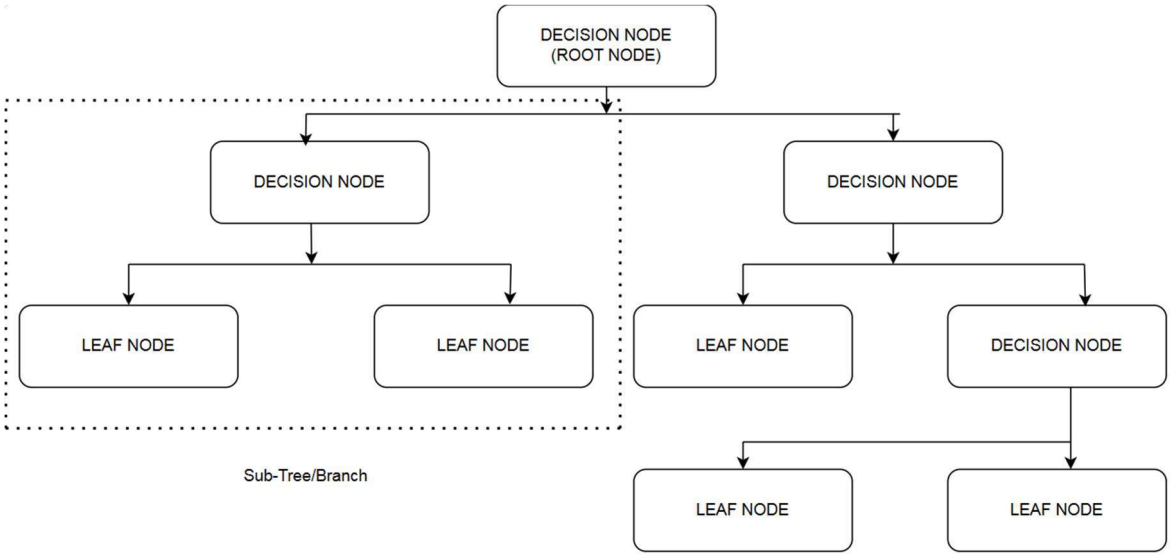
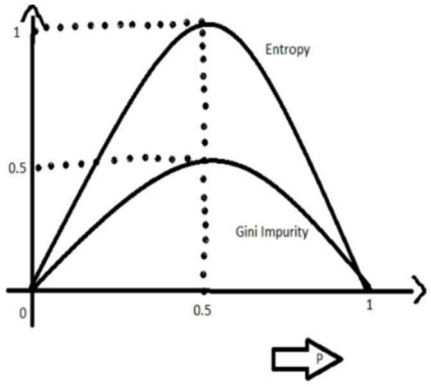


Fig.3.3.3.1 Decision Tree

Decision trees are hierarchical structures used for classification and regression tasks. They make decisions by splitting data based on features, optimizing criteria like Gini impurity (for classification) or mean squared error (for regression). These trees create branches and leaf nodes, representing decisions and outcomes, offering interpretability due to their transparent decision-making process. Prone to over fitting, techniques like pruning or limiting tree depth help prevent excessive complexity. Ensemble methods like Random Forests combine multiple trees for improved performance, making decision trees widely used for their simplicity and interpretive qualities. Performance is assessed using metrics like accuracy or mean squared error, depending on the task.



$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

$$Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$$

Fig.3.3.3.2 Entropy and Gini Impurity

3.3.4 K-NEAREST NEIGHBOR

K-Nearest Neighbors (KNN) is a simple yet effective algorithm used for classification and regression tasks. It operates by assigning a new data point's label or value based on the majority vote or average of its K nearest neighbors in the feature space. KNN doesn't involve explicit training but stores all available data for prediction, making it memory-intensive for large datasets.The choice of K impacts model performance, with smaller K values leading to more flexible and potentially noisy predictions, while larger K values offer smoother but less detailed outcomes. KNN's simplicity and flexibility make it useful for various applications, yet its computational demand increases significantly with dataset size. Performance assessment typically involves metrics like accuracy or mean squared error, based on the task at hand.

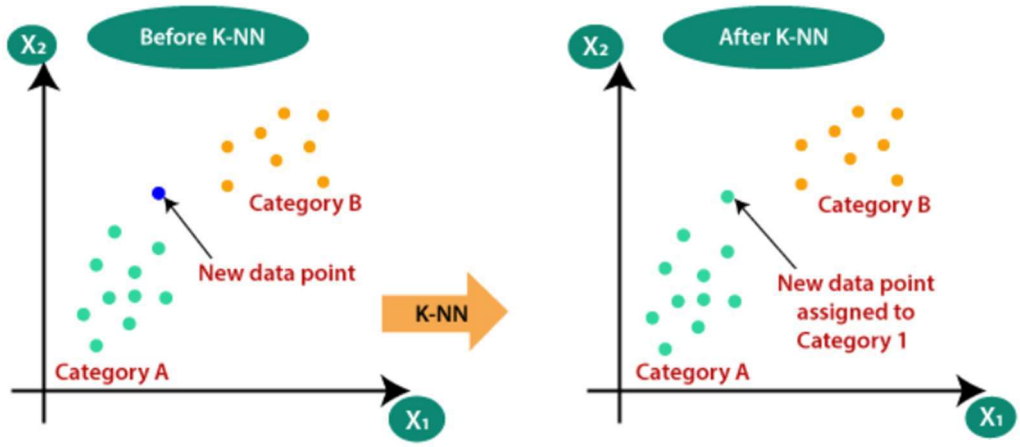


Fig.3.3.4 K-Nearest Neighbor

3.3.5 RANDOM FOREST

Random Forest is an ensemble learning method used for classification and regression tasks that operates by constructing multiple decision trees during training and outputting the mode of the classes for classification or the mean prediction for regression. Each tree in the forest is built from a random subset of the training data and a random subset of features, which helps in reducing overfitting and improving generalization. This randomness ensures that the individual trees are not highly correlated, thereby enhancing the overall model's robustness and accuracy. Random Forest handles a large number of input features and can capture complex interactions among them, making it particularly effective in scenarios with diverse and noisy data. It also provides a measure of feature importance, which can be useful for

understanding the relative significance of different predictors in the dataset. This method is widely appreciated for its versatility, high accuracy, and resistance to overfitting, making it a popular choice for tasks like credit card fraud detection, where distinguishing between fraudulent and legitimate transactions is critical.

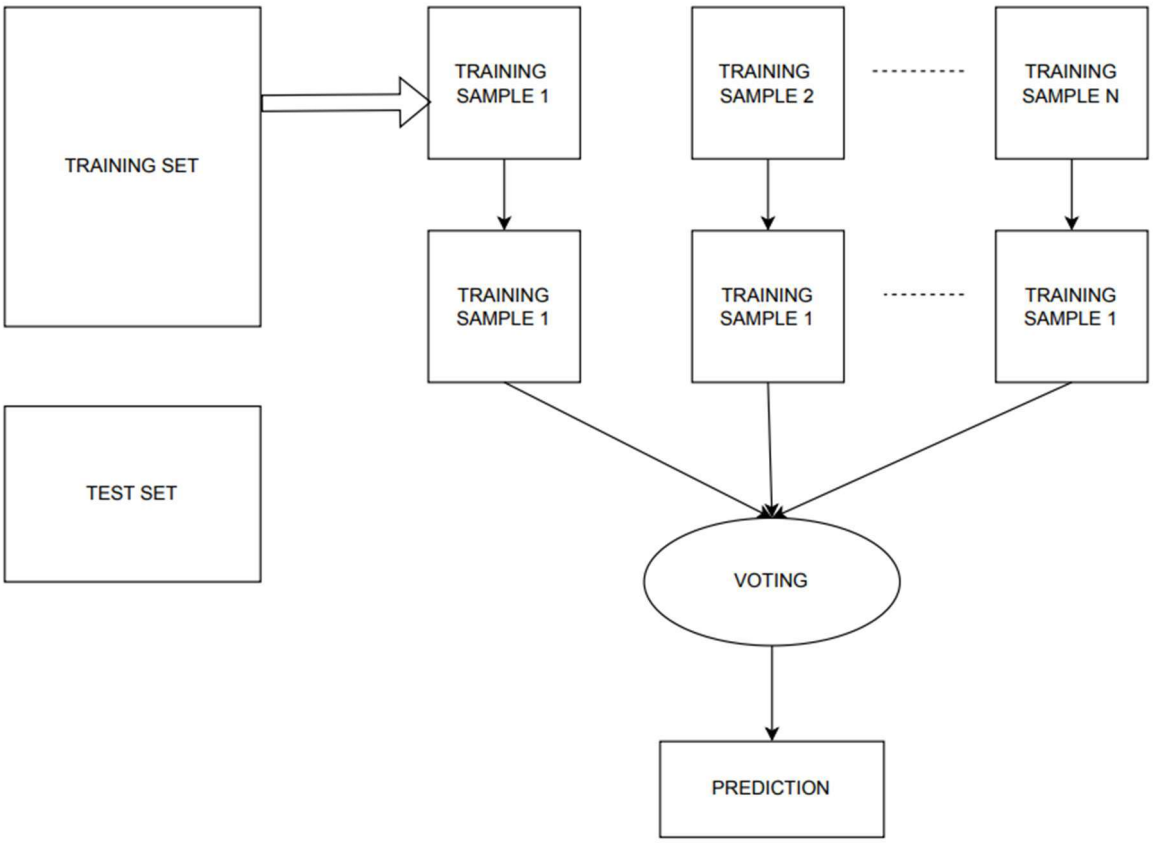


Fig.3.3.5 Random Forest

3.4 SYSTEM ARCHITECTURE

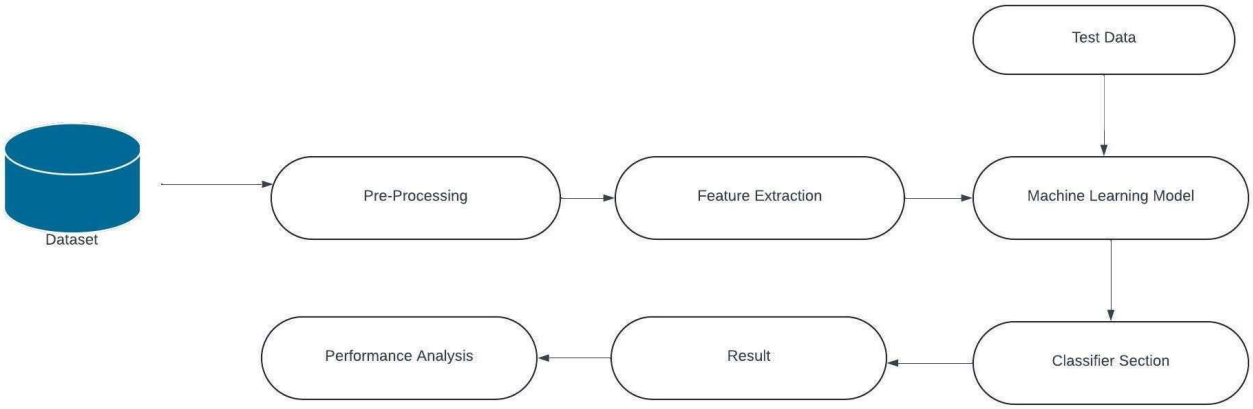


Fig.3.4 System Architecture

We start by reading dataset of credit card transaction in our project. Then we apply preprocessing techniques and do some analysis of our data .After this we handles imbalance dataset. We have created model by applying five different algorithm, we will select the best one.

3.5 PLATFROM USED

MICROSOFT WORD:

Microsoft Word is a widely-used word processing software developed by Microsoft, offering a user-friendly interface for creating, editing, and formatting documents. It provides extensive text formatting options, built-in templates, and collaboration tools for real-time co-authoring. Integration with other Microsoft Office applications and cross-platform compatibility make it a versatile tool for document creation.

GOOGLE COLAB:

Google Colab is a cloud-based platform offering free access to Jupyter note- books and computational resources like GPUs and TPUs. It allows collaborative coding in Python, supporting libraries like TensorFlow and PyTorch for machine learning and data analysis. With no setup required, users can write and execute code, import datasets from Google Drive, and share notebooks effortlessly. Colab provides a convenient environment for learning, experimenting, and deploying machine learning models, making it popular among researchers, students, and developers. Session limitations, occasional resource constraints, and data privacy considerations are factors to consider while using Colab.

3.6 SYSTEM DEVELOPMENT

1. Modules used:

   (a) numpy - used for concatenating , dropping columns and performing other mathematical operations

   (b) pandas - Provides data structures like Data Frames for handling structured data and powerful tools for data cleaning, transformation, and exploration.

   (c) Scikit-learn - Scikit-learn is a machine learning library providing tools for various machine learning tasks, including classification, regression, clustering, and model evaluation.

(d) Matplotlib - It is a comprehensive library for creating static, animated, and interactive visualizations in Python. It's particularly useful for generating plots, charts, and figures for data analysis and presentation.

3.7 WORKFLOW OF THE PROJECT

1. We start by including our dataset in our project.

2. Data Analysis was done, and we find out that data is highly imbalanced.

3. After data analysis, we did data pre-processing which includes handling missing values, duplicate rows etc.

4. Created under sampled dataset(984 samples) by taking random sample from majority class and Created one more dataset using oversampling (40,000 samples) by generating new samples in positive class .Choosing 20,000 records from each class.

5. Divided the dataset into training and testing. 80% of the data was used for training and 20% was used for testing in both the dataset created by applying under sampling and oversampling.

6. Selected relevant features by finding correlation of every attribute with every other attribute to enhance performance and reduce dimensions.

7. Applied Logistic Regression, Support Vector Machines, Decision Tress, K-Nearest Neighbour and Random Forest in both the dataset.

8. Evaluate the performance of each algorithm by predicting outcomes on the test data from both oversampled and under sampled sets.

9. Compare the predictions of each algorithm against the test data, calculating accuracy, precision, recall etc. for each algorithm on both datasets.

10. Analyse the results to determine the most effective algorithm in terms of accuracy, precision, recall, and overall performance on both oversampled and under sampled datasets.

CHAPTER 4

Evaluation Metrics and Result

4.1 EVALUATION METRICES

Confusion matrix

A confusion matrix is a table that summarizes the performance of a classification model by showing the actual versus predicted classifications. It includes:

- True Positives (TP): Correctly predicted fraudulent transactions.
- True Negatives (TN): Correctly predicted legitimate transactions.
- False Positives (FP): Legitimate transactions incorrectly predicted as fraudulent.
- False Negatives (FN): Fraudulent transactions incorrectly predicted as legitimate.

The confusion matrix provides a comprehensive view of the model's performance and helps in understanding the types of errors made.

Accuracy

Accuracy is the ratio of correctly predicted instances to the total instances. It is calculated as:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives. While accuracy is a useful measure, it can be misleading in imbalanced datasets like credit card fraud detection, where the number of legitimate transactions far exceeds fraudulent ones.

Precision

Precision, also known as positive predictive value, measures the proportion of true positive predictions out of all positive predictions. It is calculated as:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$

Precision is crucial when the cost of false positives is high, such as in fraud detection, where falsely identifying a legitimate transaction as fraudulent can cause inconvenience to

customers.

Recall

Recall, or sensitivity, measures the proportion of actual positives correctly identified. It is calculated as:

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall is important in fraud detection because it indicates how effectively the model identifies actual fraudulent transactions, minimizing the number of missed fraud cases.

F1-Score

The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is calculated as:

$$F1 - Score = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score is particularly useful when dealing with imbalanced datasets, offering a better measure of a model's performance than accuracy alone.

Using these metrics in your evaluation ensures a thorough assessment of your model's performance in detecting credit card fraud, highlighting both its strengths and areas for improvement.

4.2 RESULTS

Table2:Results of Under Sampling and over sampling

ALGORITHMS	Under sampling			Over sampling		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
LOGISTIC REGRESSION	0.93	0 – 1.00 1 – 0.87	0 – 0.88 1 – 1.00	0.97	0 – 0.98 1 – 0.96	0 – 0.96 1 – 0.98
SUPPORT VECTOR MACHINE	0.88	0 – 1.00 1 – 0.77	0 – 0.81 1 – 1.00	0.95	0 – 0.99 1 – 0.91	0 – 0.92 1 – 0.99
DECISION TREE	0.90	0 – 0.91 1 – 0.91	0 – 0.91 1 – 0.91	0.99	0 – 0.99 1 – 0.99	0 – 0.99 1 – 0.99
K-NEAREST NEIGHBOR	0.64	0 – 0.66 1 – 0.63	0 – 0.64 1 – 0.65	0.87	0 – 0.85 1 – 0.91	0 – 0.92 1 – 0.83
RANDOM FOREST	0.92	0– 0.88 1 – 1.00	0 – 1.00 1 – 0.86	0.99	0 – 0.99 1 – 1.00	0 – 1.00 1 – 0.99

From the results, it is evident that Logistic Regression and Random Forest performed best under the undersampling strategy, achieving accuracies of 93% and 92% respectively. These models were able to maintain robust performance despite the reduction in sample size, demonstrating their effectiveness in handling imbalanced datasets through undersampling.

In contrast, the oversampling approach significantly boosted the performance of the Decision Tree and Random Forest models, both achieving an impressive accuracy of 99%. This indicates that these models benefit greatly from having a balanced dataset with more samples, allowing them to learn more effectively from the data.

Overall, while Logistic Regression and Random Forest are reliable choices for undersampled data, Decision Tree and Random Forest excel when oversampling is applied, making them the preferred models in scenarios where resampling techniques are used to mitigate class imbalance.

CHAPTER 5
CONCLUSION

I've successfully developed a system that, given sufficient time and data, approaches our intended goal closely. I have created a system that can help in financial and banking sector to enhance security , cost savings - By minimizing fraudulent activities, companies can save substantial amounts of money that would otherwise be lost to fraud cases and reimbursements , Improved Customer Trust: Efficient fraud detection ensures that customers feel secure about using their credit cards, fostering trust and loyalty towards the company's services, Real time detection , Adaptability and Learning , Automated processes . It will also help the Credit Card users to Reduced Financial Losses: By promptly identifying fraudulent activities, users are less likely to suffer financial losses due to fraudulent transactions, Improved trust , Enhance security.

I have used dataset of European Cardholders of 2013 which contains data of transaction held in two consecutive days , Most of the attributed were hidden due to security purpose , but some important attribute like amount , time was given. This dataset was very imbalanced so handle it i utilized two techniques Under sampling and Oversampling .

Firstly I applied under sampling by taking 492 records from both the classes , and applied five Algorithms which are Logistic Regression , Support Vector Machine(SVM) , K-Nearest Neighbor(KNN),Decision Tree , Random Forest and then I concluded that Logistic Regression and Random Forest gives very well result with accuracy of 93%. After this I used oversampling by taking 20,000 records from each class (created new instance of positive class) , and then applied same five algorithms Logistic Regression , Support Vector Machine(SVM) , K-Nearest Neighbor(KNN),Decision Tree, Random Forest and then I concluded that Decision Tree and Random Forest performs very well by giving accuracy of 99%.

CHAPTER 6

FUTURE WORK

From the above analysis, it is clear that many machine learning algorithms are used to detect the fraud but we can observe that the results are not satisfactory. So, we would like to implement deep learning algorithms to detect credit card fraud accurately.As with any such project, there is some room for improvement here. This project can be improved by integrating various machine learning algorithms together. We can also use multiple other algorithms for our project. We can also do some improvement in the dataset as we have concluded that accuracy is high when we have more data then surely we will able to reduce fraud transaction and reduce false - positive. However, it will require some official supports from the banks also.

REFERENCES

[1] A. Gupta, M. Lohani, and M. Manchanda, "Financial fraud detection using naive Bayes algorithm in highly imbalance data set," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 5, pp. 1559–1572, 2021.

[2] F. Itoo, Meenakshi, and S. Singh, "Comparison and analysis of logistic regression, naïve bayes and knn machine learning algorithms for credit card fraud detection," *International Journal of Information Technology*, vol. 13, pp. 1503–1511, 2021.

[3] D. Tanouz, R. R. Subramanian, D. Eswar, G. P. Reddy, A. R. Kumar, and C. V. Praneeth, "Credit card fraud detection using machine learning," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2021, pp. 967–972.

[4] H. Z. Alenzi and N. O. Aljehane, "Fraud detection in credit cards using logistic regression," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, 2020.

[5] H. Najadat, O. Altiti, A. A. Aqouleh, and M. Younes, "Credit card fraud detection based on machine and deep learning," in *2020 11th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2020, pp. 204–208.

[6] Y. K. Saheed, M. A. Hambali, M. O. Arowolo, and Y. A. Olasupo, "Application of ga feature selection on naive bayes, random forest and svm for credit card fraud detection," in *2020 international conference on decision aid sciences and application (DASA)*. IEEE, 2020, pp. 1091–1097.

[7] A. Bhanusri, K. R. S. Valli, P. Jyothi, G. V. Sai, and R. Rohith, "Credit card fraud detection using machine learning algorithms," *Journal of Research in Humanities and Social Science*, vol. 8, no. 2, pp. 4–11, 2020.

[8] R. Sailusha, V. Gnaneswar, R. Ramesh, and G. R. Rao, "Credit card fraud detection using machine learning," *2020 4th International Conference on Intelligent*

Computing and Control Systems (ICICCS), pp. 1264–1270, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:219990550

[9] Y. Jain, N. Tiwari, S. Dubey, and S. Jain, "A comparative analysis of various credit card fraud detection techniques," *International Journal of Recent Technology and Engineering*, vol. 7, no. 5, pp. 402–407, 2019.

[10] O. Adepoju, J. Wosowei, H. Jaiman *et al.*, "Comparative evaluation of credit card fraud detection using machine learning techniques," in *2019 Global Conference for Advancement in Technology (GCAT)*. IEEE, 2019, pp. 1–6.

[11] M. U. Safa and R. Ganga, "Credit card fraud detection using machine learning," *Interna- tional Journal of Research in Engineering, Science and Management*, vol. 2, no. 11, pp. 372–374, 2019.

[12] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection-machine learning methods," in *2019 18th International Symposium INFOTEH- JAHORINA (INFOTEH)*. IEEE, 2019, pp. 1–5.

[13] D. Dighe, S. Patil, and S. Kokate, "Detection of credit card fraud transactions using machine learning algorithms and neural networks: A comparative study," in *2018 Fourth Interna- tional Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018, pp. 1–6.

[14] S. Kiran, J. Guru, R. Kumar, N. Kumar, D. Katariya, and M. Sharma, "Credit card fraud detection using naïve baiyes model based and knn classifier," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 3, p. 44, 2018.

[15] S. Maniraj, A. Saini, S. Ahmed, and S. Sarkar, "Credit card fraud detection using machine learning and data science," *International Journal of Engineering Research*, vol. 8, no. 9,pp. 110–115, 2019.