DATA MINING FINAL PROJECT ON
CREDIT CARD FRAUD DETECTION DATA

BY

Mr. Rutik Kothwala

rk29015n@pace.edu

UID: -U01868702

In

Seidenberg College of Computer Science and Informa on Science

Pace University

TABLE OF CONTENT

Introduction:

Credit card fraud is a significant concern for financial institutions and cardholders worldwide. With the increasing use of credit cards for online transactions, detecting and preventing fraudulent activities has become crucial. In this project, we aim to develop a data mining solution to detect credit card fraud using a dataset obtained from Kaggle.

Problem Statement:

The objective of this project is to build a robust and accurate credit card fraud detection system using data mining techniques. Given a dataset containing historical credit card transactions, we need to develop a model that can effectively distinguish between legitimate transactions and fraudulent ones.

Challenges:

Imbalanced Data: Credit card fraud is a relatively rare event compared to legitimate transactions, leading to a severe class imbalance in the dataset. This imbalance can pose a challenge intraining a robust model that accurately detects fraud instances while minimizing false positives.

Feature Engineering: The dataset includes anonymized features,which may require careful analysis and feature engineering to extract meaningful patterns and relationships that can aid in frauddetection.

Objectives:

Exploratory Data Analysis: Perform a thorough analysis of the provided dataset to gain insights into the distribution of features,identify potential outliers, and understand the characteristics of fraudulent transactions.

Data Preprocessing: Cleanse and preprocess the dataset, including handling missing values, dealing with outliers, andnormalizing/standardizing features. Consider appropriate strategies to handle the class imbalance issue.

Feature Engineering: Extract relevant features from the datasetand create new features that can potentially enhance the performance of the fraud detection model.

Model Development: Implement and evaluate various data miningalgorithms suitable for credit card fraud detection, such as logisticregression, decision trees, random forests, and support vector machines. Employ appropriate techniques for model training, validation, and evaluation.

Performance Evaluation: Assess the performance of the developed models using appropriate evaluation metrics, such as accuracy, precision, recall, and F1-score. Compare and analyze the results to identify the most effective model for credit card frauddetection.

Dataset Description:

The Credit Card Fraud Detection dataset, available on Kaggle https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, is a comprehensive collection of credit card transactions recorded over a two-day period. The dataset is anonymized due to privacyconcerns and consists of a total of 284,807 transactions, out of which only 492 (0.17%) are labeled as fraudulent.

The dataset includes a set of numerical features extracted from the transaction data, as well as the transaction amount and the target variable indicating whether the transaction is fraudulent ornot. The features have undergone a dimensionality reduction technique called Principal Component Analysis (PCA) for confidentiality reasons.

The specific details of the dataset are as follows:

Time: The number of seconds elapsed between this transactionand the first transaction in the dataset.

V1-V28: These are the anonymized features resulting from the PCA transformation. The original features have been transformedto maintain the privacy of the individuals involved.

Amount: The transaction amount, which can be used as a featurefor analysis.
Class: The target variable, where 1 indicates a fraudulenttransaction and 0 indicates a non-fraudulent transaction.

DATA TYPES OF ALL THE FEATURES OF OUR DATA:

Time: Continuous variable representing the number of seconds elapsed between this transaction and the first transaction in thedataset.

Data type: Numeric (float or integer)
V1-V28: Anonymized features resulting from a PCA (PrincipalComponent Analysis) transformation to protect sensitive information.

Data type: Numeric (float)
Amount: Transaction
amount.

Data type: Numeric (float)
Class: Binary variable indicating whether the transaction isfraudulent (1) or not (0).

Data type: Categorical (integer or boolean)

So here we can see that all the features are of Numeric type andour target features class is of Categorical data type.

Outliers:

As the dataset is transformed with PCA we can assume that theoutliers are already treated.

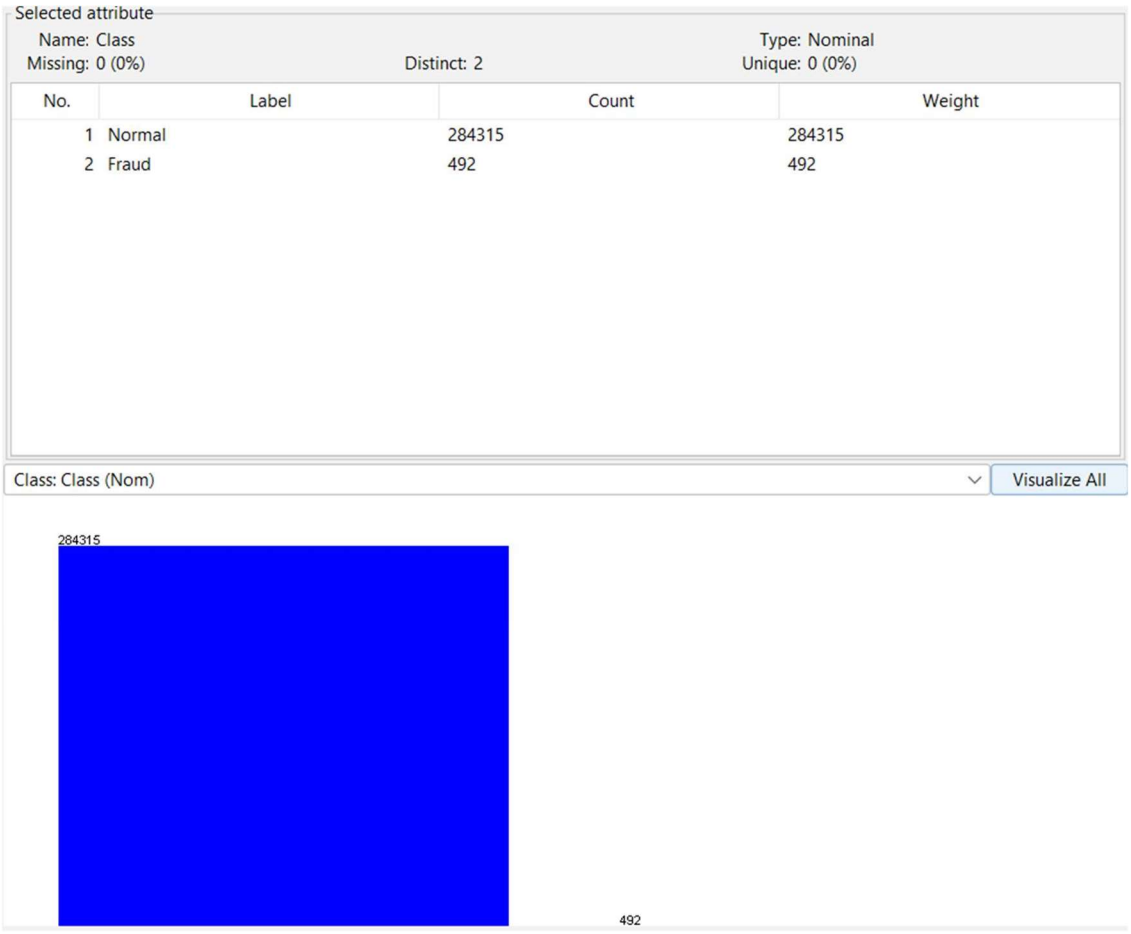
Observation of distribution of our target class:

```
Non-Fraudulent : 99.83 %
    Fraudulent : 0.17 %
==============================================
```

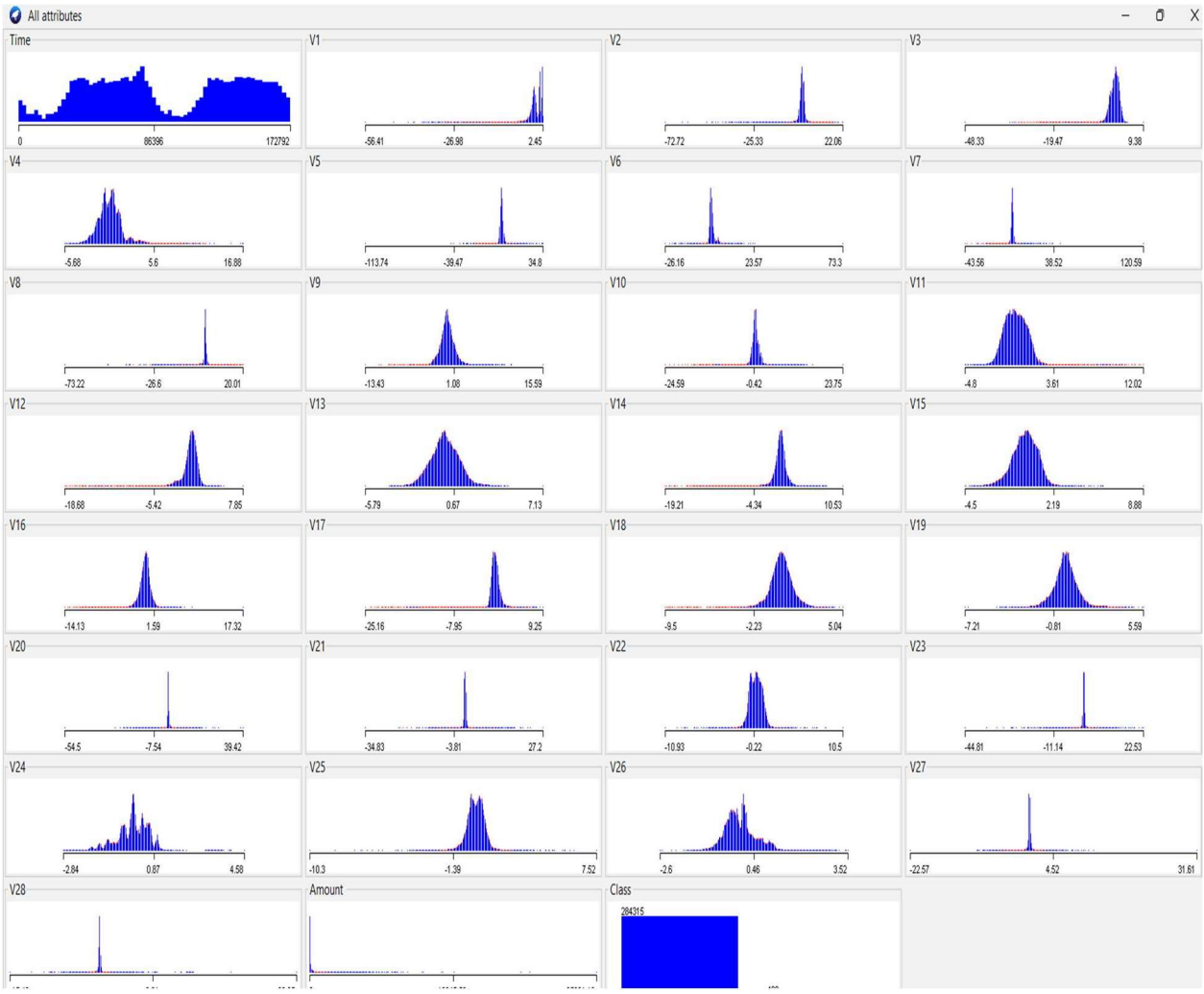
```
 Normal_share= 99.82725143693798
 Fraud_share= 0.1727485630620034
==============================================
 Imbalance Percentage = 0.173047500131896
```

It is easily observed that dataset is highly imbalanced as non-Fraudulent data is 99.83 % where as Fraudulent data is only 0.17%.

Here at First in Weka Data type For Class was showing numeric then I converted to Nominal with labeled Normal andFraud.



Distribution diagram for all the features of our dataWRT To CLASS:



We can see that the majority of the feature distributions for bothfraudulent and legitimate transactions overlap.

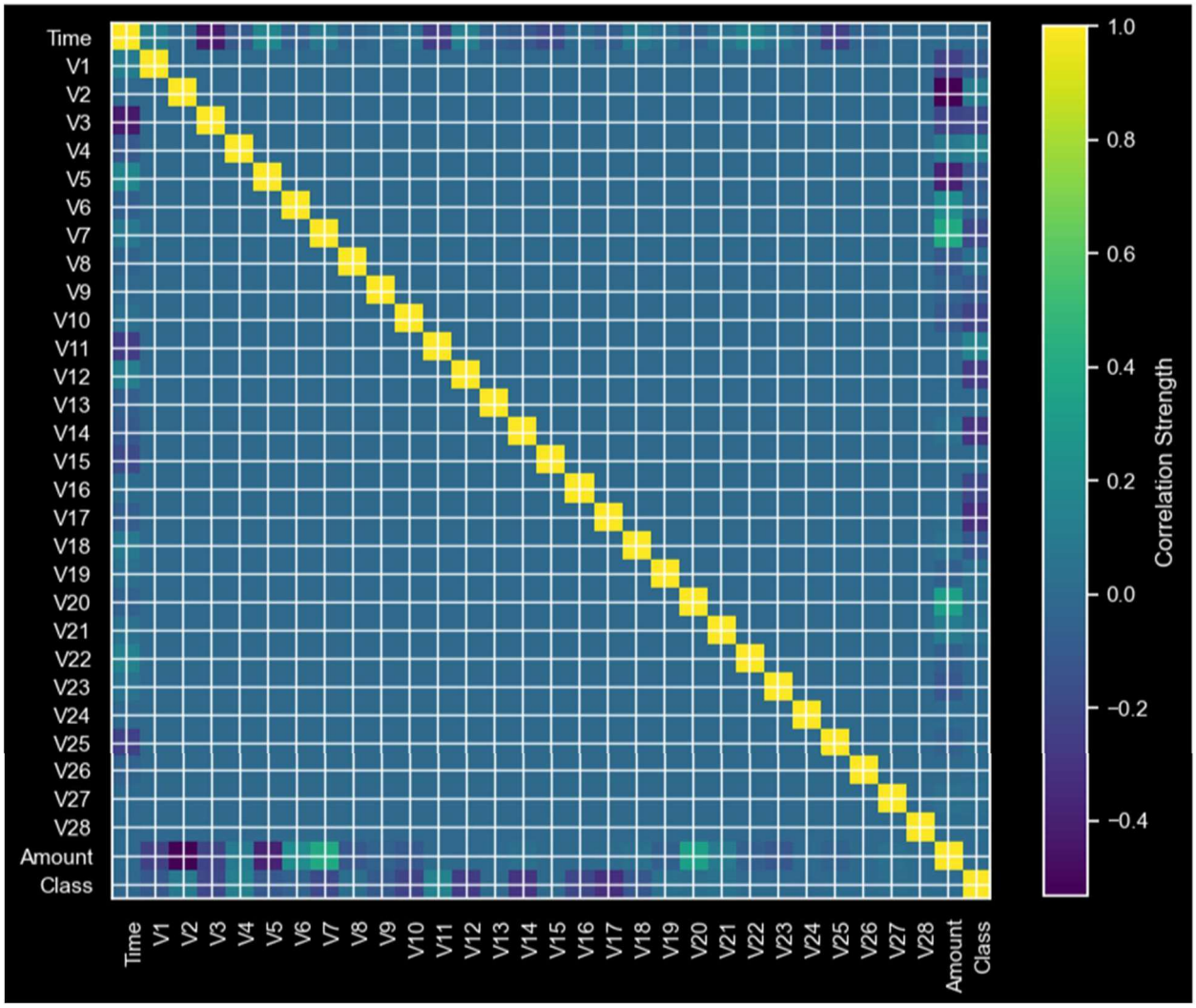
Time vs Class Scatter Plot.

The distribution of fraudulent transactions over time offers no newinformation because both fraudulent and non-fraudulent transactions are spread out over time.

Amount vs Class Scatter Plot

It is obvious that transactions with little amounts are more likely to be fraudulent than those with huge amounts.

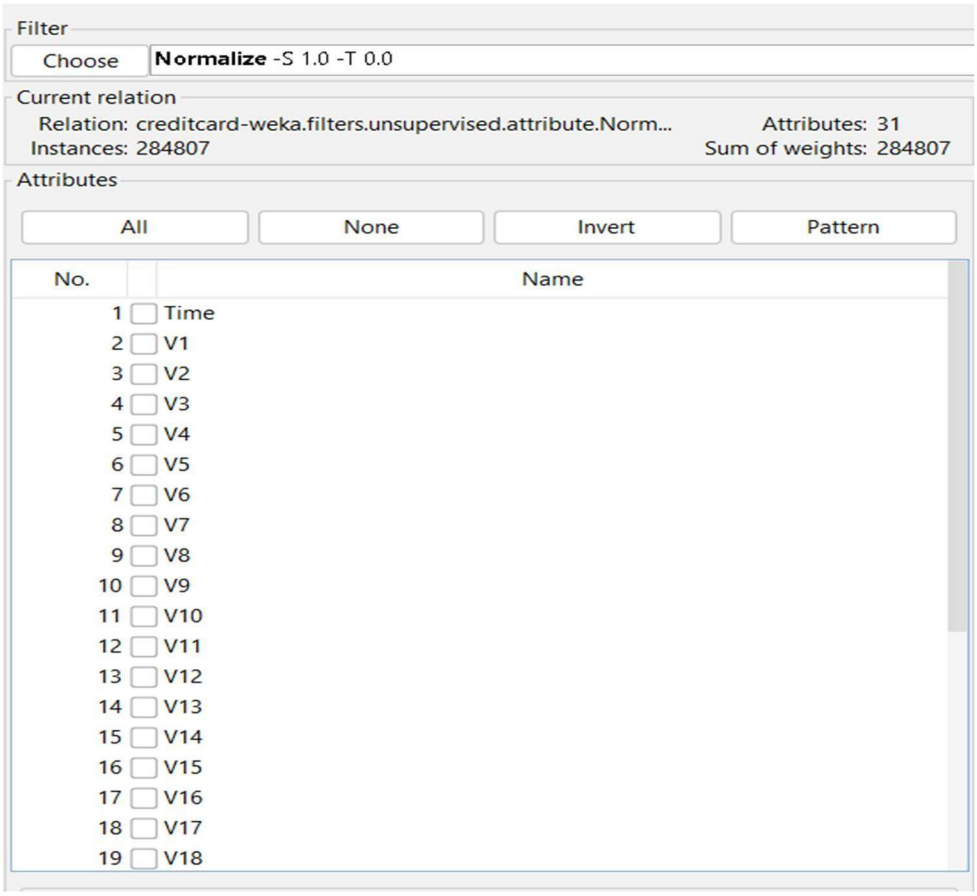
HEATMAP:



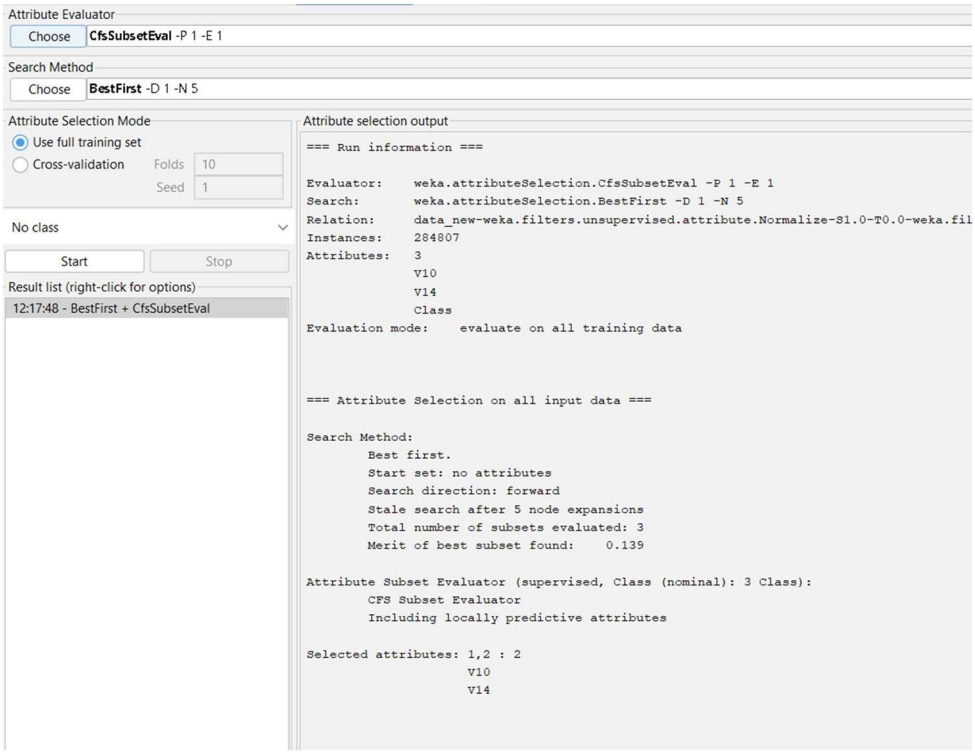
All the features have correlation < 0.8 with our Target Class.

Normalization in Weka

The primary purpose of normalization is to bring all features or variables to a similar scale or range. This is important because many machine learning algorithms are sensitive to the scale of the input features.



ATTRIBUTE SELECTION USING WEKA

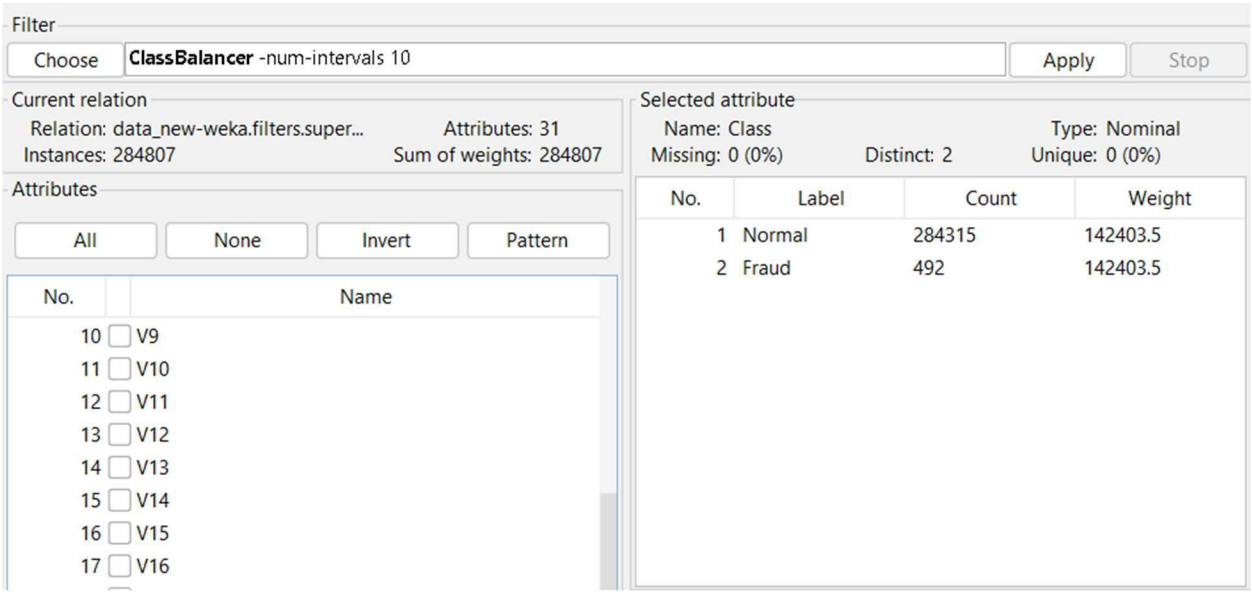


The attribute selection process performed using the CfsSubsetEval algorithm with the BestFirst search algorithm resulted in the selection of two attributes: V10 and V14. These attributes were determined to have the highest predictive power for the target variable (Class) among all the available attributes.

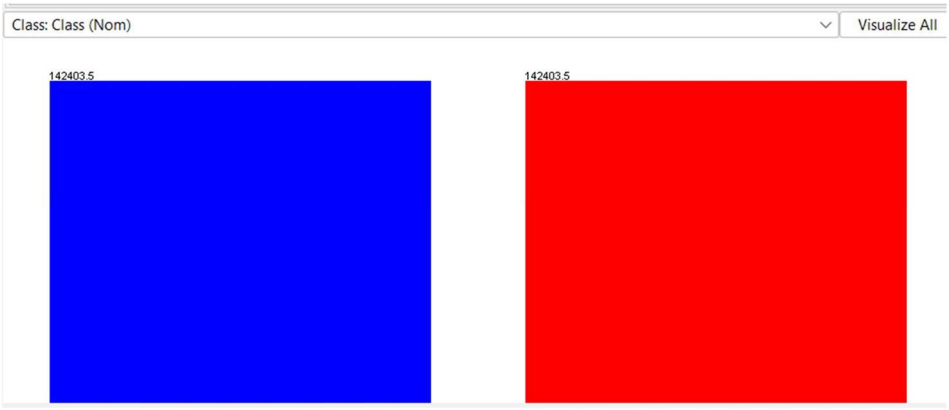
The selected attributes, V10 and V14, are the 11th and 15th attributes in dataset, respectively.

It's important to note that attribute selection algorithms aim to identify a subset of attributes that are most relevant to the target variable while discarding less informative or redundant attributes. The evaluation metric used in this case is the CFS (Correlation-based Feature Selection) merit, which measures the predictive power of each attribute individually as well as in combination with other attributes.

CLASS BALANCER: To balanced the imbalanced Data



DISTRIBUTION OF CLASS AFTER USING CLASSBALANCER.



By using the filter with class balancing, you can ensure that both the fraud and non-fraud classes are represented equally or according to the specified target distribution. This can help mitigate any bias in the model towards the majority class and improve the performance of fraud detection algorithms.

MODEL IMPLEMENTATION

Here are brief explanations of the Naive Bayes, SMO, and J48 models in Weka

- Naive Bayes: Naive Bayes classifiers are based on Bayes' theorem, which is a mathematical formula that can be used to calculate the probability of an event occurring given the probability of other events that have already occurred. Naive Bayes classifiers assume that the features of a data set are independent of each other. This means that the probability of a feature being present in a data point is not affected by the presence or absence of other features. This assumption is often not true in practice, but it can still be a good approximation in many cases. Naive Bayes classifiers are relatively simple to train and they can be very effective for a variety of classification tasks.
- SMO: SMO stands for "Sequential Minimal Optimization." It is an algorithm for training support vector machines (SVMs). SVMs are a type of machine learning algorithm that can be used for both classification and regression tasks. SVMs work by finding a hyperplane that separates the data points into two classes. The SMO algorithm is a more efficient algorithm for training SVMs than the original SVM algorithm. This is because SMO only needs to optimize a subset of the parameters at each iteration, which can save a lot of time for large data sets.
- J48: J48 is a decision tree algorithm that is based on the ID3 algorithm. Decision trees are a type of supervised learning algorithm that can be used to classify data. Decision trees work by recursively partitioning the data set based on the values of the features. The J48 algorithm is a popular choice for classification tasks because it is relatively easy to understand and interpret. Decision trees can be used to visualize the relationship between the different features and the class labels. This can be helpful for understanding how the model makes its predictions.

Naïve Byes (Original Data)

```
=== Summary ===

Correctly Classified Instances       83504               97.7318 %
Incorrectly Classified Instances      1938                2.2682 %
Kappa statistic                          0.1292
Mean absolute error                      0.0227
Root mean squared error                  0.1491
Relative absolute error                626.59   %
Root relative squared error            330.6228 %
Total Number of Instances            85442

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.978    0.149    1.000      0.978    0.989      0.243    0.964     1.000     Normal
                 0.851    0.022    0.072      0.851    0.132      0.243    0.968     0.091     Fraud
Weighted Avg.    0.977    0.149    0.998      0.977    0.987      0.243    0.964     0.998

=== Confusion Matrix ===

     a     b   <-- classified as
 83356  1912 |     a = Normal
    26   148 |     b = Fraud
```

The evaluation on the test split of the data shows that the classifier correctly classified 97.73% of the instances as Normal or Fraud. However, there were 1938 instances (2.27%) that were incorrectly classified. The Kappa statistic, which measures the agreement between the classifier's predictions and the actual classes, is reported as 0.1292. The mean absolute error and root mean squared error are also provided.

The evaluation further breaks down the performance by class. For the Normal class, the true positive rate (TP Rate) is 0.978, indicating that 97.8% of Normal instances were correctly classified. The false positive rate (FP Rate) is 0.149, meaning that 14.9% of Fraud instances were incorrectly classified as Normal. The precision, recall, and F-measure for both classes are given.

The confusion matrix shows the number of instances classified correctly (diagonal elements) and the misclassifications (off-diagonal elements). In this case, 83,356 instances of the Normal class were correctly classified, while 1,912 were misclassified as Fraud. Similarly, 148 instances of the Fraud class were correctly classified, and 26 instances were misclassified as Normal.

In summary, the Naive Bayes classifier achieved relatively high accuracy on the test split, but it seems to struggle with correctly identifying instances of the Fraud class.

SMO (ORIGINAL DATA)

```
=== Summary ===

Correctly Classified Instances       85391               99.9403 %
Incorrectly Classified Instances        51                0.0597 %
Kappa statistic                          0.8388
Mean absolute error                      0.0006
Root mean squared error                  0.0244
Relative absolute error                 16.4433 %
Root relative squared error             54.1917 %
Total Number of Instances            85442

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.236    1.000      1.000    1.000      0.843    0.882     1.000     Normal
                 0.764    0.000    0.930      0.764    0.839      0.843    0.882     0.711     Fraud
Weighted Avg.    0.999    0.235    0.999      0.999    0.999      0.843    0.882     0.999

=== Confusion Matrix ===

     a     b   <-- classified as
 85258    10 |     a = Normal
    41   133 |     b = Fraud
```

This is a binary classification model built using the SMO algorithm. The model is trained on a dataset consisting of normalized features, including time, V1-V28 (anonymous variables), and amount. The model achieves an accuracy of 99.94% on the test split with 85391 correctly classified instances out of 85442 total instances. The confusion matrix shows that there were 10 false positives (normal transactions classified as fraud) and 41 false negatives (fraudulent transactions classified as normal). The precision and recall for the fraud class are 0.930 and 0.764 respectively, indicating that the model has a high precision in identifying fraud but may miss some fraudulent transactions.

.

J48 MODEL (ORIGINAL DATA)

```
=== Summary ===

Correctly Classified Instances       85398               99.9485 %
Incorrectly Classified Instances        44                0.0515 %
Kappa statistic                          0.8622
Mean absolute error                      0.0008
Root mean squared error                  0.0226
Relative absolute error                 22.5454 %
Root relative squared error             50.1043 %
Total Number of Instances            85442

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.207    1.000      1.000    1.000      0.866    0.885     1.000     Normal
                 0.793    0.000    0.945      0.793    0.862      0.866    0.885     0.775     Fraud
Weighted Avg.    0.999    0.206    0.999      0.999    0.999      0.866    0.885     0.999

=== Confusion Matrix ===

     a     b   <-- classified as
 85260     8 |     a = Normal
    36   138 |     b = Fraud
```

The detailed accuracy by class provides additional metrics for each class (Normal and Fraud), including the true positive rate, false positive rate, precision, recall, F-measure, Matthew's correlation coefficient (MCC), ROC area, and PRC area.

The confusion matrix provides a breakdown of the actual and predicted classifications, with the rows representing the actual classes and the columns representing the predicted classes. In this case, the model correctly classified 85260 instances as Normal and 138 instances as Fraud, with 8 Normal instances misclassified as Fraud and 36 Fraud instances misclassified as Normal.

Overall, the J48 model appears to perform well on this dataset, with a high percentage of correctly classified instances and relatively low error metrics. However, further analysis and testing may be required to ensure the model's accuracy and generalizability.

Naïve Bayes Percent Split 70% with Balance Class

```
=== Summary ===

Correctly Classified Instances       85568.8963          91.9404 %
Incorrectly Classified Instances      7501.0942           8.0596 %
Kappa statistic                          0.8392
Mean absolute error                      0.0797
Root mean squared error                  0.2785
Relative absolute error                 15.8798 %
Root relative squared error             55.4827 %
Total Number of Instances            93069.9905

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.973    0.126    0.867      0.973    0.917      0.845    0.968     0.961     Normal
                 0.874    0.027    0.975      0.874    0.921      0.845    0.968     0.969     Fraud
Weighted Avg.    0.919    0.072    0.925      0.919    0.920      0.845    0.968     0.965

=== Confusion Matrix ===

     a        b      <-- classified as
 41574.32  1133.46 |        a = Normal
  6367.64 43994.58 |        b = Fraud
```

The Naive Bayes model for credit card fraud detection(balance) shows promising results with an accuracy of 91.94% on the test split. The model's performance is reasonable, but it is important to interpret the results with caution. While the precision and recall values are relatively high for both the normal and fraud classes, it is crucial to consider the trade-off between the two metrics. The model's generalization ability may depend on the specific characteristics of the dataset used and may not necessarily apply to other datasets or real-world scenarios. Further analysis and evaluation with different algorithms and datasets are necessary to validate the model's effectiveness. In conclusion, the Naive Bayes classifier demonstrates potential for credit card fraud detection, but more comprehensive investigations are needed to draw definitive conclusions about its performance and generalizability.

SMO Percent Split 70% with Balance Class

```
=== Summary ===

Correctly Classified Instances       88819.3915          95.4329 %
Incorrectly Classified Instances      4250.5989           4.5671 %
Kappa statistic                          0.9085
Mean absolute error                      0.0457
Root mean squared error                  0.2137
Relative absolute error                  9.1043 %
Root relative squared error             42.5682 %
Total Number of Instances            93069.9905

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.982    0.069    0.924      0.982    0.952      0.910    0.956     0.915     Normal
                 0.931    0.018    0.984      0.931    0.957      0.910    0.956     0.953     Fraud
Weighted Avg.    0.954    0.041    0.956      0.954    0.954      0.910    0.956     0.936

=== Confusion Matrix ===

     a        b      <-- classified as
 41930.43   777.34 |        a = Normal
  3473.26 46888.96 |        b = Fraud
```

The model trained using the SMO algorithm in WEKA shows promising results in classifying credit card transactions as normal or fraudulent(Balance). The attributes with the highest weights indicate their relative importance in the classification process. The model achieved an overall accuracy of 95.43% on the test set, with a Kappa statistic of 0.9085, indicating substantial agreement beyond chance. The precision, recall, and F-measure for both classes were also high, suggesting good performance for both normal and fraud transactions. However, it's important to note that the evaluation results are based on a specific test split, and generalization to new and unseen data should be approached cautiously. Further validation on different datasets and evaluation of real-world performance are necessary to ascertain the model's reliability and robustness. Overall, the model shows promise in detecting fraudulent credit card transactions, but its performance and generalization capabilities need to be further validated.

J48 Percent Split 70% with Balance Class

```
=== Summary ===

Correctly Classified Instances       83446.9125          89.6604 %
Incorrectly Classified Instances      9623.078           10.3396 %
Kappa statistic                          0.7953
Mean absolute error                      0.1036
Root mean squared error                  0.3215
Relative absolute error                 20.658  %
Root relative squared error             64.0444 %
Total Number of Instances            93069.9905

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.190    0.817      0.998    0.899      0.812    0.905     0.817     Normal
                 0.810    0.002    0.998      0.810    0.895      0.812    0.905     0.913     Fraud
Weighted Avg.    0.897    0.088    0.915      0.897    0.896      0.812    0.905     0.869

=== Confusion Matrix ===

     a        b      <-- classified as
 42636.15    71.62 |        a = Normal
  9551.45 40810.76 |        b = Fraud
```

In the evaluation results provided, the J48 classifier model achieved an accuracy of 89.66% on the test set. This is a good accuracy, and it suggests that the model is able to generalize well to new data. The Kappa statistic is also high, which indicates that the model is not simply overfitting the training data.

The confusion matrix shows that the model is slightly better at predicting normal transactions than fraudulent transactions. However, the model is still able to correctly classify a large majority of fraudulent transactions.

Overall, the J48 classifier model appears to be a good choice for this classification task. It is accurate, efficient, and easy to understand.

Conclusion:

Based on the project of credit card fraud detection using data mining techniques, conducted experiments with various classifiers before and after balancing the dataset.

Before balancing the dataset, the J48 classifier performed the best among the models tested. It demonstrated strong performance in accurately classifying fraudulent and non-fraudulent transactions, considering the available features.

However, after balancing the dataset model emerged as the best performer is SMO.

The SMO classifier proved to be a robust model for credit card fraud detection after balancing the dataset. By optimizing the support vector machine algorithm, it successfully identified fraudulent activities while minimizing false positives.

Overall, the experiments highlighted the importance of dataset balancing in credit card fraud detection. Before balancing, the J48 classifier performed well, but after addressing the class imbalance, SMO classifier outperformed the J48 classifier.

It is important to note that the conclusion may vary depending on the specific data set and the evaluation metrics used. Further exploration and experimentation with other classifiers and techniques could potentially yield even better results for credit card fraud detection**.**