**Individual Report of Data Analysis using CRISP-DM methodology**

CSC40080

Statistical data analytics and databases

**Table of contents**

[**Introduction and Business Context**](#_vhmwi7nr4ape) **3**

[**Data Selection and Pre-Processing**](#_datq9k2shbbg) **4**

[**Machine Learning Method(s) and their Implementation**](#_p1eqahd4jmao) **7**

[**Evaluation of Results**](#_prwdeb1nrkxa) **8**

[ZeroR](#_m3pct8gezwhz) 8

[Naive Bayes](#_n9im00rh1l99) 9

[**References**](#_sygu4vas7r65) **10**

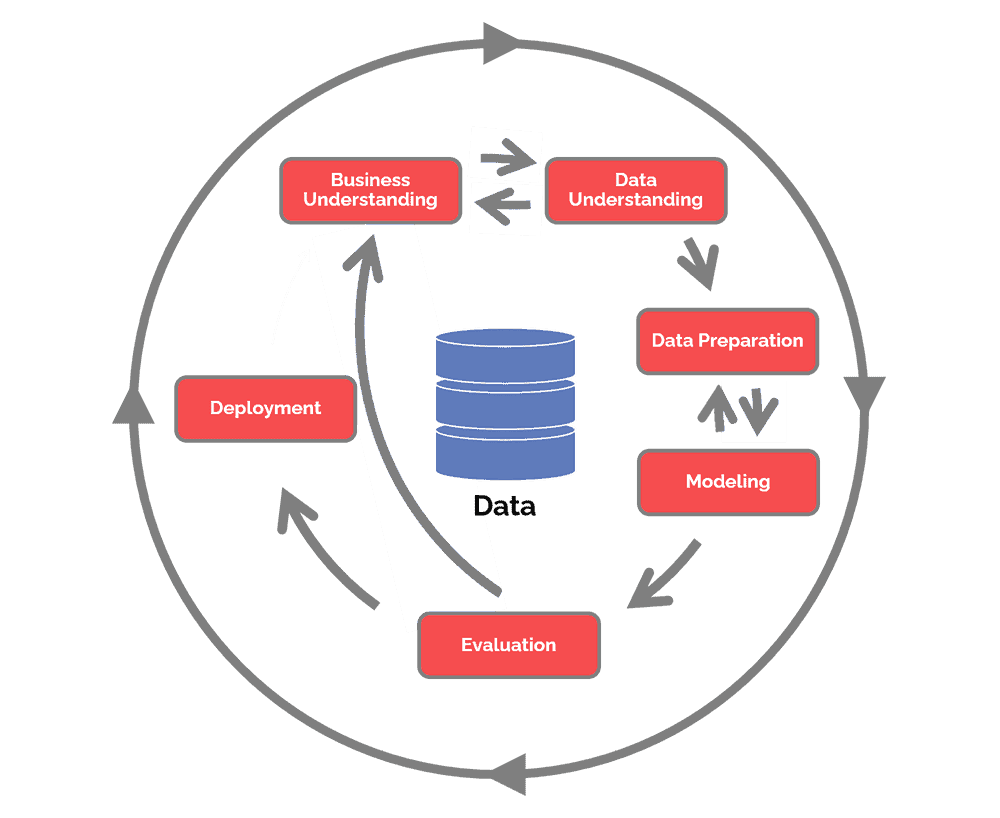
[**Appendices**](#_rbhgjd5c91cl) **12**

#### 

#### **Introduction and Business Context**

It is incredibly important for banks and credit lenders to be able to identify fraudulent activity when it happens so that the card can be blocked and the account can be secured until the issue is resolved. The company does not currently use the CRISP-DM methodology or machine learning in order to identify fraudulent transactions. At present fraudulent transactions are identified by customers and our fraud teams manually checking customers accounts and responding to alerts raised when transactions occur above a set amount.

Machine learning refers to a field of study that is focused on building and understanding algorithms use data to improve performance on a set of tasks (Mitchell, 1997). Studies have shown that machine learning can be effectively used to take credit card fraud training datasets and identify cases of fraud in testing datasets (Perols, 2011), the Naive Bayes model has been shown to perform particularly well on these datasets (Varmedja et. al., 2019; Awoyemi et. al., 2017; Itoo and Singh., 2021).

CRISP-DM stands for Cross Industry Standard Process for Data Mining (Shearer, 2000), CRISP-DM defines six major phases of data mining (Harper, 2006), they are business understanding, data understanding, data preparation, modelling, evaluation, and deployment. CRISP-DM is a cyclical process, the sequence is not strict and there is frequent moving back and forth between phases as the figure to the right depicts. Using this methodology allows for the business understanding to inform and be informed by the data understanding aswell as to be informed by the evaluation of the machine learning models.

| Figure 1: Data Science Process Alliance, 2021 |
| --- |

By implementing machine learning with the CRISP-DM methodology to identify cases of fraud, fraudulent cases could be identified faster and stopped. Hidden patterns can be identified, increasing business understanding and fraudulent transactions can be prevented before they occur. The company will also be able to measure and improve the performance of different features deployed to protect our customers against fraud.

The ultimate goal of this task is to determine the efficacy of the CRISP-DM methodology with machine learning to improve the business understanding of credit card fraud and to better identify the fraudulent transactions taking place, with a view to implementing this across the company. More employees will be available to identify customers of fraudulent transactions on their accounts as they occur, therefore, protecting our customers.

#### **Data Selection and Pre-Processing**

The credit card fraud dataset was obtained from Kaggle (Machine Learning Group at ULB, 2018). It was originally collected and analysed through collaboration between Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles). This dataset consists of credit card transactions made across 2 days in september 2013 by european card holders. This dataset is highly unbalanced due to the nature of fraud being an extreme minority of transactions. The dataset contains 284807 transactions and of these transactions 492 were fraudulent, accounting for 0.172% of all the transactions in the dataset.

The ‘Time’ feature gives the seconds elapsed between each transaction and the first transaction in the dataset. The ‘Amount’ feature gives the the amount of the transaction. The features V1 through V28 are the principal features obtained through PCA transformation, and contain only numerical values due to the PCA transformation, The original features that V1-V28 represent cannot be disclosed to protect the confidentiality of the cardholders. The ‘Class’ feature identifies if the transaction was fraudulent, the value 1 represents a fraudulent transaction.

| Instances | 284807 |
| --- | --- |
| Missing values | 0 |
| Outliers | 48,314 |

In weka, replace missing values was used to change these values to NULL, however no missing values were identified. I used interquartile range in weka to identify the outliers with the value of 3 and the equations Q3 + OF\*IQR < x <= Q3 + EVF\*IQR or Q1 - EVF\*IQR <= x < Q1 - OF\*IQR. From this 48314 outliers were identified, however these were not removed as doing so would have removed all fraudulent transactions, rendering the dataset unusable.

Below is the summary table for the instances, attributes and sum of weights for the testing set and training sets, See appendix A for training and testing set visualisation.

| Summary |  |  |
| --- | --- | --- |
|  | Training Set | Testing set |
| Instances | 199365 | 85442 |
| Attributes | 31 | 31 |
| Sum of Weights | 199365 | 85442 |

The dataset was randomised in weka and divided into training and testing sets with a 70/30 split. The training set (in-sample) contained 199365 instances, and the testing set (out of sample) contained 85,442 instances.

The summary statistics for the time feature are tabled below.

| Feature: Time |  |  |
| --- | --- | --- |
|  | Training Set | Testing Set |
| Distinct | 107302 | 63602 |
| Unique | 51851 (26%) | 46269 (54%) |
| Minimum | 0 | 0 |
| Maximum | 172792 | 172787 |
| Mean | 94935.31 | 94530.474 |
| StdDev | 47509.313 | 47437.788 |

This table shows that the testing and training sets are evenly matched.

The means (measures of centre) are similar, although not identical, the training set has a slightly higher centre point, the difference is not significant with a difference of 404.836 only. The standard deviation (measures of variability) are also similar, with a difference of only 71.525, it is not a significant difference. Meaning that the two sets are well matched on this feature, although there is slightly more variability and a higher centre point in the training set this is to be expected in a larger dataset.

The summary statistics for the amount feature are tabled below.

| Feature: Amount |  |  |
| --- | --- | --- |
|  | Training Set | Testing Set |
| Distinct | 27374 | 17512 |
| Unique | 13927 (7%) | 10079 |
| Minimum | 0 | 0 |
| Maximum | 19656.53 | 25691.16 |
| Mean | 87.521 | 90.284 |
| StdDev | 244.228 | 263.348 |

For this feature the difference between the means is 2.763 making the two sets very similar, with the testing set having a slightly higher centre point. The difference in standard deviation between the two sets is 19.12 with the testing set having more variation than the training set, but the difference still remaining insignificant. For this feature the two sets are well matched.

The summary statistics for the class feature are tabled below.

| Class |  |  |
| --- | --- | --- |
|  | Testing Set | Training Set |
| Distinct | 2 | 2 |
| Unique | 0 (0%) | 0 (0%) |
| Minimum | 0 | 0 |
| Maximum | 1 | 1 |
| Mean | 0.002 | 0.002 |
| StdDev | 0.042 | 0.04 |

Given that the class feature has only 0 or 1 as the value, we can expect the summary statistics to be very similar. The means for these sets are identical, and the standard deviation only has a difference of 0.002, meaning that the training set has more variation which is to be expected as the training set has more instances than the testing set. This does tell us that the ratio of fraudulent to non-fraudulent transactions must be similar both datasets.

The training and testing sets do not overlap, both sets are highly unbalanced due to fraud being the extreme minority in the data.

The preprocessing that was performed included using replace missing values and using interquartile range within weka to replace any missing values with ‘null’ and to identify any outliers and extreme values. No missing values were identified. Whilst outliers and extreme values were identified within the dataset, these were not removed. Removing the outliers and extreme values would have compromised the entire dataset as the outliers and extreme values are linked to all of the instances of fraud. Removing outliers and/or extreme values would have removed all instances of fraud within the dataset rendering the dataset unusable. Randomisation was performed on the data. The datasets were split 70% training and 30% testing within weka.

#### **Machine Learning Method(s) and their Implementation**

Since the problem of credit card fraud is a classification problem the machine learning model that will be used will be a classifier. As mentioned in the introduction, Naive Bayes has been shown to perform very well on this classification task and often out performs the other models it is tested against (Varmedja et. al., 2019; Awoyemi et. al., 2017; Itoo and Singh., 2021). To compare the model against I will be using the ZeroR model, as it is a simple classifier and performs well as a baseline to compare to (Peters, 2013). Both models will be trained with the training dataset with a 10-fold cross-validation.

The naive bayes classifier is a classifier that applies the bayes theorem( P(A/B)=P(B/A)P(A)/P(B)) (Stuart, and Ord.,1994) with strong independence assumptions between features. This classifier belongs to a group of related classifiers called probabilistic classifiers (Hastie et. al., 2009) which also includes logistic regression and multi-layer perceptrons. The naive bayes classifier will assume that every feature in the dataset is independent from other features, for this dataset it will assume that time, amount and V1-V28 are all independent from one another but all contribute to the probability that the value of the ‘class’ feature is 1 (fraud). The advantages of using Naive Bayes are that it doesnt require much training data, despite plenty being provided in this case, it can handle both continuous and discrete data (Shah, 2021), it is very fast to train, if all features are independent from one another it can provide very accurate results (MLNerds, 2021). The disadvantages are that the Naive Bayes conditional independence assumption assumes all features are independent which doesnt always reflect real-life as often features are related to one another (Shah, 2021). Additionally this model has a zero probability problem where if there are no occurrences of a certain attribute and value together in the training set, but they are presented in the testing set, the probability estimate will be zero and will produce zero when all probabilities are multiplied together (Chauhan, 2022).

Zero R is the simplest classification method and if often used as a benchmark to test other classification methods against. ZeroR ignores all predictors and relies on the target only, Zero R is unable to predict whether a specific instance is fraudulent or not, it is only able to present the majority class (Sadawi, 2014).

Weka was used to perform all preprocessing, dataset splitting, model training and testing. The parameters for the Naive Bayes classifier were, batch size = 50, number of decimal places = 2, and with debug; display model in old format; do not check capabilities; and use supervised discretization all set to false, and use kernel estimator set to true. The parameters for the ZeroR model were batch size = 50, number of decimal places = 2, with debug and do not check capabilities set to false.

Cross Validation (10 fold) was used in the training of both of the models followed by a full test on the testing set. Cross-validation was used as it provides a more accurate model, meaning more accurate results when testing on out of sample data (GeeksforGeeks, 2017). The Cross-validation also means that the model was tested on one of the fold sections making it act as a validation set, helping to determine the error on the unseen data, allowing for the control and tweaking of the model therefore minimising the likelihood of overtraining.

#### **Evaluation of Results**

Due to the dataset being highly imbalance, RMSE and Confusion Matrices will be used to present results and assess performance, as usual metrics such as accuracy do not measure the results from imbalance datasets well.

##### ZeroR

ZeroR was run with a batch size of 50, due to being an incredibly simple model we can expect to see plenty of error in this models confusion matrix. The model’s metrics will show that it performed well because it was able to identify the majority value within the class.

|  | Train | Test |
| --- | --- | --- |
| RMSE | 0.403 | 0.42 |

Confusion Matrix

| Train |  |  | Test |  |
| --- | --- | --- | --- | --- |
| **0** | **1** |  | **0** | **1** |
| 199012 | 0 |  | 85303 | 0 |
| 353 | 0 |  | 139 | 0 |

Looking at the RMSE the model performed better in training than it did on the test, this is always to be expected with training and testing models.

Looking at the confusion matrix we can see that ZeroR is working how it should, in both training and testing it has correctly predicted that 0 (non-fraud) will be the majority, and as we can see the fraud cases for this model have all been classified as false negatives (type 2 error), because ZeroR only predicts the majority class.

##### Naive Bayes

Initially the first set of training was run with 10 fold cross validation with all the default parameters, in the table below this is labelled Train 1. In Train 2 batch size was increased to 200 which led to overfitting. In Train 3 batch size was decreased to 50 which led to better results. Train 4 the batch size remained at 50 and was tested with the kernel estimator set to true. Train 4 produced the best results. The model was then tested.

|  | Train 1 | Train 2 | Train 3 | Train 4 | Test |
| --- | --- | --- | --- | --- | --- |
| RMSE | 0.9772 | 0.0035 | 0.1056 | 0.2805 | 0.4931 |

Confusion Matrix

| Train 1 |  |  | Train 2 |  |  | Train 4 |  |  | Test |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** |  | **0** | **1** |  | **0** | **1** |  | **0** | **1** |
| 7808 | 191204 |  | 199012 | 0 |  | 194250 | 4762 |  | 81655 | 3648 |
| 0 | 353 |  | 0 | 353 |  | 61 | 292 |  | 29 | 103 |

Train 1 had many errors, which is reflected across the results presented, it is clear that here with the default parameters the model was underfitting. In Train 2 the model has too few errors infact it performed perfectly which shows that it was overfitting the data. Train 4 shows much better performance of the model, it is no longer over or underfitting, it is fitting the data well, this can be attributed to the parameter tuning, with the batch size decreased and the kernel estimated set as true. The Test run shows that the model performed worse than train 4, but this is to be expected as it is performing on unseen data, and the performance can still be described as good.

Statistically speaking, by only comparing the RMSE, it appears as though the ZeroR model is the better model as on testing the error rate is lower. However ZeroR is not the more useful model. By looking at the confusion matrices we can see that Naive Bayes performs significantly better than the Zero R model did. Naive Bayes is the better model for the task, Zero R was chosen as the baseline to test against. It can be seen clearly how much better the performance of Naive Bayes is solely based off the fact that it is not just choosing the majority value for the feature. Naive bayes performed well.

Moving forwards with this task, I would suggest moving away from Weka as using an IDE with python would allow for more precise tweaking of parameters and hyper parameters, additionally for this task Weka did not perfom as well as another IDE might, Weka crashed repeatedly through testing, it was intended that more classification models would be included, however Weka was unable to handle this. This classification task does not require a complex model to perform well, it may also be performed well by a support vector machine model or a multi-layer perceptron model. It is clear that machine learning with CRISP-DM has the ability to aid in developing anti-fraud tools for the company and to improve business understanding. Naive Bayes was able to correctly classify a significant proportion of the test data, showing that there are patterns within the data that the model was able to identify to classify a transaction as fraudulent or not.

#### **References**

Awoyemi, J.O., Adetunmbi, A.O. and Oluwadare, S.A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis.  *International conference on computing networking and informatics (ICCNI)* (pp. 1-9). IEEE.

Chauhan, N. (2022). *Naïve Bayes Algorithm: Everything You Need to Know*. [online] Available at: https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html.

Data Science Process Alliance (2021). *CRISP-DM*. [online] Available at: https://www.datascience-pm.com/crisp-dm-2/.

GeeksforGeeks. (2017). *Cross Validation in Machine Learning*. [online] Available at: <https://www.geeksforgeeks.org/cross-validation-machine-learning/>.

Harper, Gavin; Stephen D. Pickett (2006). [Methods for mining HTS data](https://caridokumen.com/download/methods-for-mining-hts-data-_5a462410b7d7bc7b7af27f4a_pdf). [*Drug Discovery Today*](https://en.wikipedia.org/wiki/Drug_Discovery_Today). 11 (15–16): 694–699. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1016/j.drudis.2006.06.006](https://doi.org/10.1016%2Fj.drudis.2006.06.006). [PMID](https://en.wikipedia.org/wiki/PMID_(identifier)) [16846796](https://pubmed.ncbi.nlm.nih.gov/16846796)

Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009). [*The Elements of Statistical Learning*](https://web.archive.org/web/20150126123924/http://statweb.stanford.edu/~tibs/ElemStatLearn/). p. 348

Itoo, F. and 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), pp.1503-1511.

MLNerds (2021). *Naive Bayes Classifier : Advantages and Disadvantages*. [online] Machine Learning Interviews. Available at: <https://machinelearninginterview.com/topics/machine-learning/naive-bayes-classifier-advantages-and-disadvantages/>.

Machine Learning Group at ULB (2018). *Credit Card Fraud Detection*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

Mitchell, Tom (1997). *Machine Learning*. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.

Perols, J. (2011). Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, *30*(2), pp.19-50.

Peters, C.A. (2013). *Intrusion and fraud detection using multiple machine learning algorithms*. University of Manitoba (Canada).

Sadawi, N. (2014). *The ZeroR Classifier .. What it is and How it Works*. *YouTube*. Available at: https://www.youtube.com/watch?v=kUbYN4AcPmA.

Shah, R. (2021). *Naïve Bayes Algorithm’s Advantages and Disadvantages | Data Science and Machine Learning*. [online] Kaggle. Available at: https://www.kaggle.com/getting-started/225022.

Shearer C. .(2000) *The CRISP-DM model: the new blueprint for data mining*, J Data Warehousing ; 5:13—22

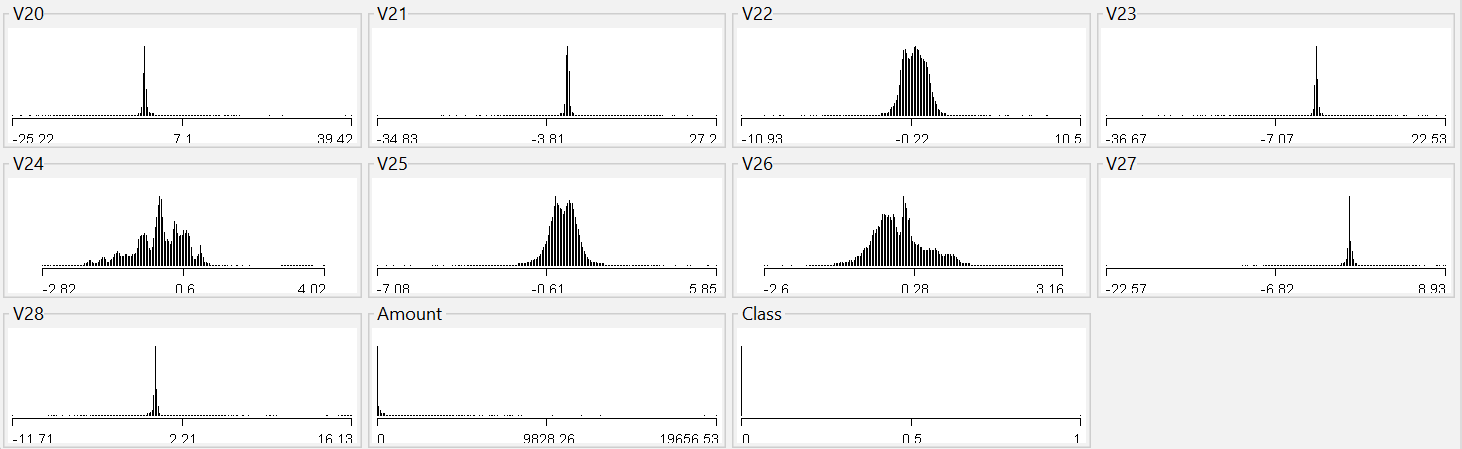
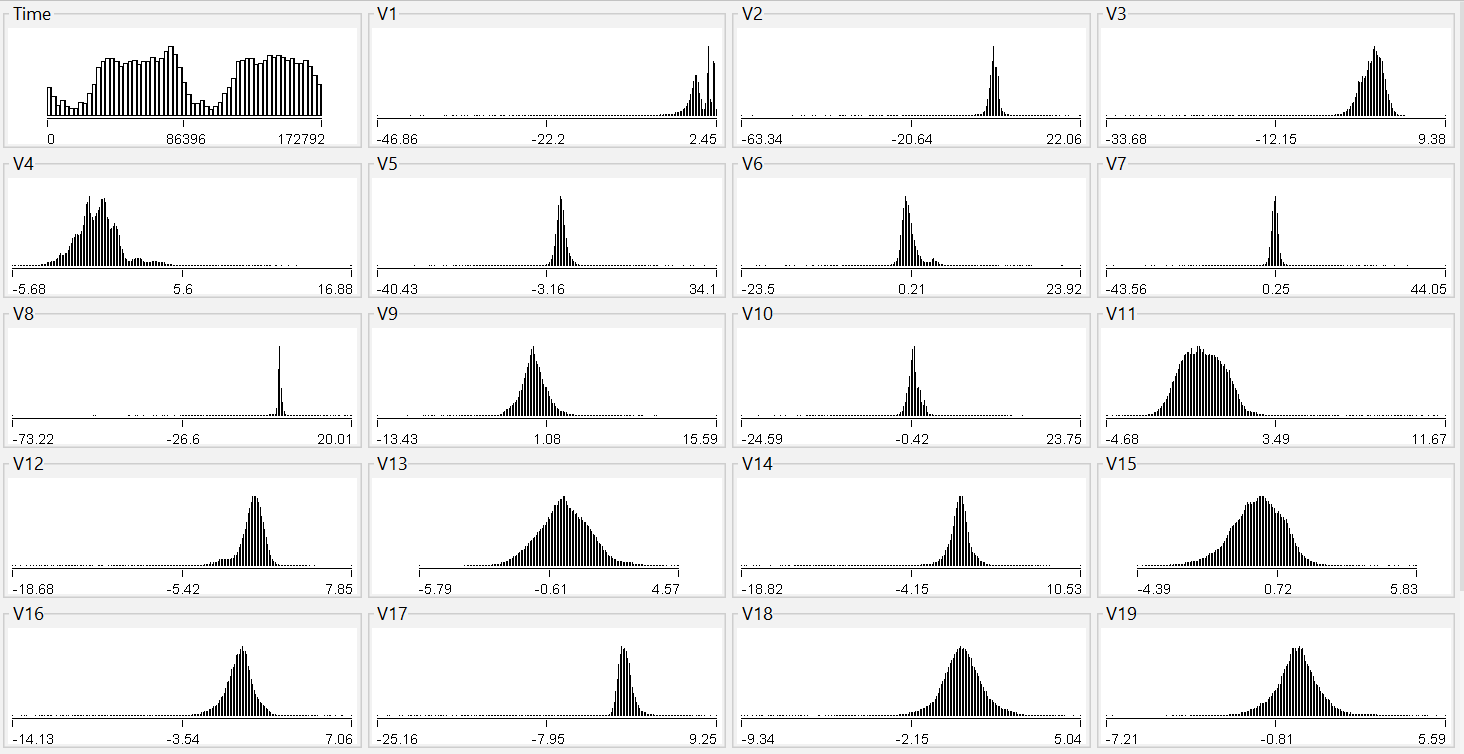
Stuart, A.; Ord, K. (1994). *Kendall's Advanced Theory of Statistics: Volume I—Distribution Theory*, [Edward Arnold](https://en.wikipedia.org/wiki/Edward_Arnold_(publisher))

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

#### **Appendices**

Appendix A: Training and Testing set Summary Visualisation

Training dataset



Testing / Out of sample set

