Hacettepe University

Computer Engineering Department

BBM479/480 End of Project Report

Project Details

Title	Credit Card Fraud Detection Using ML Techniques & Comparing
Supervisor	Murat Aydos

Group Members

	Full Name	**Student ID**
1	İlkim İclal Aydoğan	21992814
2	Diren Boran Sezen	21946553
3	Hasan Çağrı Sarıkaya	2200356852
4		

Abstract of the Project (/ 10 Points)

Explain the whole project shortly including the introduction of the field, the problem statement, your proposed solution and the methods you applied, your results and their discussion, expected impact and possible future directions. The abstract should be between 250-500 words.

Credit card fraud is a serious issue affecting millions worldwide and poses significant financial risks to individuals, businesses, and the global economy. Fraudulent activities range from unauthorized transactions and identity theft to card counterfeiting and account takeover. Detecting fraudulent transactions is crucial for financial institutions to protect their customers and minimize losses. As fraudsters continually adapt their tactics, traditional rule-based fraud detection systems often struggle to keep pace, necessitating the exploration of more sophisticated and adaptive solutions. Machine learning, with its ability to learn patterns from vast amounts of data, has emerged as a promising avenue for enhancing fraud detection capabilities. This project investigates the application of various machine learning models, including both supervised and unsupervised techniques, to identify fraudulent credit card transactions effectively.

We leveraged two datasets: a publicly available dataset featuring real-world credit card transactions and a simulated dataset enriched with comprehensive features from IBM's TabFormer repository. The latter offered a unique opportunity to incorporate user demographics, transaction details, location data, and potential error flags into our analysis.

We addressed the inherent class imbalance in fraud detection by employing undersampling, oversampling (SMOTE), and a combination of both techniques. This allowed for a more balanced representation of fraudulent and legitimate transactions during model training. We then trained and evaluated a variety of machine learning models, including logistic regression, k-nearest neighbors, support vector machines, random forest, and decision trees on both datasets.

We explored the potential of anomaly detection techniques like One-Class SVM and Isolation Forest, which showed promise in identifying unusual transaction patterns that may not be captured by traditional classification models.

We also explored neural network implementations for a better understanding and see if deep learning models have a better performance in detecting fraudulent cases then ML models.

The insights gained from this project underscore the effectiveness of combining diverse machine learning techniques and emphasize the importance of selecting the right model for the problem. Future research could focus on exploring more sophisticated deep learning architectures, developing real-time fraud detection systems, and investigating explainable AI models to enhance transparency and trust in fraud detection decisions.

Introduction, Problem Definition & Literature Review (/ 20 Points)

Introduce the field of your project, define your problem (as clearly as possible), review the literature (cite the papers) by explaining the proposed solutions to this problem together with limitations of these problems, lastly write your hypothesis (or research question) and summarize your proposed solution in a paragraph. Please use a scientific language (you may assume the style from the studies you cited in your literature review). You may borrow parts from your previous reports but update them with the information you obtained during the course of the project. This section should be between 750-1500 words.

The rapid advancement of digital payments has brought convenience to consumers but also increased the vulnerability to fraudulent activities, especially in credit card transactions. Fraud detection has thus become a critical area of focus in the financial sector. Machine learning (ML) offers promising techniques to analyze transaction data and identify potential fraudulent activities, aiming to protect consumers and financial institutions from significant financial losses. This project seeks to evaluate the efficiency of various machine learning algorithms—Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest Classifier (RFC), and Decision Tree Classifier (DTC)—in detecting credit card fraud. The study further extends to implementing anomaly detection techniques and neural network architectures to test their potential in enhancing fraud detection systems.

The primary problem addressed in this project is the detection and prevention of fraudulent credit card transactions using machine learning techniques. Given the vast number of daily transactions, it is challenging to manually identify fraud. Effective fraud detection models need to accurately classify transactions as fraudulent or legitimate, minimizing both false positives (legitimate transactions flagged as fraud) and false negatives (fraudulent transactions not detected). The goal is to evaluate and compare the performance of various machine learning algorithms in achieving high accuracy and efficiency in fraud detection.

Numerous studies have explored the application of machine learning techniques in credit card fraud detection. Here, we review some of the significant contributions and their limitations.

Logistic Regression is a statistical method widely used for binary classification problems, including fraud detection. It models the probability of a transaction being fraudulent based on a set of input features. *Itoo & Singh (2021)* performed random undersampling to address the imbalanced dataset and concluded that Logistic Regression (LR) showed optimal performance across all data proportions compared to Naive Bayes (NB) and K-Nearest Neighbors (KNN). The LR model demonstrated better sensitivity, specificity, precision, and F-measure than the NB and KNN techniques. *Alenzi & Aljehane (2020)* propose a logistic regression-based classifier for detecting credit card fraud, addressing issues of dirty data by employing mean-based and clustering-based cleaning methods. Using cross-validation (folds=10), the classifier achieved superior performance with an accuracy of 97.2%, sensitivity of 97%, and an error rate of 2.8%, outperforming K-nearest neighbors and voting classifiers.

K-Nearest Neighbors (KNN) is a non-parametric algorithm that classifies transactions based on their proximity to other transactions in the feature space. *Manlangit, Azam & Shanmugam (2019)*, propose a KNN model with a synthetic minority oversampling technique (SMOTE). Results reveal that the proposed model performed well. The KNN model achieves a precision of 98.32% and 97.44. Some papers we cite here like *Alenzi & Aljehane (2020)* also use KNN but the performance is not optimal.

Support Vector Machines (SVM) are effective in high-dimensional spaces and have been used for fraud detection. There are a lot of paper that tries to optimize SVM by utilizing different techniques for parameter optimization and better performance like *Li et al. (2021)* or *Pavithra & Thangadurai (2019)*. *Pavithra & Thangadurai (2019)* have a good precision score of 0.963 and an accuracy of %96.3 but we tried traditional SVM in our project because of implementation reasons.

Random Forest Classifier (RFC) is an ensemble learning method that operates by constructing multiple decision trees. According to *Marazqah Btoush et al. (2023)*, RFC is the most widely used method for detecting credit card fraud. *Amusan et al. (2021)* demonstrated the effectiveness of RFC in fraud detection on skewed data, achieving the highest accuracy (95.19%) compared to K-Nearest Neighbors (KNN), Logistic Regression (LR), and Decision Trees (DT). Additionally, *Ata and Hazim (2020)* applied RFC alongside other techniques such as Support Vector Machines (SVM), Naive Bayes (NB), and KNN. Their findings indicated that the RFC consistently outperformed the other methods across various dataset ratios.

The Decision Tree Classifier (DTC) is widely recognized for its simplicity and interpretability. While numerous studies leverage Decision Trees, they often incorporate additional techniques to enhance performance. For instance, *Choubey and Gautam (2020)* combined supervised algorithms such as Decision Trees (DT), Random Forests (RF), Logistic Regression (LR), Naive Bayes (NB), and K-Nearest Neighbors (KNN). Their research found that a hybrid classifier combining DT and KNN outperformed any individual classifier. Similarly, *Hammed and Soyemi (2020)* described the use of the DT algorithm enhanced with regression analysis, achieving improved performance with a misclassification rate of 18.4%. However, for our assignment, we chose to employ a straightforward Decision Tree without integrating other models or techniques. By using a simple, "vanilla" Decision Tree, we aimed to maintain the model's inherent interpretability and simplicity while demonstrating its standalone effectiveness.

Anomaly Detection

Anomaly detection techniques aim to identify outliers in the transaction data that deviate from normal patterns, often indicating fraud. Isolation forest is one of the many algorithms for anomaly detection. The Isolation Forest algorithm identifies anomalies by isolating data points through random partitioning, where anomalies are isolated more quickly due to their distinct characteristics. This method relies on the principle that anomalies are few and different, making them easier to isolate in fewer steps compared to normal data points. According to *Marazqah Btoush et al. (2023)*, it is not as popular as other ML models for fraud detection even though the nature of the models is to detect outliners. In *Meenu et al. (2020)* they do not use any data sampling methods and use the same dataset as ours. The Isolation Forest model they propose is evaluated by AUC and the efficiency is to be said 98.72%.

Neural networks, particularly deep learning models, have shown promise in capturing complex patterns in large datasets. There are multiple papers that implements different deep learning models such as CNN and ATTENTION like *Agarwal et al. (2021)*. But we wanted to keep it simple again and use ANNs for training and evaluation. In the ANN aspect, there are several articles such as *Asha & Suresh Kumar (2021)*. They conclude that ANN gives better accuracy than SVM and KNN but worse precision an recall.

Limitations

Despite the advancements, each method has limitations. Logistic Regression and Decision Trees may struggle with complex data patterns. KNN and SVM can be computationally intensive and challenging to scale. Random Forests require significant computational resources, and deep learning models, while powerful, demand large datasets and high computational costs.

The primary research question guiding this study is: “Which machine learning algorithm among Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest Classifier, and Decision Tree Classifier is most effective in detecting fraudulent credit card transactions based on accuracy, precision, recall, and F1-score? Additionally, how do other methods, such as anomaly detection and neural networks, compare to these traditional machine learning models in terms of performance?” By addressing this question, we aim to identify the strengths and weaknesses of each method and determine the most suitable approach for fraud detection.

Proposed Solution

The proposed solution involves a comprehensive evaluation of different machine learning models, including Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest Classifier, and Decision Tree Classifier, on a benchmark credit card fraud dataset. The performance of these models will be compared based on accuracy, precision, recall, and F1-score. Subsequently, we will implement anomaly detection techniques and neural network architectures, to assess their efficacy in fraud detection. The models also will be trained and tested on a different dataset to evaluate their generalizability and robustness. The expected outcome is to identify the most effective model or combination of models for fraud detection in credit card transactions, thereby enhancing the security and reliability of financial systems.

Methodology (/ 25 Points)

Explain the methodology you followed throughout the project in technical terms including datasets, data pre-processing and featurization (if relevant), computational models/algorithms you used or developed, system training/testing (if relevant), principles of model evaluation (not the results). Using equations, flow charts, etc. are encouraged. Use sub-headings for each topic. Please use a scientific language. You may borrow parts from your previous reports but update them with the information you obtained during the course of the project. This section should be between 1000-1500 words (add pages if necessary).

This section details the technical approaches and procedures undertaken to develop a system for credit card fraud detection. Leveraging both publicly available dataset (credit card transactions dataset from Kaggle) and a simulated dataset from IBM TabFormer.

1. **Datasets:**

    - **Credit Card Fraud Detection Dataset(Kaggle dataset)**: This dataset, sourced from Kaggle, contains transactions made by credit cards in September 2013 by European Card holders. Dataset presents 284,807 credit card transactions labeled as either fraudulent (492 instances) or legitimate. Features consist of 28 principle components obtained from PCA transformation, along with Time (seconds elapsed since first transaction) and Amount (transaction amount).

    - **Simulated Credit Card Transaction Dataset (IBM TabFormer dataset)**: This Dataset provided by IBM's TabFormer repository, simulates credit card transactions with features such as user demographics, card details, merchant informations, transactions specifics(amount, time, location), and potential transaction errors. It includes 24,386,900 transactions, 29,757 are fraudulent.

    - **United States ZipCodes Database:** This data set from "https://www.unitedstateszipcodes.org/zip-code-database/" contains 42,735 ZIP codes from the United States with a lot of features. We used this dataset's latitude, longitude coordinates, and estimated population features for each ZIP code to enrich our data.

2. **Data Preprocessing And Feature Engineering:**

    - **Credit Card Fraud Detection Dataset**: The Amount and Time features scaled using RobustScaler to handle effectively. Other features from V1 to V28 contain only numerical inputs results from a PCA transformation and in the public dataset didn't contain original values because of confidentiality issues.

    - **Simulated Credit Card Transaction Dataset:**
        - For categorical features "Use Chip" and "Errors" we used One-hot encoding. Figures below are for use chip types - fraudlines rates and error types - fraudlines rates

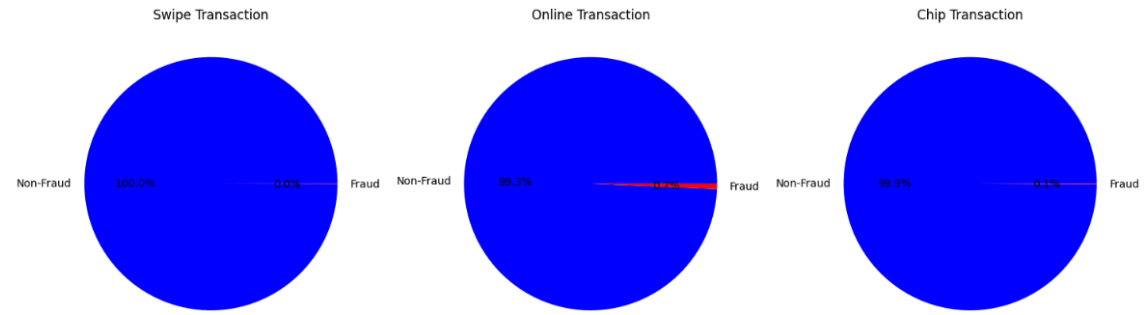


Fig1.

*Figure 1 shows types of “Use Chip” feature in the Simulated Credit Card Transaction Dataset. The highest fraudlines rate (0.7% fraud) is in Online Transaction type.It is followed by Chip transaction with rate of 0.1% fraud. Finally, the Swipe transaction’s fraud rate is 0.0%.

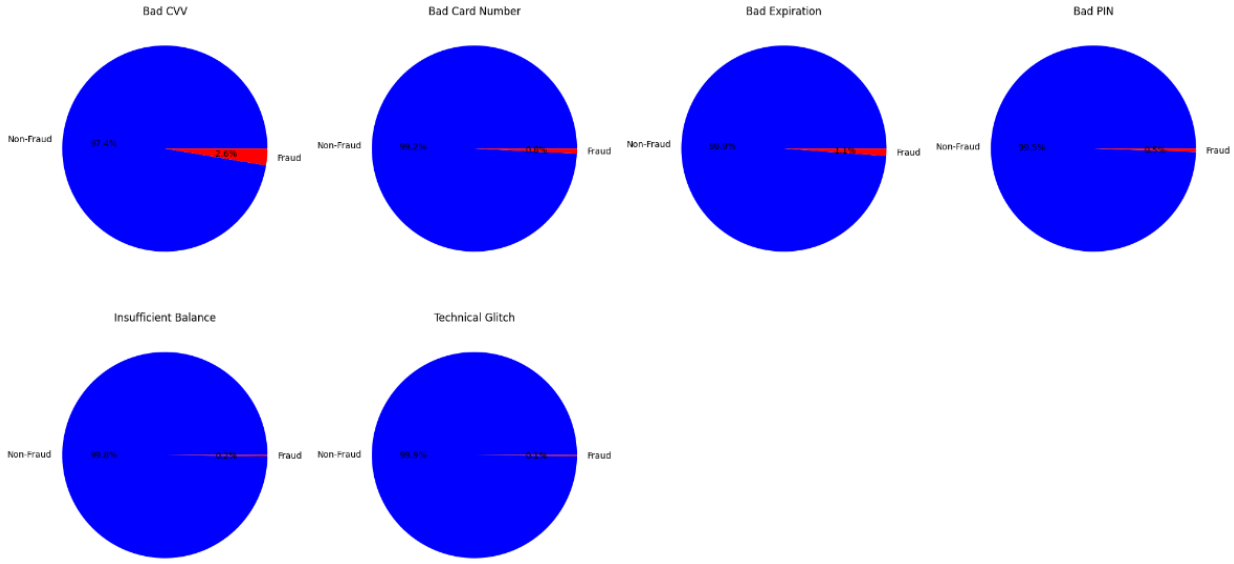
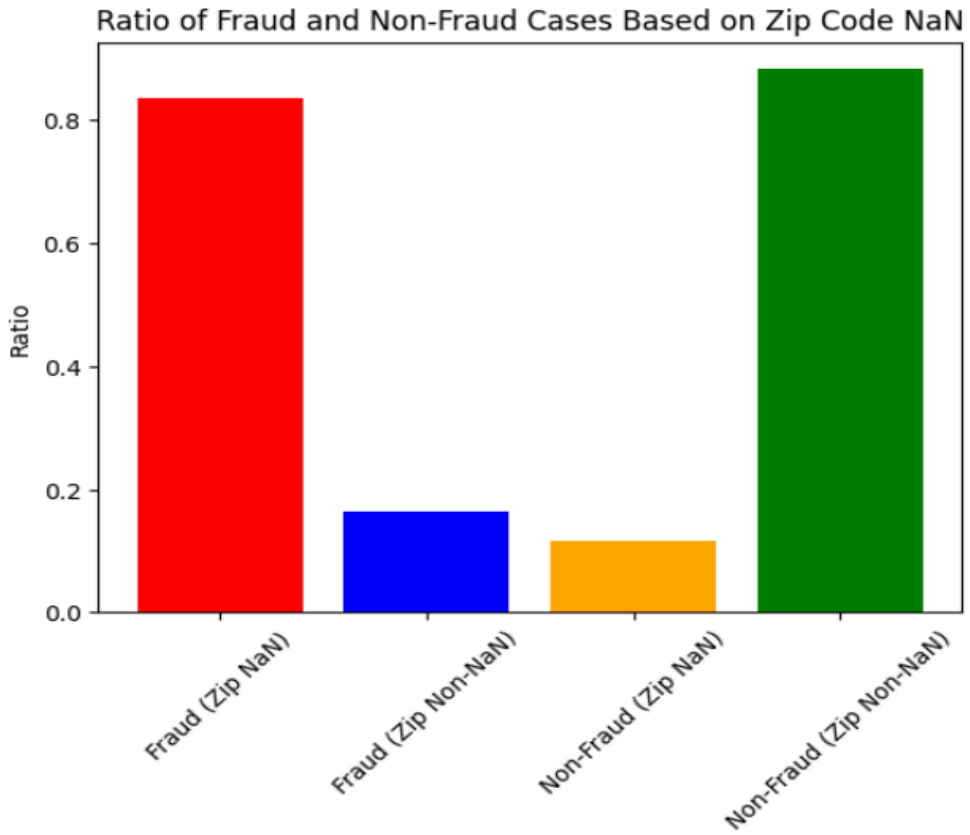


Fig.2

*Figure 2 shows the types of Errors and the proportions of fraud and non-fraud cases. The highest fraud rate (2.6% fraud) is in Bad CCV. It is followed by Bad Expiration (1.1% fraud). And Bad Card Number (0.7% fraud).

- Combined features “user” and “card” to user_card and combined features “Merchant Name” and “Merchant State” to merchant_location,
- Merging external ZIP code data to derive latitude, longitude and estimated_population.
- Using these location coordinates we calculated the distance of each transaction from New york city.
- Because there is a high correlation between missing zip code and fraud rate we did not drop nan values in these fields.



- We Converted Time features into timestamps. And Scaled features “Amount”, “Distance” and “estimated population” using RobustScaler.

The simulated credit card transactions dataset contains several high-cardinality categorical features, such as merchant_id, Zip, MCC, merchant_location, and user_card. These features have a large number of unique values, making them challenging to handle directly in machine learning models.

To process these features we decided to use embedding instead of Frequency Encoding or Target Encoding because:

- **Frequency Encoding:** While simple to implement, frequency encoding is sensitive to outliers. Rare categories may have extreme frequencies that don't reflect their true importance. Additionally, it doesn't capture any relationships between categories, only their individual frequencies

.

- **Target Encoding:** This method can be prone to overfitting, especially with smaller datasets. It requires careful cross-validation to avoid data leakage, where information from the validation or test sets leaks into the training process, leading to overly optimistic performance estimates.

Embedding, on the other hand, offers several advantages:

- **Captures Relationships:** Embeddings can learn complex relationships between categories, representing them in a continuous space where similar categories are closer together.

- **Handles High Cardinality:** Embeddings are well-suited for high-cardinality features, as they map each category to a dense, low-dimensional vector, avoiding the dimensionality associated with one-hot encoding.

In this project, an embedding layer is used to learn a 10-dimensional embedding for each of the high-cardinality categorical features. These embeddings are then concatenated with the other features and used as input to the machine learning models

- **Data Sampling:** To address the class imbalance, we experimented with two sampling techniques:
  Undersampling(One sided selection): This technique selectively removes majority class samples (legitimate transactions) to balance the class distribution.
  Oversampling (SMOTE - Synthetic Minority Over-sampling Technique):SMOTE generates synthetic minority class samples (fraudulent transactions) by interpolating between existing minority samples.

  Four versions of training sets were created:
  - Original: The Dataset without any sampling.
  - Undersampled: The dataset after applying One side selection.
  - Oversampled (SMOTE): The dataset after applying Smote.
  - Undersampled then oversampled (SMOTE): the dataset after applying both undersampling and oversampling..

3. **Model Training and Evaluation**
   - **Supervised Learning Algorithm Models:** Various machine learning models were trained and evaluated on both datasets.

     - **Logistic Regression (LR):** A linear model for binary classification.
     - **K-nearest Neighbors (KNN):** A non-parametric model that classifies based on the proximity of data points.
     - **Support Vector Machines (SVM):** A model that finds a hyperplane to separate classes.
     - **Random Forest Classifier (RFC):** An ensemble model that combines multiple decision trees.
     - **Decision Tree Classifier (DTC):** A tree based model that makes decisions based on feature values.

   For each model and dataset combination we performed the following:

   1. **Model Training:** The model was trained on a training set.
   2. **Model Evaluation:** Trained model evaluated on the test using the following metrics:
      - Precision: the fraction of correctly predicted positive observations out of all positive observations.
      - Recall: The fraction of correctly predicted positive observations out of all actual positive observations.
      - F1 Score: The harmonic mean of precision and recall, providing a balanced measure of model performance.
      - AUC-ROC Score: The area under the Receiver Operation Characteristic curve, measuring model's ability to distinguish between classes.

- Cross Validation: 5-fold cross validation was performed on the training set to assess model generalization.

4. **Neural Network(ANN):**

In addition to the traditional machine learning models, we implemented an Artificial Neural Network (ANN) with three fully connected layers and tanh activation functions. The ANN trained using Adam optimizer and cross-entropy loss. We evaluated the ANN's performance on four versions (Original, Undersampled, Oversampled and Undersampled then oversampled) of both datasets using the same metrics (Precision, Recall, F1- Score, AUC-ROC Score and Cross validation) as the other models.

5. Anomaly Detection:

We also explored two anomaly detection techniques:

1. **One-class SVM:** This model learns the distribution of the majority class (legitimate transactions) and identifies instances that deviate significantly from this distribution as anomalies (potential fraud).
2. **Isolated Forest:** This model isolates anomalies by randomly partitioning the frame feature space and identifying instances that require fewer partitions to be isolated.

We evaluated the anomaly detections models using precision, recall, F1 score, and compared their performance to the classification models.

Results & Discussion (/ 30 Points)

Explain your results in detail including system/model train/validation/optimization analysis, performance evaluation and comparison with the state-of-the-art (if relevant), ablation study (if relevant), a use-case analysis or the demo of the product (if relevant), and additional points related to your project. Also include the discussion of each piece of result (i.e., what would be the reason behind obtaining this outcome, what is the meaning of this result, etc.). Include figures and tables to summarize quantitative results. Use sub-headings for each topic. This section should be between 1000-2000 words (add pages if necessary).

The results & discussions section provides a comprehensive analysis of prediction metrics for every machine learning model that we use throughout the year. Those corresponding machine-learning models were trained and tested with two different datasets. Two datasets' specific characteristics were important considerations for us to make solid observations on the subject. The main question remained the same: What is the best way to handle credit card fraud? We dived into many algorithms and hopefully, we wanted to select the best machine-learning model as much as we can observe.

The results & discussions consist of three main sections and subsections for each algorithm. In every subsection, results will be discussed through various metrics and making a call for "Is it the best option so far among every model?"

I. Credit Card Fraud Analysis with Dataset One (Real-World Transactions)

1. Supervised Learning Algorithms

For the 5 supervised learning algorithms; recall, precision, F1 score, accuracy, AUC-ROC score and Cross-Validation Mean Score metrics will be considered as a guide to which model is more accurate in predicting fraudulent results.

Precision calculation consists of True Positives and False Positives which means how many of the detected frauds were actually a fraud.

Recall is True Positive Rate which means how many records are correctly predicted as fraud.

The F1 score is a harmonic mean of the precision and recall. It shows us our model's balance between good prediction on identifying frauds and catching a good number of fraudulent activity.

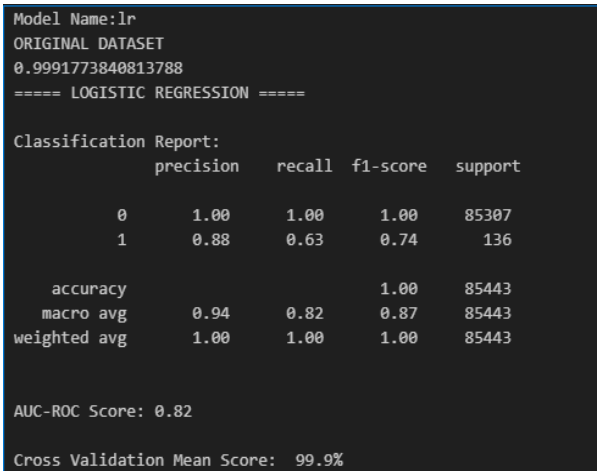
We have an AUC-ROC score which computes the area under the ROC curve. It is important to get bigger than 0.5 to show our models are not just random guessing and have good prediction scores.

Lastly, we have CV Mean Score which is when we do CV to mitigate overfitting or underfitting and try to consider unseen data, mean of that will give us an overall perspective to discuss the model's success.

We will examine four situations. We trained our models with the original dataset, undersampled dataset, oversampled (SMOTE) dataset and a combination of both; undersampled SMOTE. We will see which one is giving the best metrics.

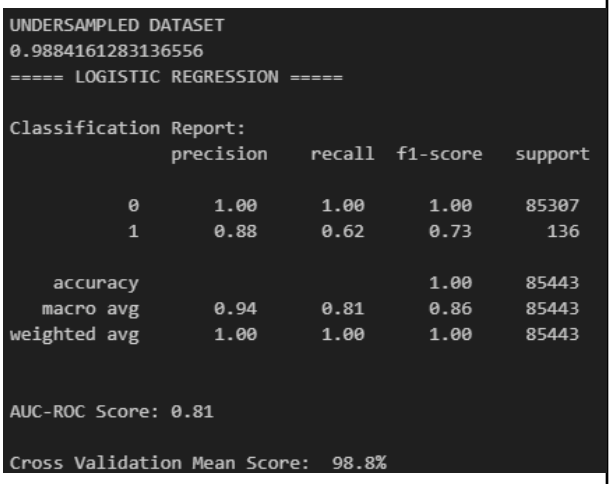
1.1. Logistic Regression

Here are the four classification reports and metric calculations. The dataset has been trained with the Logistic Regression model and the outcomes are below.



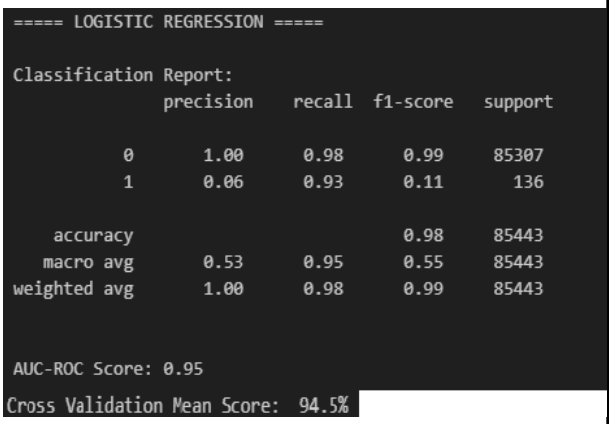
```
Model Name:lr
ORIGINAL DATASET
0.9991773840813788
===== LOGISTIC REGRESSION =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.88      0.63      0.74       136

    accuracy                           1.00     85443
   macro avg       0.94      0.82      0.87     85443
weighted avg       1.00      1.00      1.00     85443


AUC-ROC Score: 0.82

Cross Validation Mean Score:  99.9%
```

Fig-1: original



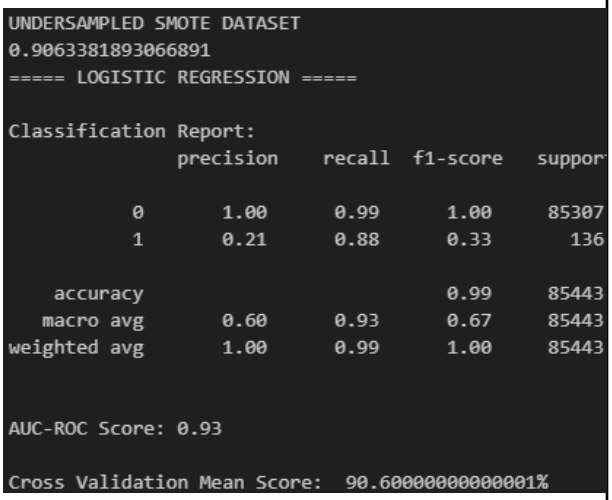
```
UNDERSAMPLED DATASET
0.9884161283136556
===== LOGISTIC REGRESSION =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.88      0.62      0.73       136

    accuracy                           1.00     85443
   macro avg       0.94      0.81      0.86     85443
weighted avg       1.00      1.00      1.00     85443


AUC-ROC Score: 0.81

Cross Validation Mean Score:  98.8%
```

Fig-2: undersampled



```
===== LOGISTIC REGRESSION =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.98      0.99     85307
           1       0.06      0.93      0.11       136

    accuracy                           0.98     85443
   macro avg       0.53      0.95      0.55     85443
weighted avg       1.00      0.98      0.99     85443


AUC-ROC Score: 0.95
Cross Validation Mean Score:  94.5%
```

Fig-3: SMOTE



```
UNDERSAMPLED SMOTE DATASET
0.9063381893066891
===== LOGISTIC REGRESSION =====

Classification Report:
              precision    recall  f1-score   suppor

           0       1.00      0.99      1.00     85307
           1       0.21      0.88      0.33       136

    accuracy                           0.99     85443
   macro avg       0.60      0.93      0.67     85443
weighted avg       1.00      0.99      1.00     85443


AUC-ROC Score: 0.93

Cross Validation Mean Score:  90.60000000000001%
```

Fig-4: undersampled SMOTE

Best precision came from the original and undersampled dataset. Among the fraud detections %88 of them were correct Logistic Regression predicts the fraud class is correct. The best recall came from SMOTE dataset. This means the model caught %93 of the fraud activities with the SMOTE dataset. The best F1 score belongs to the original dataset. It balances well between precision and recall with a percentage of %73. The best AUC-ROC score belongs to SMOTE with 0.95 and the best Cross-Validation Mean is %99.9 with the original dataset.

We can say that the original dataset under Logistic Regression, gives us the best metrics on the credit card fraud detection dataset. By applying SMOTE on the dataset we can oversample the data and can get good AUC-ROC and recall scores.

1.2. K-Nearest Neighbors

Here are the four classification reports and metric calculations. The dataset has been trained with the KNN model and outcomes are below.

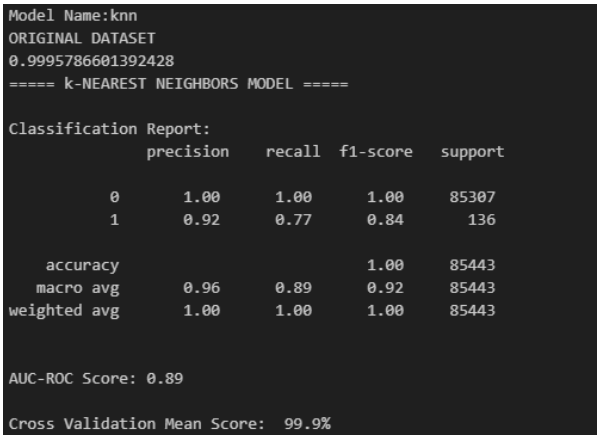


Fig-1: original

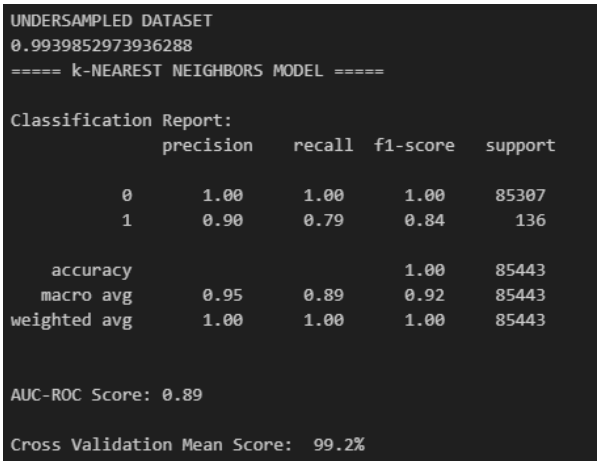


Fig-2: undersampled

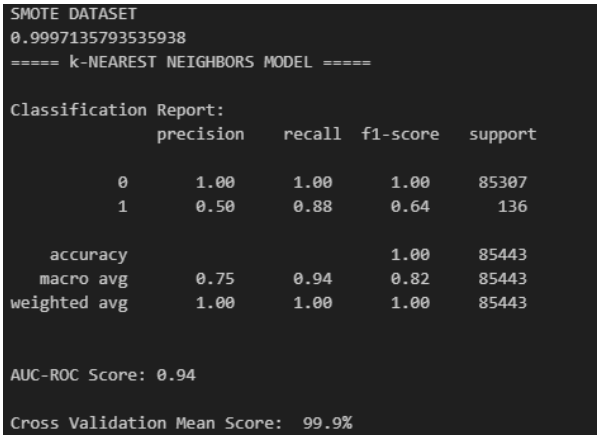


Fig-3: SMOTE

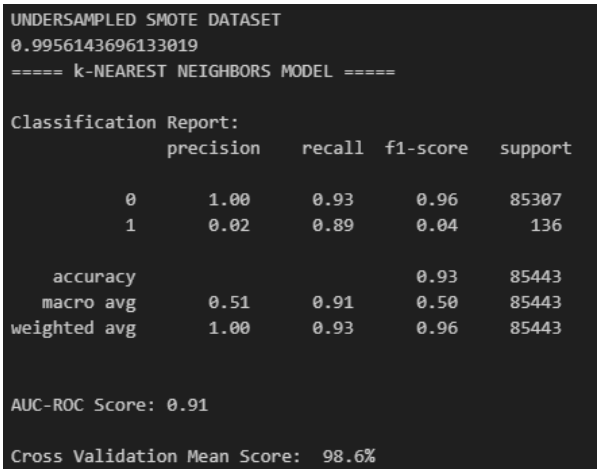


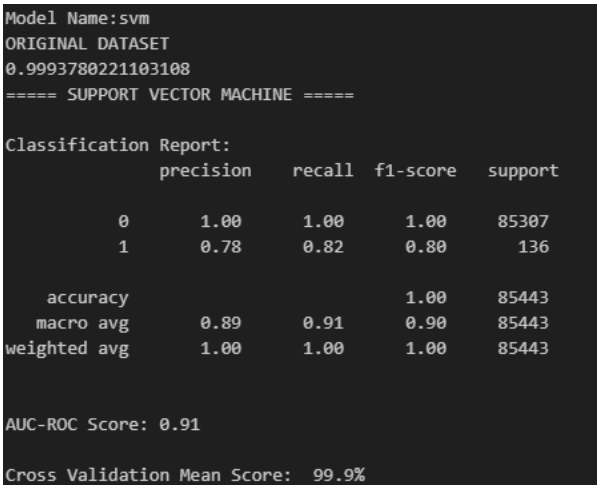
Fig-4: undersampled SMOTE

Best precision: original %92
Best recall: undersampled SMOTE %89
Best F1 Score: original and undersampled %84
Best AUC-ROC Score: SMOTE %94
Best Cross Validation Mean Score: original and SMOTE %99

With KNN, we can say that SMOTE datasets will give good results on most of the metrics. Also, KNN gives us slightly better performance than Logistic Regression and applying over-sampling was a good call for KNN.

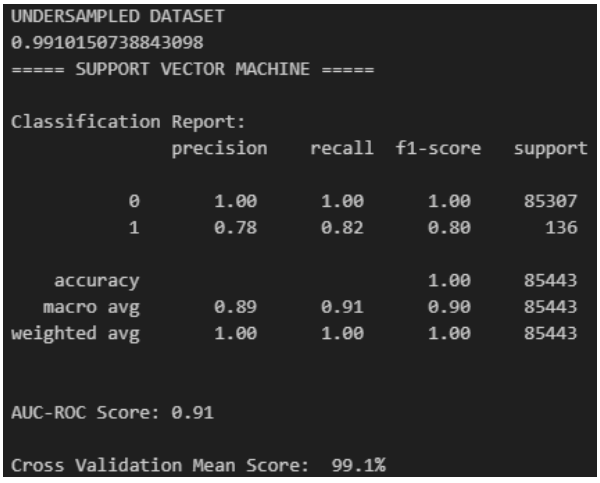
1.3. Support Vector Machines

Here are the four classification reports and metric calculations. The dataset has been trained with the SVM model and the outcomes are below.



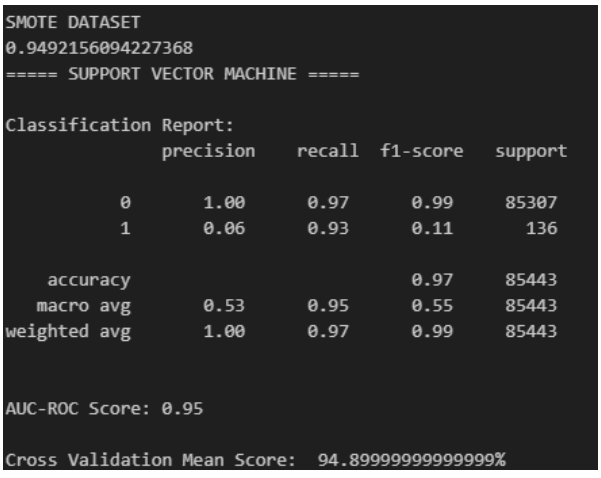
```
Model Name:svm
ORIGINAL DATASET
0.9993780221103108
===== SUPPORT VECTOR MACHINE =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.78      0.82      0.80       136

    accuracy                           1.00     85443
   macro avg       0.89      0.91      0.90     85443
weighted avg       1.00      1.00      1.00     85443


AUC-ROC Score: 0.91

Cross Validation Mean Score:  99.9%
```

Fig-1: original



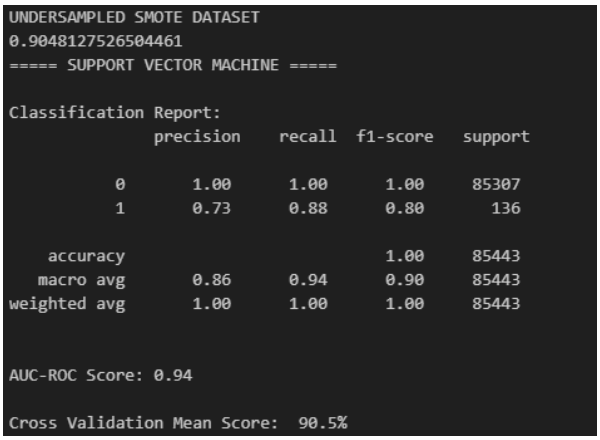
```
UNDERSAMPLED DATASET
0.9910150738843098
===== SUPPORT VECTOR MACHINE =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.78      0.82      0.80       136

    accuracy                           1.00     85443
   macro avg       0.89      0.91      0.90     85443
weighted avg       1.00      1.00      1.00     85443


AUC-ROC Score: 0.91

Cross Validation Mean Score:  99.1%
```

Fig-2: undersampled



```
SMOTE DATASET
0.9492156094227368
===== SUPPORT VECTOR MACHINE =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.99     85307
           1       0.06      0.93      0.11       136

    accuracy                           0.97     85443
   macro avg       0.53      0.95      0.55     85443
weighted avg       1.00      0.97      0.99     85443


AUC-ROC Score: 0.95

Cross Validation Mean Score:  94.89999999999999%
```

Fig-3: SMOTE



```
UNDERSAMPLED SMOTE DATASET
0.9048127526504461
===== SUPPORT VECTOR MACHINE =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.73      0.88      0.80       136

    accuracy                           1.00     85443
   macro avg       0.86      0.94      0.90     85443
weighted avg       1.00      1.00      1.00     85443


AUC-ROC Score: 0.94

Cross Validation Mean Score:  90.5%
```

Fig-4: undersampled SMOTE

Best precision: original and undersampled %78
Best recall: SMOTE %93
Best F1 Score: original undersampled and undersampled SMOTE %80
Best AUC-ROC Score: SMOTE %95
Best Cross Validation Mean Score: original %99

With SVM, we did not do good precision and F1 score. Our recall was fine and we did a good cross-validation mean score with the original data. Once again data with SMOTE gives more recall, AUC-ROC score and F1 Score.

1.4. Random Forest Classifier

Here are the four classification reports and metric calculations. The dataset has been trained with the Random Forest model and the outcomes are below.

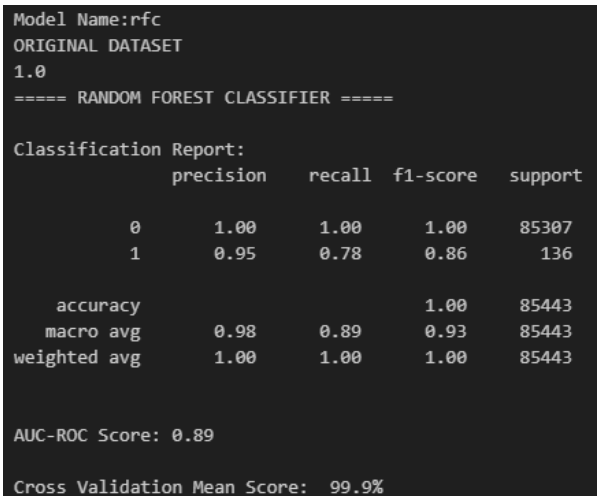


Fig-1: original

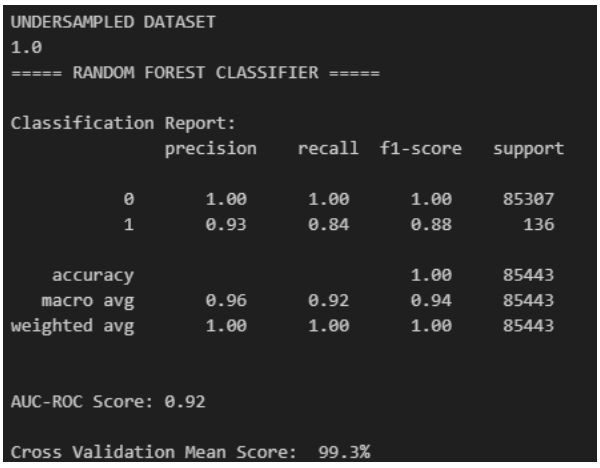


Fig-2: undersampled

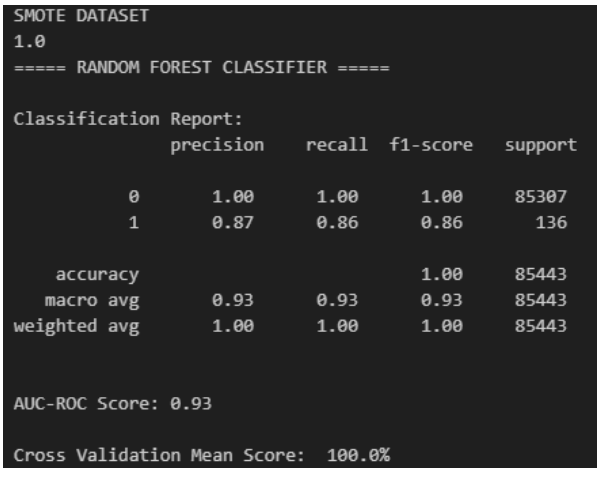


Fig-3: SMOTE

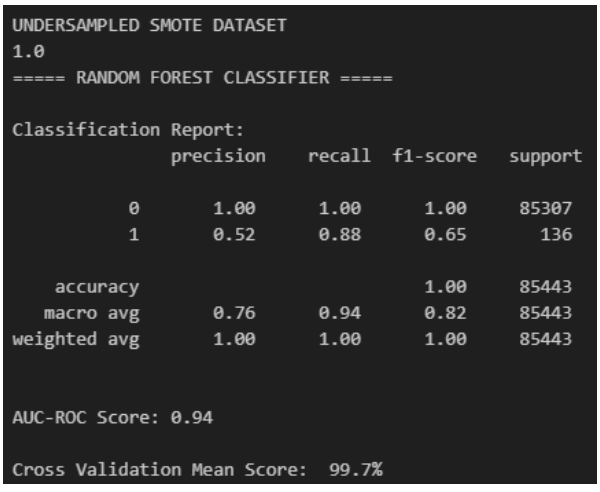


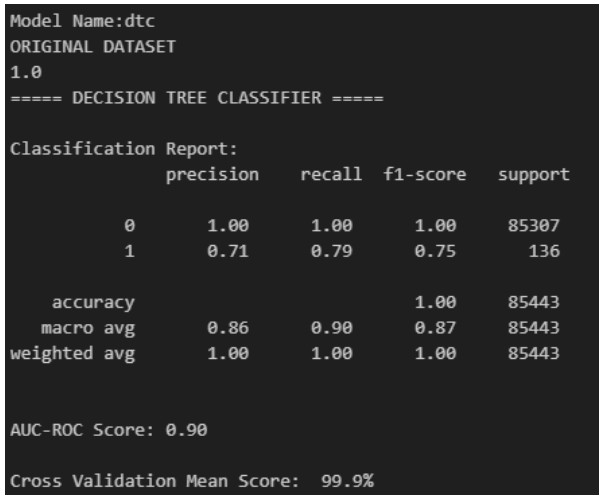
Fig-4: undersampled SMOTE

Best precision: original %95
Best recall: undersampled SMOTE %88
Best F1 Score: undersampled %88
Best AUC-ROC Score: undersampled SMOTE %94
Best Cross Validation Mean Score: SMOTE %100

Using Random Forest Classifier and undersampling/oversampling techniques we get a good recall and F1 score on predicting fraud activity. Combining undersampling and SMOTE will give us better performance for RFC and good precision on original data.

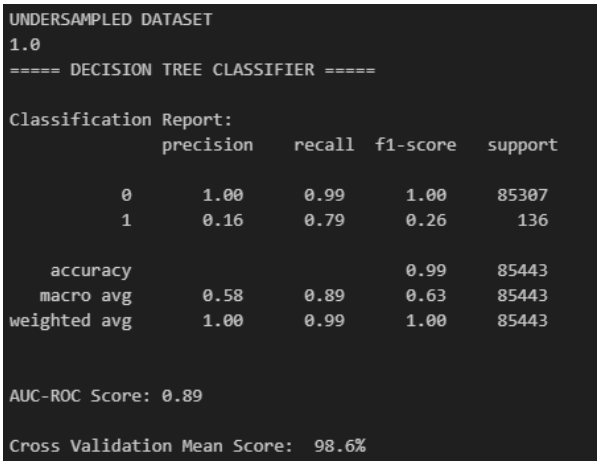
1.5. Decision Tree Classifier

Here are the four classification reports and metric calculations. The dataset has been trained with the Decision Tree model and the outcomes are below.



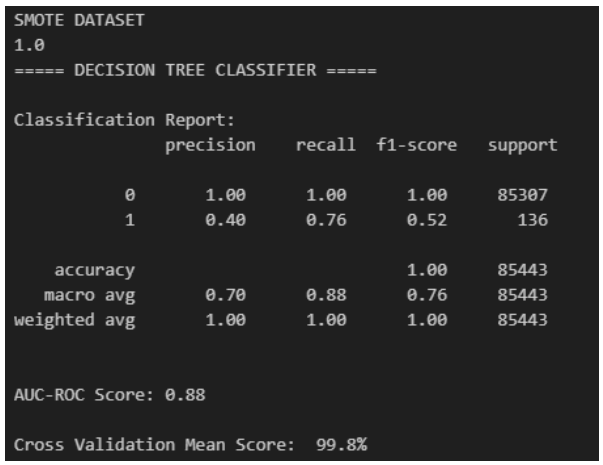
```
Model Name:dtc
ORIGINAL DATASET
1.0
===== DECISION TREE CLASSIFIER =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.71      0.79      0.75       136

    accuracy                           1.00     85443
   macro avg       0.86      0.90      0.87     85443
weighted avg       1.00      1.00      1.00     85443


AUC-ROC Score: 0.90

Cross Validation Mean Score:  99.9%
```

Fig-1: original



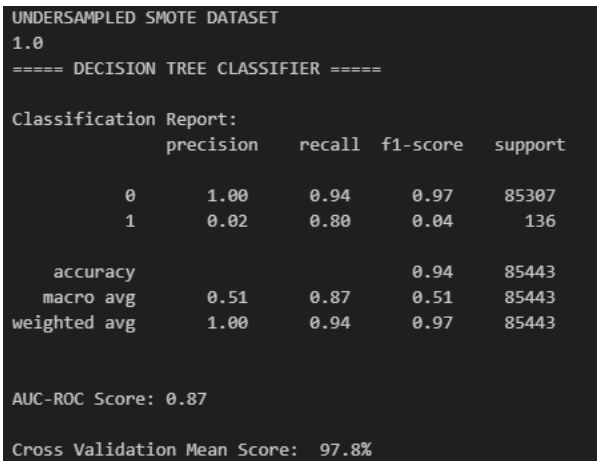
```
UNDERSAMPLED DATASET
1.0
===== DECISION TREE CLASSIFIER =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00     85307
           1       0.16      0.79      0.26       136

    accuracy                           0.99     85443
   macro avg       0.58      0.89      0.63     85443
weighted avg       1.00      0.99      1.00     85443


AUC-ROC Score: 0.89

Cross Validation Mean Score:  98.6%
```

Fig-2: undersampled



```
SMOTE DATASET
1.0
===== DECISION TREE CLASSIFIER =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.40      0.76      0.52       136

    accuracy                           1.00     85443
   macro avg       0.70      0.88      0.76     85443
weighted avg       1.00      1.00      1.00     85443


AUC-ROC Score: 0.88

Cross Validation Mean Score:  99.8%
```

Fig-3: SMOTE



```
UNDERSAMPLED SMOTE DATASET
1.0
===== DECISION TREE CLASSIFIER =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.94      0.97     85307
           1       0.02      0.80      0.04       136

    accuracy                           0.94     85443
   macro avg       0.51      0.87      0.51     85443
weighted avg       1.00      0.94      0.97     85443


AUC-ROC Score: 0.87

Cross Validation Mean Score:  97.8%
```

Fig-4: undersampled SMOTE

Best precision: original %71
Best recall: undersampled SMOTE %80
Best F1 Score: original %75
Best AUC-ROC Score: original %90
Best Cross Validation Mean Score: original %99.9

Lastly we trained our data with Decision Tree Classifier in Supervised Learning. We get bad results compared to other model implementations. Undersampling/oversampling techniques that we applied did not give us effective numbers.

1.6. Conclusion For Supervised Learning Algorithms

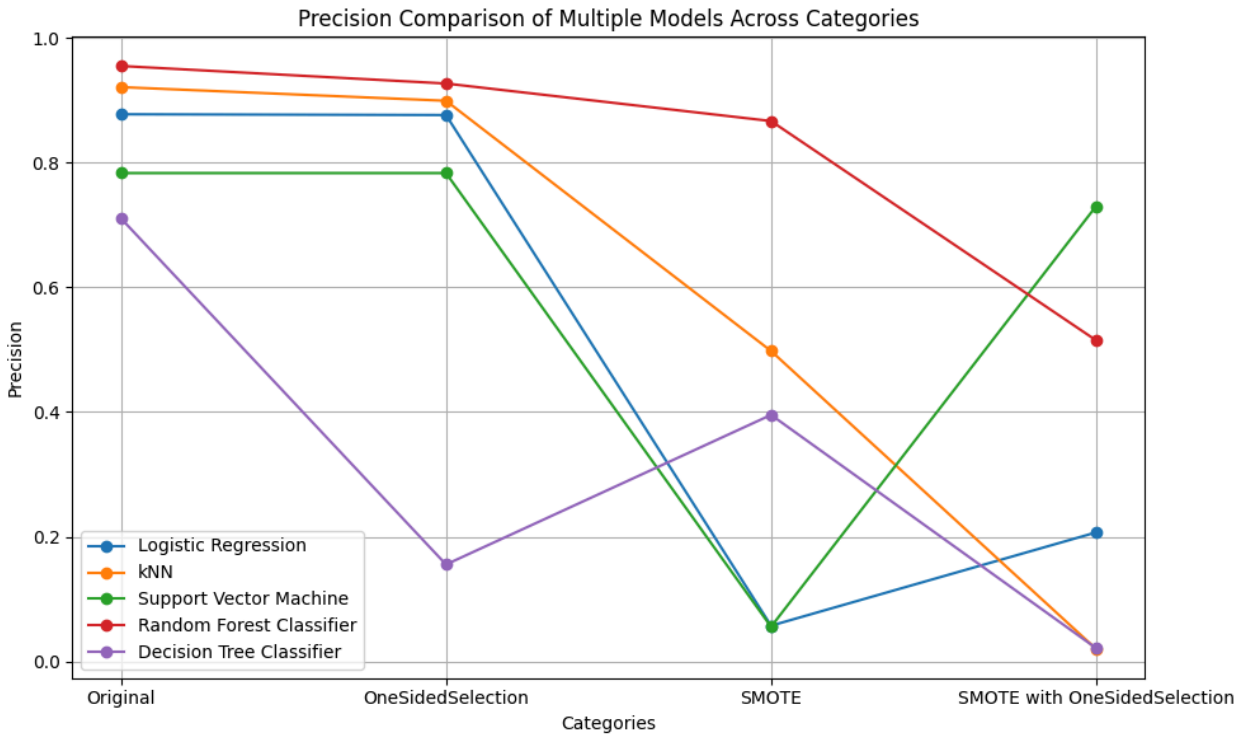


Figure-1

We trained our original, undersampled, oversampled and undersampled smote data with 5 different machine learning models. We can conclude that we got the best result for each metric with different algorithms but the Random Forest classifier did a very good job overall among the others as can be seen in Figure-1. Its robustness to overfitting and handling large dataset specifications impacts the overall conclusion. %95 precision score was the best result we got among all algorithms. Secondly, KNN proved that it was a good choice with Logistic Regression. Decision Tree Classifier and SVM did not give us the best metrics overall but they also proved that there will always be trade-offs between metrics when choosing to handle a

delicate matter like credit card fraud detection. Of course, there were trade-offs between undersampling/oversampling the data or using the original data. Considering the outside parameters like budget, time, memory, processor use, and data size; processing the data can make things easier for us to detect fraudulent activities. We can always select a lighter model with a processed dataset in real-life situations.

In precision-manner Random Forest Classifier with the original dataset can be good but if we want to consider the AUC Score meaning, the measure of the ability to distinguish between classes, Logistic Regression with SMOTE dataset made a huge impact as shown in Figure-2

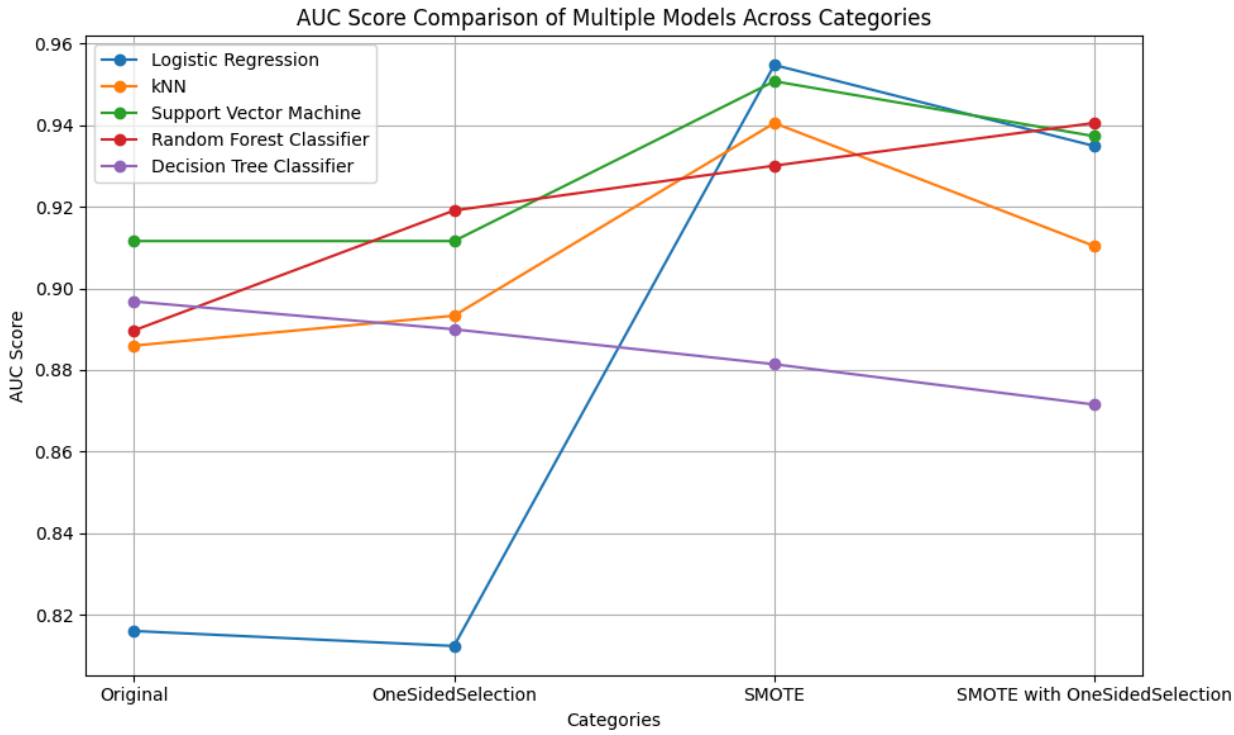


Figure-2

We get good precisions through Random Forest with the original dataset but we also want a good hybrid score of both precision and recall. That would be more likely to understand how well and completely we predict and detect the fraud. Since F-Score is a harmonic mean of precision and recall it is important to look at that too.

When we undersampled the dataset with OneSidedSelection, the Random Forest Classifier with the undersampled dataset gave us the best F-Score. Now this is a clearer picture to understand how well models behave both in precision and in recall manner with those processed datasets.

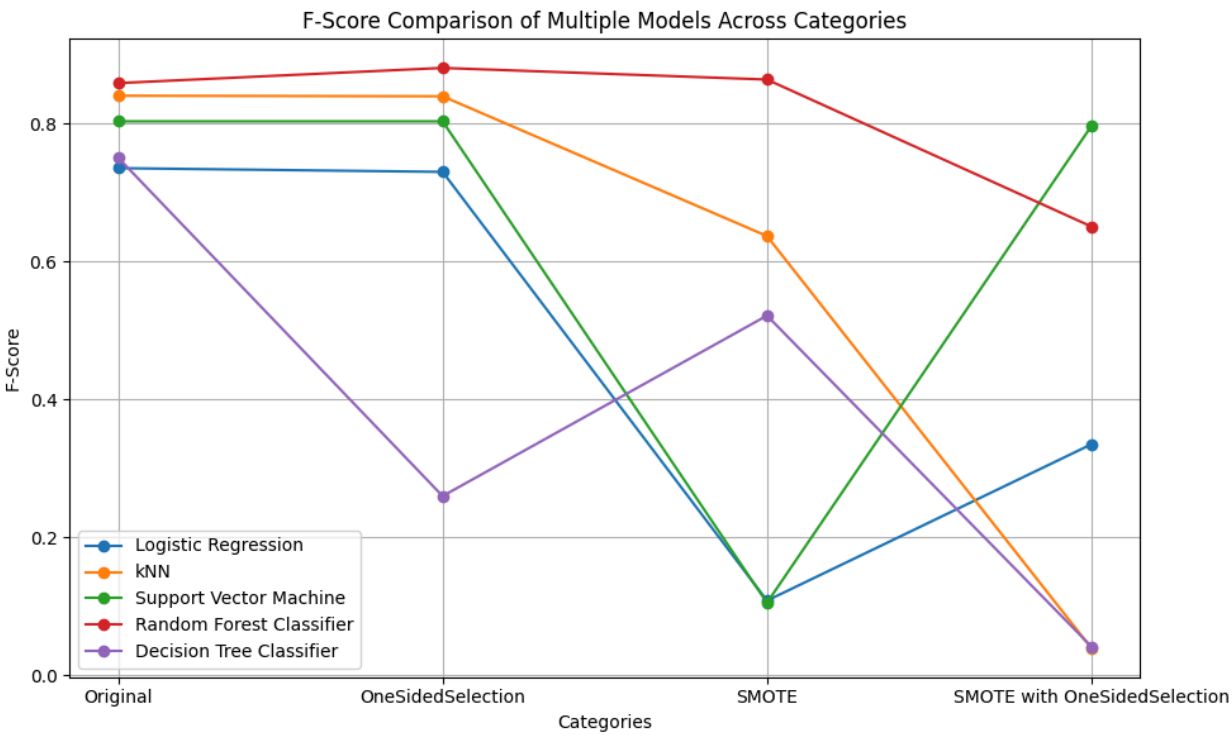


Figure-3

Lastly, we have Cross Validation Accuracy Comparison. As we can see from the graph we managed to get the best CV Accuracy on Random Forest with SMOTE dataset. Other models are as good as this one but oversampling the dataset improved our quality of cross-validations.

Notice that SMOTE with OneSidedSelection wasn't a good choice for cross-validation. Especially SVM and Logistic Regression decrease dramatically with undersampled, SMOTE and undersampled SMOTE data.

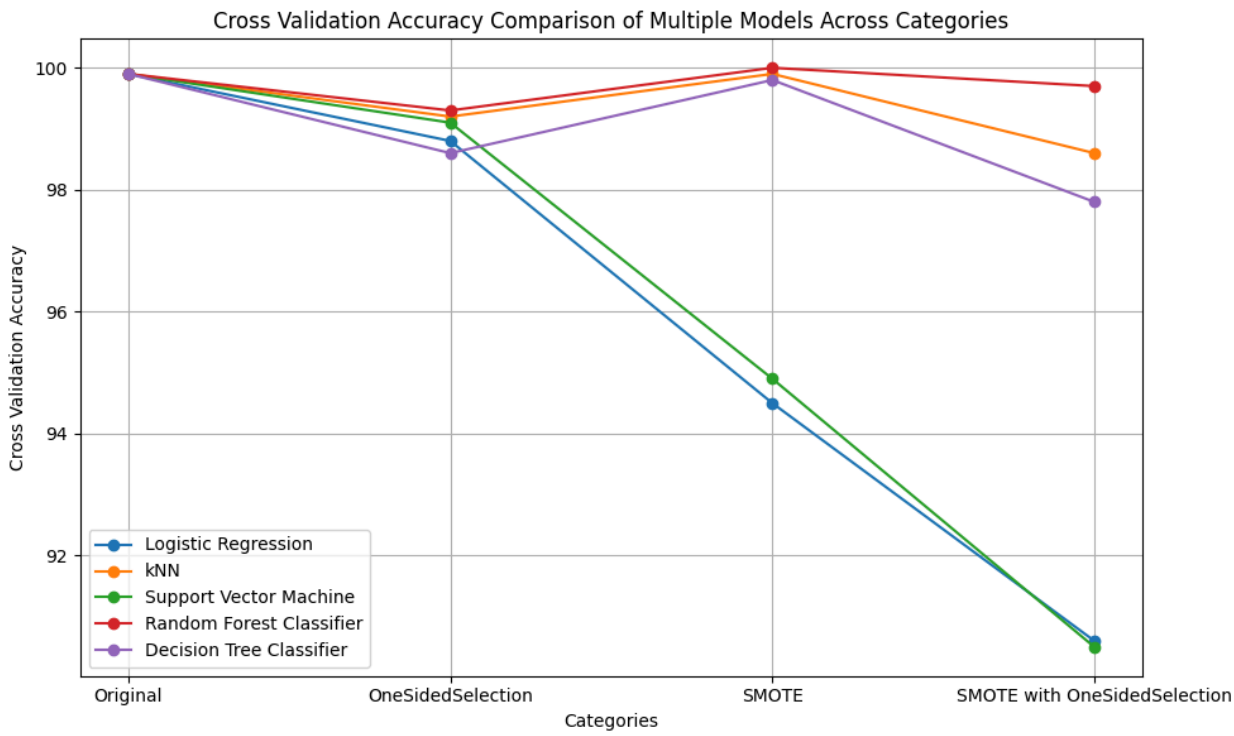

Cross Validation Accuracy Comparison of Multiple Models Across Categories
Cross Validation Accuracy
100
98
96
94
92
Logistic Regression
kNN
Support Vector Machine
Random Forest Classifier
Decision Tree Classifier
Original
OneSidedSelection
SMOTE
SMOTE with OneSidedSelection
Categories

Figure-4

2. Anomaly Detection Algorithms

We had to try anomaly detection algorithms as well. Especially to see how good idea to use anomaly detection for the detection of credit card fraud. We did not undersample or oversample the dataset because making synthetic anomalies could affect the models' reliability. Also since the anomalies are so much less than normal data in the dataset, this could serve our purpose to find anomalies.

We consider mostly the F-Score as a measurement metric.

2.1. One-Class Support Vector Machine

Here are the metric calculations for the first algorithm One-Class Support Vector Machine. First, we trained the model with a default threshold then set a customized threshold and small differences emerged.

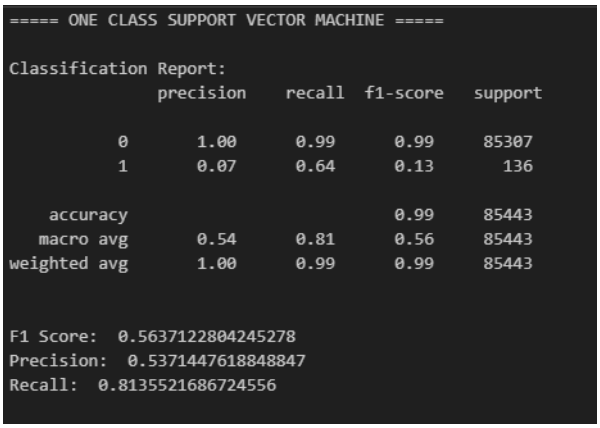


Fig-1: one-class SVM with default threshold

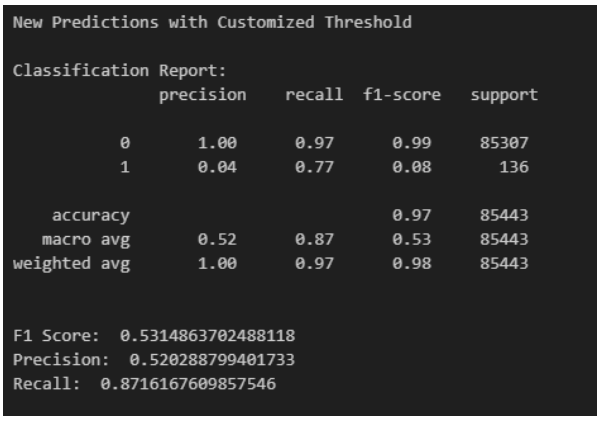


Fig-2: one-class SVM with customized threshold

We got %56 F1 Score for the default threshold and %53 F1 Score for the customized threshold. These poor results proved that applying one-class SVM to raw credit card data was not a good option. We concluded that detecting fraud activity as anomalies using one-class SVM was a poor choice.

2.2 Isolation Forest

We decided to run an Isolation Forest anomaly detection technique after One-Class Support Vector Machine's weak results.

```
===== ISOLATION FOREST =====

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.22      0.51      0.31       136

    accuracy                           1.00     85443
   macro avg       0.61      0.75      0.65     85443
weighted avg       1.00      1.00      1.00     85443


F1 Score:  0.6548614804054165
Precision:  0.6119843620967135
Recall:  0.7522815088617649
```

Fig-1: Isolation Forest

We got %75 Recall, %61 Precision and %65 F1 Score which was better than One-Class SVM but still not as well as the Random Forest Classifier from previous section.

Anomaly Detection was a try. We were dealing with a critical subject like credit card fraud and seeing fraud activity as an anomaly was an idea. We discussed what could also be done to get better results with anomaly detection algorithms like better tuning of the models.

3. Neural Networks

Last step was making predictions with Neural Networks. Since it's more complex we expected some trade-offs for good metric scores.

Here are the final results for Neural Network

```
Accuracy on test set: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.77      0.80      0.79       136

    accuracy                           1.00     85443
   macro avg       0.89      0.90      0.89     85443
weighted avg       1.00      1.00      1.00     85443

AUC Score: 0.95
```

Fig-1: original

```
Accuracy on test set: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.77      0.83      0.80       136

    accuracy                           1.00     85443
   macro avg       0.88      0.92      0.90     85443
weighted avg       1.00      1.00      1.00     85443

AUC Score: 0.96
```

Fig-2: undersampled

```
Accuracy on test set: 0.97
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.99     85307
           1       0.05      0.93      0.10       136

    accuracy                           0.97     85443
   macro avg       0.53      0.95      0.54     85443
weighted avg       1.00      0.97      0.99     85443

AUC Score: 0.99
```

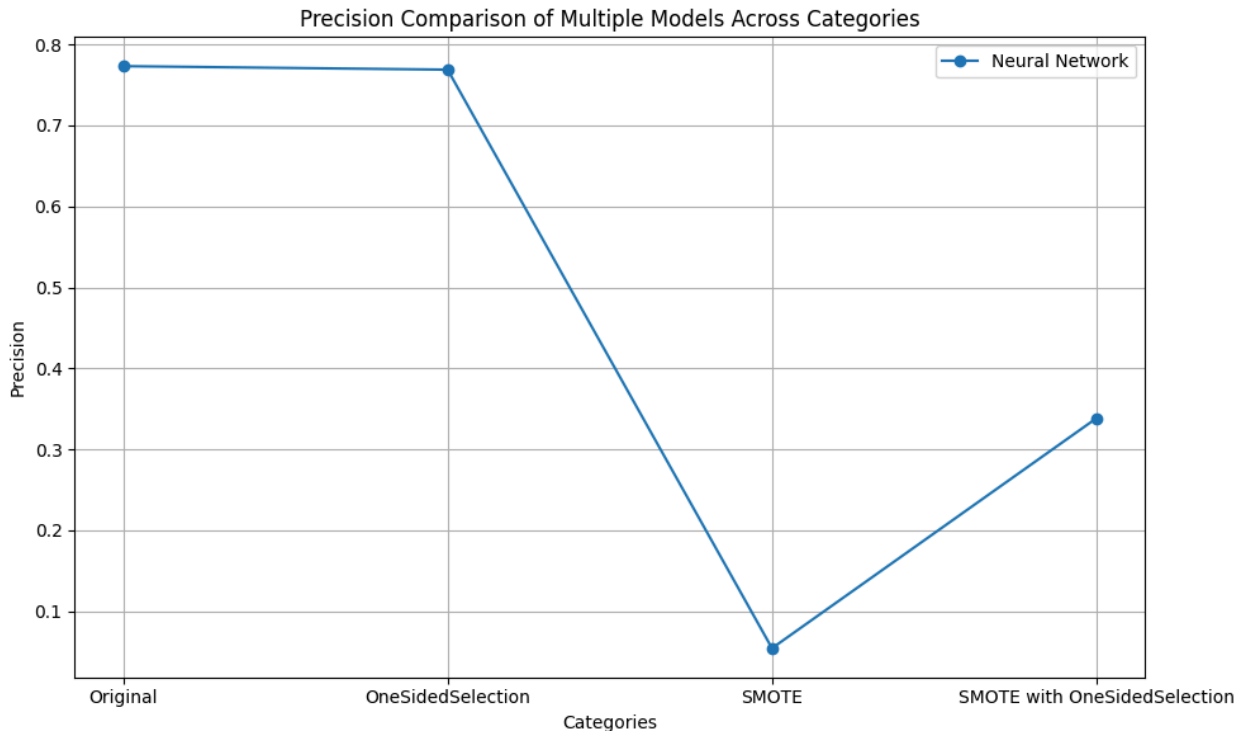
Fig-3: SMOTE

```
Accuracy on test set: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.34      0.82      0.48       136

    accuracy                           1.00     85443
   macro avg       0.67      0.91      0.74     85443
weighted avg       1.00      1.00      1.00     85443

AUC Score: 0.97
```

Fig-4: undersampled SMOTE

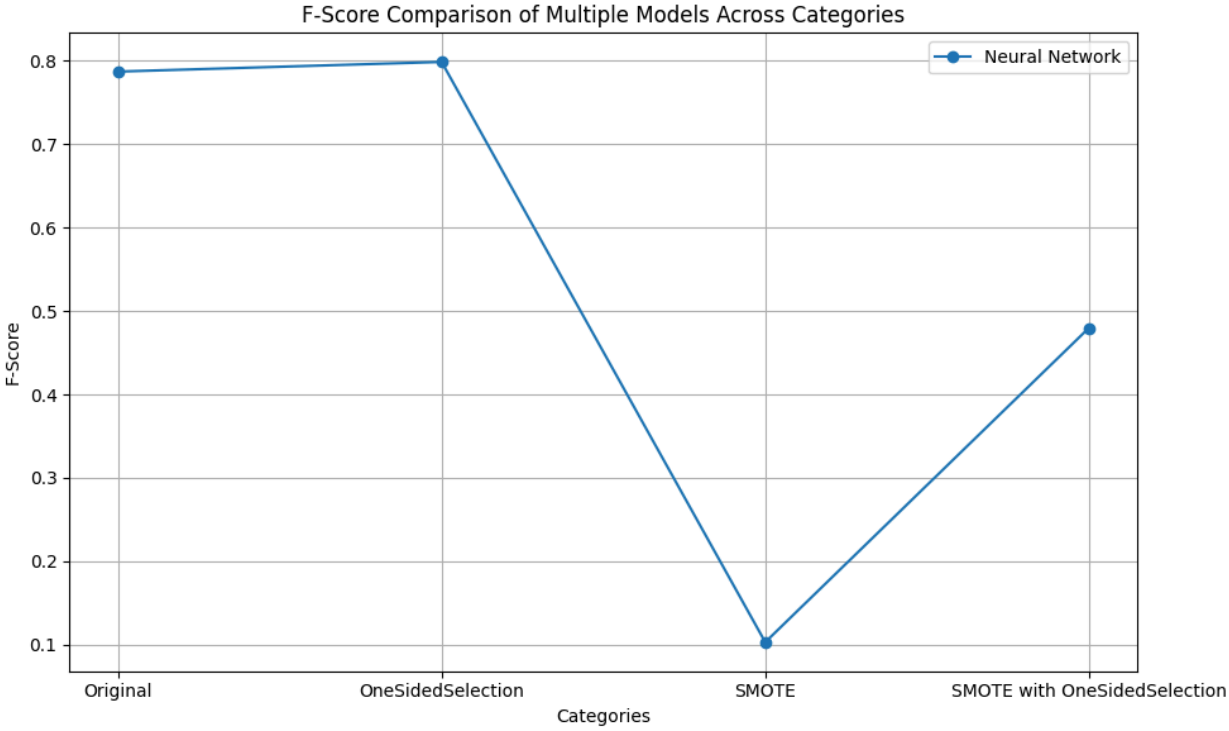
Best precision: original and undersampled %77
Best recall: SMOTE %93
Best F1 Score: undersampled %80
Best AUC Score: SMOTE %99

With the neural networks we get good precision and recall, good F1 score and good AUC score so far. Neural Networks were a good technique for detecting fraud activities on credit card fraud datasets. We conclude that by using neural networks we can achieve our goal. Now let's look at some comparison graphs between different approaches.



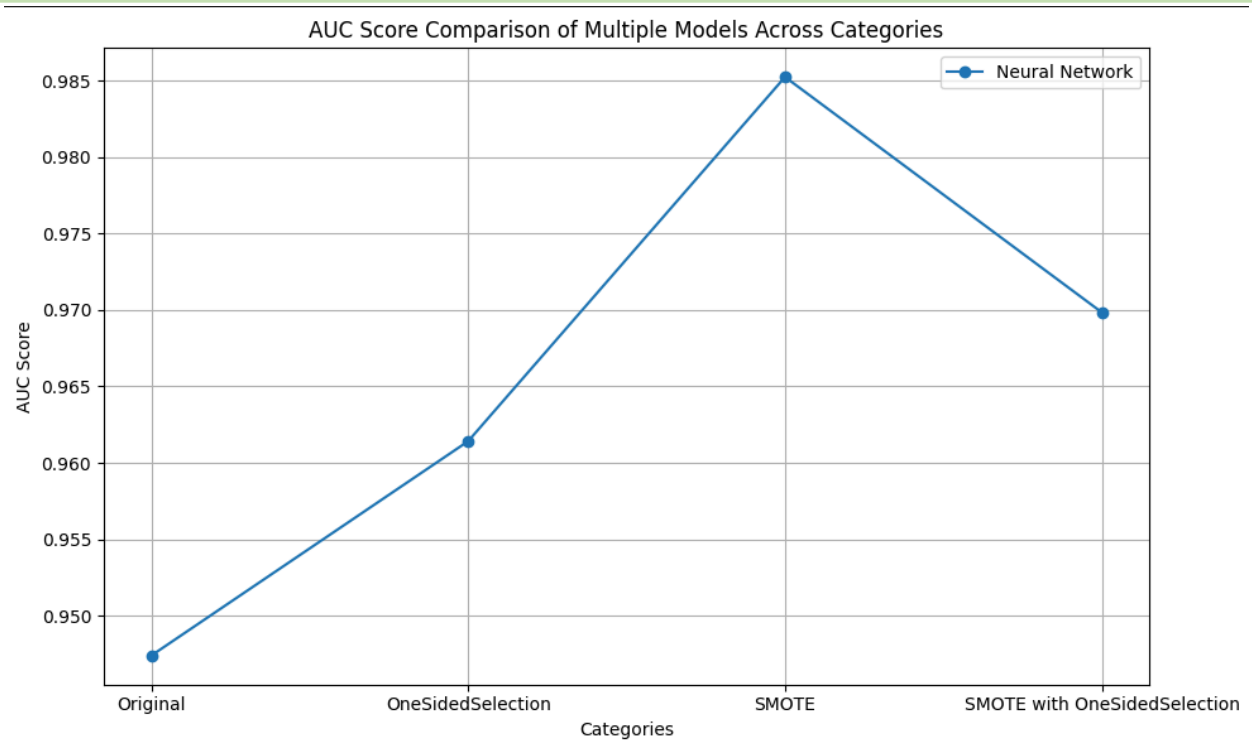
Graph-1

In precision comparison we see the best precision was on the original dataset's neural network model. SMOTE had a bad effect on neural networks. With this low precision, we conclude that we can't use SMOTE datasets with neural networks.



Graph-2

F-Score was best with the undersampled dataset. Even though the recall score was good with the SMOTE dataset low precision resulted in a low F1 score. Undersampling the dataset can slightly increase our performance in predicting fraud activities.



Graph-3

Surprising outcomes were in AUC Scores. SMOTE dataset gave the best results with the AUC Scores. And worse on the original dataset. Considering the trade-offs between dataset processing on neural networks we concluded that it is best to use undersampled data with neural networks to get consistent and successful metric scores.

II. Credit Card Fraud Analysis with Dataset Two (Simulated Transactions)

We wanted to analyze with an extra and different dataset. This dataset that we processed and trained contains simulated transactions. However, with this dataset, processing was nearly impossible and some hardware and runtime related issues emerged. We get weak scores with this dataset. Processing was quite difficult. In the Impact and Future Directions section, we discussed how can we manage this low quality results and detect fraud activity working with synthetic data. Here are some results from using a second dataset with simulated transactions.

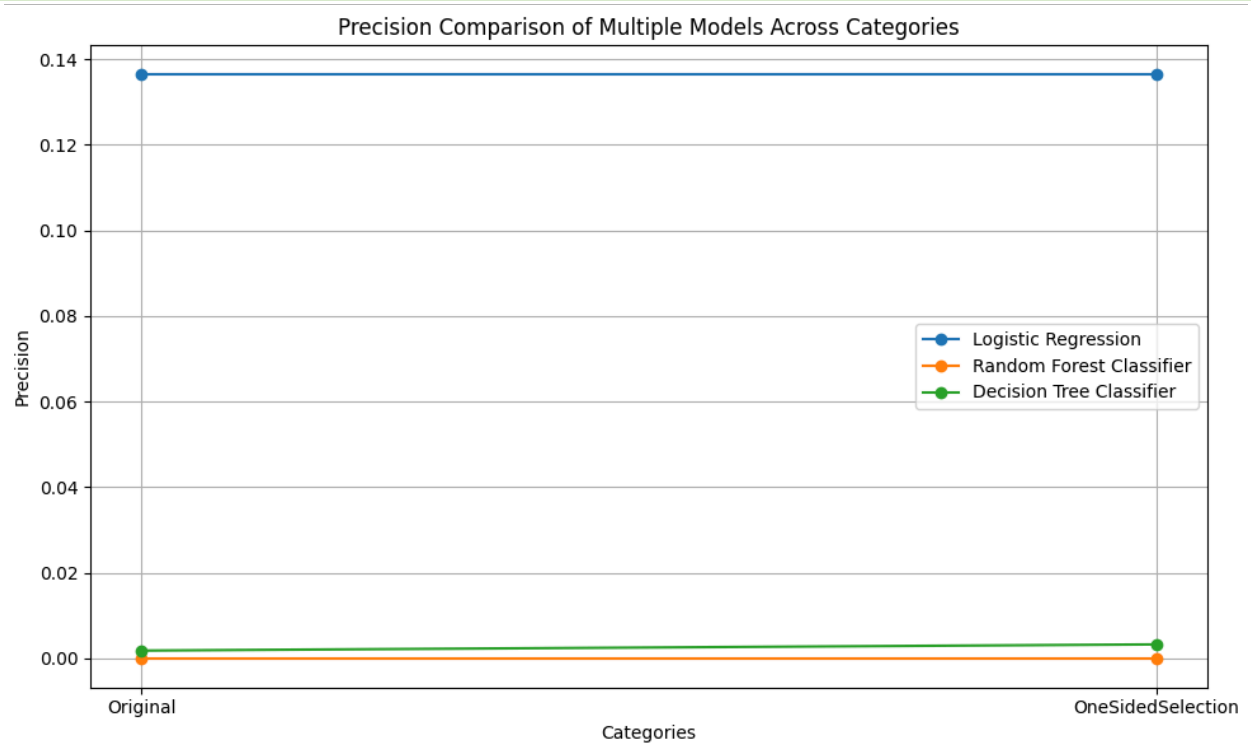


Figure-1

In Figure-1 we get very low scores and processing the data did not have a huge impact. The best precision we get is with the Logistic Regression algorithm however due to the problems we have mentioned above, it was still not enough for the experiment.

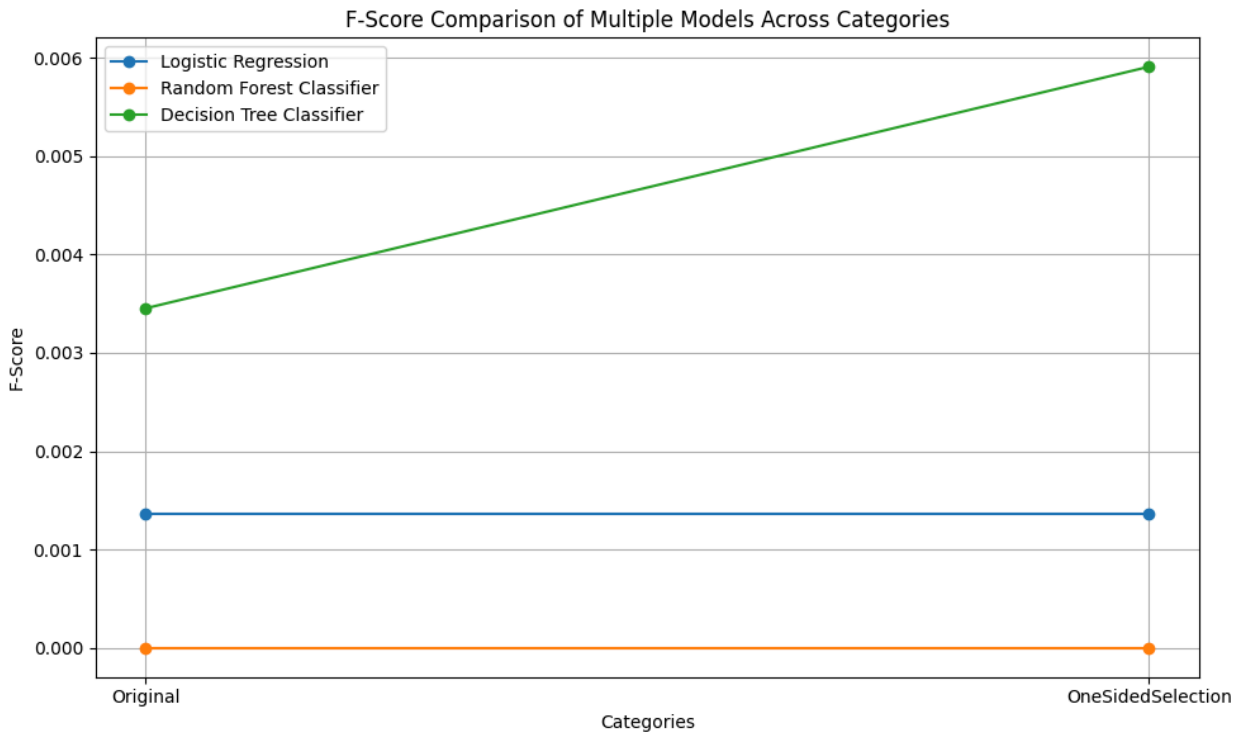


Figure-2

F-Score comparison of our models still has the same weak results. This time Decision Tree Classifiers were slightly better on F-Score. All our models even with the most reliable ones for this kind of fraud detection dataset, did very bad scores especially considering the first dataset with the real-life data.

III. Conclusion

To conclude what we have done so far, detecting credit-card fraud activity can have many trade-offs. We investigated some directions for credit card fraud detection and tried to minimize these trade-offs. We wanted to encourage to choose the right algorithms for this problem. Which methods and which approaches are more relevant for detecting fraud and inspiring future works was our motivation. Different approaches could impact drastically and ones that are working on this problem in the future context should consider those impacts.

The Impact and Future Directions (/ 15 Points)

Explain the potential (or current if exist) impacts of your outcome in terms of how the methods and results will be used in real life, how it will change an existing process, or where it will be published, etc. Also, explain what would be the next step if the project is continued in the future, what kind of qualitative and/or quantitative updates can be made, shorty, where this project can go from here? This section should be between 250-500 words.

Studying credit card fraud activities is important for giving the right feedback for methods and approaches that will be used. In real life, people suffer from this fraudulent activity every day. When we considered how widely online shopping and credit card activities influence our everyday world and how important these transactions are, it is critical to grasp the problem of attackers and fraud activities throughout the world. Credit card fraud detection can't be detected with analogous solutions every day. The world is going in a direction where we cannot do any money or online transactions without entering the range of an attacker who wants to steal our money. Transactions happen everywhere and every time. We need automation and speed. So, determining the right approaches to work on this problem, which is what we did, can have a huge impact. Our credit card fraud detection project has the potential to make a real difference in how financial institutions combat fraud. By comparing and improving upon existing methods, the research could lead to significantly higher fraud detection rates while minimizing false positives. This would translate into reduced financial losses for both businesses and consumers, creating a safer and more secure financial environment

Considering that this is also the beginning of the process and we need to process the data for different circumstances. We need to do more detailed feature engineering. This project can go from researching and applying these feature engineering tactics to making models train better and also applying more fine-tuning for the algorithms that we used. In the future, we envision our models being integrated into real-time payment processing systems, proactively preventing fraud before it even occurs. Also working with increased quality hardware should be emphasised to decrease the detection speed considering how fast this fraudulent activity can occur. This project with the mentioned improvements above and with more qualified research, can guide to a non-stop working detection tool to intimidate and eventually stop fraudulent activity for more reliable money transaction environments.

References

Itoo F, Singh S. 2021. Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. International Journal of Information Technology 13(4):1503–1511 DOI 10.1007/s41870-020-00430-y.

Alenzi HZ, Aljehane NO. 2020. Fraud detection in credit cards using logistic regression. International Journal of Advanced Computer Science and Applications 11(12):5570 DOI 10.14569/issn.2156-5570.

Manlangit S, Azam S, Shanmugam B. 2019. Novel machine learning approach for analyzing anonymous credit card fraud patterns. International Journal of Electronic Commerce Studies 10(2):175–202 DOI 10.7903/ijecs.1732.

Li C, Ding N, Zhai Y, Dong H. 2021. Comparative study on credit card fraud detection based on different support vector machines. Intelligent Data Analysis 25(1):105–119 DOI 10.3233/IDA-195011.

Pavithra T, Thangadurai K. 2019. The improving perdition of credit card fraud detection on PSO optimized SVM. The International Journal of Analytical and Experimental Modal Analysis XI(IX):478–485.

Marazqah Btoush EAL, Zhou X, Gururajan R, Chan KC, Genrich R, Sankaran P. 2023. A systematic review of literature on credit card cyber fraud detection using machine and deep learning. PeerJ Computer Science 9:e1278 https://doi.org/10.7717/peerj-cs.1278

Amusan E, Alade O, Fenwa OD, Emuoyibofarhe JO. 2021. Credit card fraud detection on skewed data using machine learning techniques. Lautech Journal of Computing and Informatics 2(1):49–56

Ata O, Hazim L. 2020. Comparative analysis of different distributions dataset by using data mining techniques on credit card fraud detection. Tehnički Vjesnik 27(2):618–626 DOI 10.17559/TV-20180427091048

Choubey R, Gautam P. 2020. Combined technique of supervised classifier for the credit card fraud detection. Shodah Sarita 7:27–32.

Hammed M, Soyemi J. 2020. An implementation of decision tree algorithm augmented with regression analysis for fraud detection in credit card. International Journal of Computer Science and Information Security (IJCSIS) 18(2):79–88

Meenu, Gupta S, Patel S, Kumar S, Chauhan G. 2020. Anomaly detection in credit card transactions using machine learning. International Journal of Innovative Research in Computer Science & Technology (IJIRCST) 8(3):2020 DOI 10.21276/ijircst.2020.8.3.5.

Agarwal A, Iqbal M, Mitra B, Kumar V, Lal N. 2021. Hybrid CNN-BILSTM-attention based identification and prevention system for banking transactions. NVEO-Natural Volatiles and Essential Oils Journal 8(5):2552–2560.

Asha RB, Suresh Kumar KR. 2021. Credit card fraud detection using artificial neural network. Global Transitions Proceedings 2(1):35–41 DOI 10.1016/j.gltp.2021.01.006.