Artificial Intelligence (AI) Solution Implementation

Executive Summary

AI can be used to address credit card fraud for YourMoney building society by detecting unusual transactions to flag as potentially fraudulent. A dataset has been selected from a public repository and various classification algorithms tested on it. Decision tree was the best performing algorithm using the original imbalanced dataset but Support Vector Machine (SVM) with Pearson VII Universal Kernel (PUK) was best after the dataset was balanced. The recommendation is to develop an SVM with PUK algorithm using YourMoney historic data and deploy it to production. It will start in passive mode, monitoring and reporting, and will move into active mode to intercept suspected fraudulent transactions once YourMoney is confident that it is performing as it should.

Business Context

UK credit card fraud losses amounted to £574.2m in 2020 (UK Finance, 2021), making fraud detection a good AI use case. It is proposed to use a supervised learning classification algorithm to monitor credit card transactions in real-time, looking for indicators that a transaction is fraudulent. If a transaction is highlighted as potentially fraudulent it would not be authorised and the card would be temporarily stopped while the customer is contacted to confirm if the transaction was legitimate. This approach is not considered to have a radical consequence because the customer retains control, hence ”black box” algorithms that are not entirely explainable can be used because AI needs to be explainable “when consequences are radical“ (Gill, 2021).

Dataset

It was recommended in the original report to test the algorithms on historic data supplied by YourMoney. That data is not currently available so to demonstrate if a supervised learning algorithm would be suitable for credit card fraud detection, and if so which one, a dataset from a public repository was used.

Four criteria were considered when selecting the dataset, in descending order of importance:

1. **Provenance**. Ideally, the data will be real data. If only synthetic data can be found it should be generated from a trusted simulator.
2. **Volume**. A larger volume of data is preferred to train and test the algorithms effectively.
3. **Accurately balanced**. Real-world data is highly imbalanced. UK credit card spending in 2020 was £162.78bn (de Best, 2022), of which £574.2m was lost to fraud (UK Finance, 2021). This means 0.34% of total spending was lost to fraud.
4. **Understandable feature set**. Understanding the features in the data is desirable to make general observations about the data and for feature engineering. This might not be possible with real data if it has been anonymised.

The dataset from Kaggle (N.D.a) was chosen because it meets the three most important criteria:

1. **Provenance**. It is real data taken from actual European credit card transactions over two days in September 2013.

2. **Volume**. It contains 284,807 transactions which were considered plenty, even considering the highly imbalanced class values. There were datasets with higher volumes; over a million rows, but they were all simulated data and it was felt having real data was more important than a bigger volume of simulated data.
3. **Accurately balanced**. Out of a total of 284,807 transactions, 492 (0.172%) were fraudulent. The fraudulent transactions had a total value of 60,128 (currency not provided) out of a total of 25,162,590, which is 0.24%. This is broadly representative of the UK's actual percentage of 0.34%.

The fourth criterion was not met with this dataset. Because real data was used, all except two features (time and amount) were anonymised using PCA transformation to preserve confidentiality. This makes feature engineering with this data impossible. For example, it is known that the majority of credit card fraud is remote, where the cardholder is not present (UK Finance, 2021), but we cannot tell this from the PCA-transformed data.

The only editing of the data was converting class from binary to nominal values to enable WEKA to process it where nominal class values are required, and also to make the results easier to read. “1” was changed to “Fraud” and “0” was changed to “Not Fraud”.

Development of the prediction model

Support Vector Machine (SVM) generally performs better than Decision Tree (DT) (Danso et al, 2014; Suhaimi & Abas, 2020), so SVM with Polynomial kernel, Pearson VII Universal Kernel (PUK) and Radial Basis Function (RBF) kernel were tested, as well as DT and Artificial Neural Network (ANN) to give options with lower and higher compute requirements respectively.

WEKA was used to build prediction models using the selected algorithms because WEKA is quick and simple to use with output that allows easy comparison. The CSV data file was loaded into WEKA. Default values for each algorithm were used aside from selecting different SVM kernels using the SMO classifier function.

Stratified 10-fold cross-validation was used because the dataset is highly imbalanced. K-fold cross-validation made use of all of the data in both training and testing, whilst stratification ensured the models were trained and tested using negative and positive class values (Raschka, 2018). Screenshots of the WEKA results are in Appendix A.

Performance of the algorithms

The five algorithms were each tested on the chosen dataset using stratified 10-fold cross-validation and the results were put into confusion matrices, as shown in figure 1.

C4.5 Decision Tree			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 284,285	False Positive (FP) 30	Not Fraud	Actual
False Negative (FN) 109	True Positive (TP) 383	Fraud	

SVM with Polynomial Kernel			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 284,251	False Positive (FP) 64	Not Fraud	Actual
False Negative (FN) 110	True Positive (TP) 382	Fraud	

SVM with Pearson VII Universal Kernel			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 284,260	False Positive (FP) 55	Not Fraud	Actual
False Negative (FN) 99	True Positive (TP) 393	Fraud	

SVM with Radial Basis Function Kernel			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 284,275	False Positive (FP) 40	Not Fraud	Actual
False Negative (FN) 310	True Positive (TP) 182	Fraud	

Artificial Neural Network			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 284,257	False Positive (FP) 58	Not Fraud	Actual
False Negative (FN) 94	True Positive (TP) 398	Fraud	

Figure 1. Confusion Matrices – Original Dataset

Next, the values from the confusion matrices were used to calculate various performance metrics described in Sokolova et al (2006), Hossin and Sulaiman (2015), Danso et al (2021) and Chicco and Jurman (2020).

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

DT had the highest accuracy, meaning that it made the most accurate predictions. However, that doesn't mean that it's the best algorithm. Accuracy is best when the datasets are symmetric (Ghoneim, 2019), which is not the case here.

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

DT also scored well on precision, which is the measure of how accurate the positive predictions were. This might not be best for credit card fraud detection though. Precision scores highly when there are fewer false positives, but it doesn't penalise false negatives, so the algorithm could be missing fraudulent transactions and still score highly for precision.

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN}$$

If the main objective of YourMoney is to identify as many genuinely fraudulent transactions as possible, it might be prepared to accept more false positives (false alarms) to achieve that. A better metric could therefore be recall, which measures how many of the total positive cases were predicted without being penalised for false positives. Indeed, Jayaswal (2020) suggests using recall for credit card fraud detection for precisely that reason. ANN scored best for recall, followed by SVM with PUK, with DT third.

$$F1\ Score = \frac{2(Precision * Recall)}{Precision + Recall}$$

F1 Score combines precision and recall. This is useful because either false positives or false negatives bring the score down, so it provides a well-balanced score of how many positives were correctly identified as well as how precisely they are grouped. Powers (2020: 39) notes that F1 "completely ignores TN which can vary freely without affecting the statistic", but that's not entirely true because changes in TN would be reflected in changes to FP or FN. DT scored best in F1, slightly ahead of ANN and SVM with PUK.

$$Specificity = \frac{TN}{TN + FP}$$

Specificity is a measure of how many of the negative cases were accurately predicted. It is similar to recall but for the negative class value. Higher scores are the result of fewer false positives, so specificity would be used where minimising false alarms is important. It is thought that YourMoney would prefer a few more false alarms if it meant identifying more true positives (fraudulent transactions), so specificity isn't considered to be a useful metric on its own for this use case, but for completeness, DT scored best. Specificity is, however, useful because it is one of the inputs to the next metric:

$$Geometric\ Accuracy = \sqrt{(Sensitivity + Specificity)}$$

Danso et al (2021) explain that geometric accuracy (GA), which is the square root of the product of sensitivity (which is the same as recall) and specificity, is good for imbalanced datasets, which is the case in this example. This metric includes all four quadrants of the confusion matrix and so provides a well-balanced view of the overall performance of the algorithm. ANN scored best in GA, followed by SVM with PUK and DT third.

$$MCC = \frac{\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN}}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}$$

Chicco and Jurman (2020) suggest Matthews Correlation Coefficient (MCC) is a more reliable metric for binary classification with imbalanced datasets so it is a good metric for this use case. DT scored best, ahead of ANN and SVM with PUK.

Error metrics can also be calculated to compare algorithms, but they were not used in this case. Error rate measures the ratio of incorrect predictions over the total sample (Hossin & Sulaiman, 2015), but the result is the inverse of accuracy so it added little value when accuracy has already been calculated:

$$Error\ rate = \frac{FP + FN}{TP + FP + TN + FN} = 1 - Accuracy$$

Mean Square Error (MSE) and Root Mean Square Error (RMSE) are used to measure the size of prediction errors for regression problems. Since a binary class with just two outputs; Fraud or Not Fraud, was used there is no variance in “how wrong” an incorrect prediction is, so these metrics were not relevant. The formulae for reference are:

$$MSE = \frac{1}{n}\sum_{j=1}^{n}\left(P_j - A_j\right) \qquad RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(P_j - A_j\right)^2}$$

All of the results, excluding error metrics, are in figure 2 with the best scoring algorithm for each metric highlighted in green.

Classifier	Correctly classified	Incorrectly classified	Accuracy	Precision	Recall/ Sensitivity	F1 Score	Specificity	Geometric Accuracy	MCC
DT	99.9512%	0.0488%	0.99951	0.927	0.778	0.846	0.99989	0.882	0.849
SVM Poly	99.9389%	0.0611%	0.99939	0.857	0.776	0.814	0.99977	0.881	0.815
SVM PUK	99.9459%	0.0541%	0.99946	0.877	0.799	0.836	0.99981	0.894	0.837
SVM RBF	99.8771%	0.1229%	0.99877	0.820	0.370	0.510	0.99986	0.608	0.550
ANN	99.9466%	0.0534%	0.99947	0.873	0.809	0.840	0.99980	0.899	0.840

Figure 2. Performance Metrics Results

The results are charted in figure 3 for visual comparison.

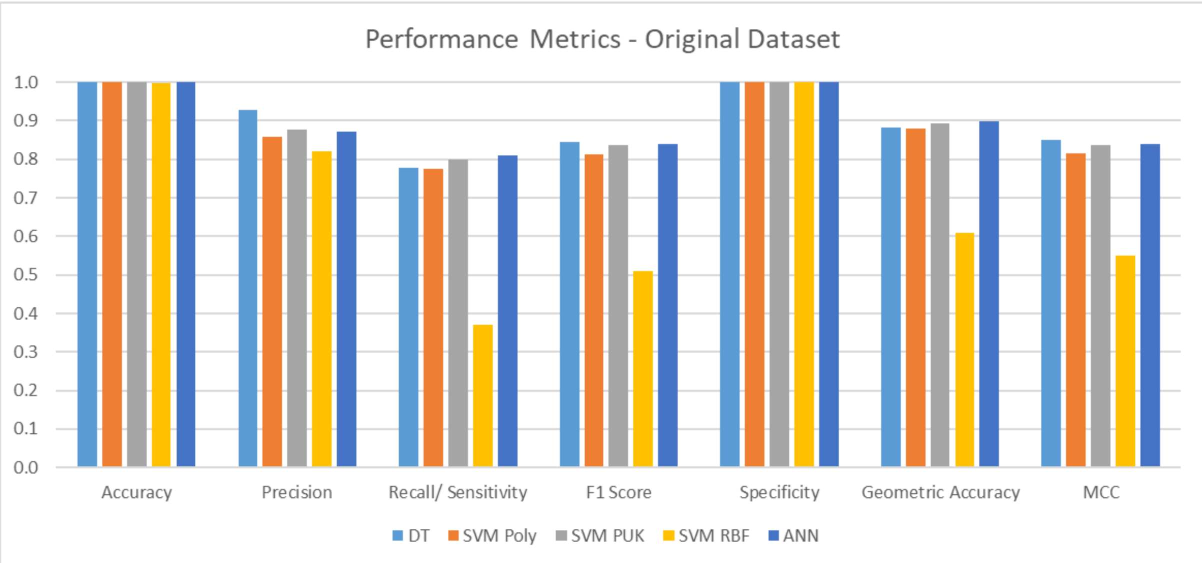


Figure 3. Performance Metrics Chart

The results were good enough to recommend proceeding to a production implementation using DT, but with such an imbalanced dataset it was decided to see if they could be improved by balancing the datasets.

Chawla et al (2002) propose Synthetic Minority Over-sampling Technique (SMOTE) to oversample with synthetic data. Their tests only oversampled up to 500% which would still leave the dataset highly imbalanced. To fully balance the dataset the minority class would need to be increased by 57,876%, which was considered to be too much; the real data would be lost amongst the vast quantity of synthetic data. To preserve as much data as possible the minority class was oversampled using the SMOTE filter at 2,000% and then the majority class was undersampled with the SpreadSubsample filter with distributionSpread at 1 to equalise the classes resulting in 10,332 instances in each. The same five algorithms were tested using the same stratified 10-fold cross-validation on the filtered dataset with the resultant confusion matrices in figure 4, metrics in figure 5, charts in figure 6 (note the different scale from the earlier chart) and WEKA screenshots in Appendix B.

C4.5 Decision Tree			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 10,158	False Positive (FP) 174	Not Fraud	Actual
False Negative (FN) 189	True Positive (TP) 10,143	Fraud	

SVM with Polynomial Kernel			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 10,248	False Positive (FP) 84	Not Fraud	Actual
False Negative (FN) 388	True Positive (TP) 9,944	Fraud	

SVM with Pearson VII Universal Kernel			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 10,307	False Positive (FP) 25	Not Fraud	Actual
False Negative (FN) 200	True Positive (TP) 10,132	Fraud	

SVM with Radial Basis Function Kernel			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 10,321	False Positive (FP) 11	Not Fraud	Actual
False Negative (FN) 784	True Positive (TP) 9,548	Fraud	

Artificial Neural Network			
Classified as			
Not Fraud	Fraud		
True Negative (TN) 10,206	False Positive (FP) 126	Not Fraud	Actual
False Negative (FN) 123	True Positive (TP) 10,209	Fraud	

Figure 4. Confusion Matrices with Balanced Dataset

Classifier	Correctly classified	Incorrectly classified	Accuracy	Precision	Recall/ Sensitivity	F1 Score	Specificity	Geometric Accuracy	MCC
DT	98.2433%	1.7567%	0.98243	0.983	0.982	0.982	0.98316	0.982	0.965
SVM Poly	97.7158%	2.2842%	0.97716	0.992	0.962	0.977	0.99187	0.977	0.955
SVM PUK	98.9111%	1.0889%	0.98911	0.998	0.981	0.989	0.99758	0.989	0.978
SVM RBF	96.1527%	3.8473%	0.96153	0.999	0.924	0.960	0.99894	0.961	0.926
ANN	98.7950%	1.2050%	0.98795	0.988	0.988	0.988	0.98780	0.988	0.976

Figure 5. Performance Metrics Results with Balanced Dataset

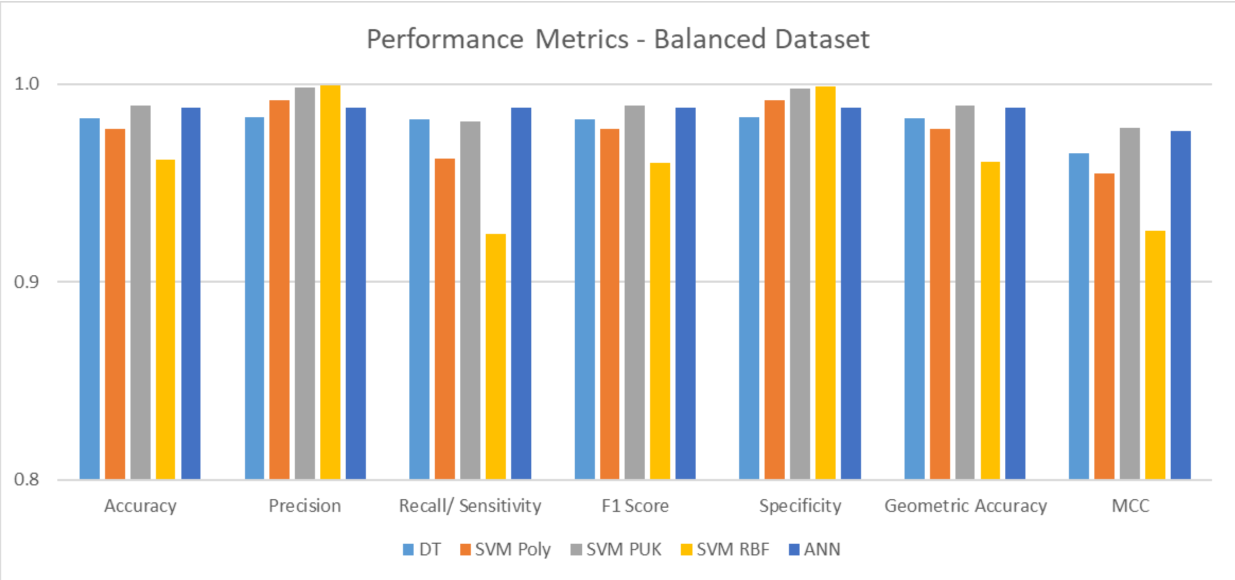


Figure 6. Performance Metrics Chart with Balanced Dataset

After balancing the dataset, accuracy and specificity deteriorated and all of the other metrics improved. Accuracy has already been shown to be unreliable on imbalanced datasets and specificity is less useful for this use case, so the **relevant** metrics all improved, leading to a conclusion that balancing the dataset was effective. SVM with PUK is now the best-performing algorithm, consistent with the expectations from Danso et al (2014) and Suhaimi & Abas (2020), although all algorithms performed well.

Selecting the best algorithm

The performance metrics on the balanced dataset show that SVM with PUK performed best, scoring highest in accuracy, precision, F1 score, specificity and MCC, and not far behind ANN on recall and GA.

Assuming YourMoney accepts that a black-box algorithm is acceptable for the credit card fraud detection use case, then SVM with PUK is recommended.

It is important to note that DT still scored well though, and was the best-performing algorithm before the dataset was balanced, so if there is any concern within YourMoney about having explainability, selecting DT instead would still provide an excellent solution with the benefit of transparent decision-making.

Applying DT to the credit card fraud detection use case

The results show that supervised learning algorithms can accurately predict fraudulent credit card transactions using the selected dataset, with improved performance after balancing the dataset. All of the algorithms performed well and SVM with PUK was selected because it performed best across the key metrics.

YourMoney should obtain a dataset of its historic credit card transactions. Domain experts should be consulted to perform feature selection (Duboue, 2020) to optimise the features on which to train the final model. An example of how this might look is provided in Appendix C using a simulated dataset from Kaggle (N.D.b).

The real dataset will be highly imbalanced like the test data, so the dataset should be balanced using a combination of SMOTE and undersampling, keeping the number of instances as high as possible. The SVM with PUK should be trained using stratified 10-fold cross-validation. Performance metrics should then be calculated to verify that the algorithm is performing as expected.

YourMoney has a cautious approach to AI, so once satisfied that the algorithm is performing well it should be deployed in passive mode, reporting fraudulent transactions to be checked manually but not intervening. Once YourMoney is satisfied that the algorithm is performing accurately and fairly it can be integrated into operations to automatically decline suspected fraudulent transactions pending confirmation by text from the card owner that the transaction is genuine.

Word count: 2,187

References

Chawla, N.V., Bowyer, K.W., Hall, L.O. & Kegelmeyer, W.P. (2002) SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, 16: 321-357.

Chicco, D. & Jurman, G. (2020) The Advantages of the Matthews Correlation Coefficient (MCC) over F1 score and Accuracy in Binary Classification Evaluation. *BMC Genomics*, 21: 1-13.

Danso, S., Atwell, E. and Johnson, O. (2014) A Comparative Study of Machine Learning Methods for Verbal Autopsy Text Classification. *International Journal of Computer Science Issues*, 10(6): 1-10. DOI: https://doi.org/10.48550/arXiv.1402.4380.

Danso, S.O., Zeng, Z., Muniz-Terrera, G. & Ritchie, C.W. (2021) *Developing an Explainable Machine Learning-Based Personalised Dementia Risk Prediction Model: a Transfer Learning Approach with Ensemble Learning Algorithms*. Frontiers in Big Data.

De Best, R. (2022) Value of credit card transactions in the United Kingdom (UK) from 2000 to 2020. Available from: https://www.statista.com/statistics/1274322/annual-credit-card-transaction-value-uk/ [Accessed 25 November 2022].

Duboue, P. (2020) *The Art of Feature Engineering: Essentials for Machine Learning*. Cambridge: Cambridge University Press.

Ghoneim, S. (2019) Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on? Available from: https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124 [Accessed 27 November 2022].

Gill, J.K. (2021) Explainable AI in Banking and Financial Services. Available from: https://www.akira.ai/blog/explainable-ai-in-financial-services/ [Accessed 26 November 2022].

Hossin, M. and Sulaiman, M.N. (2015) A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process* 5(2): 1-11.

Kaggle (N.D.a) Credit Card Fraud Detection. Available from: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud [Accessed 25 November 2022].

Kaggle (N.D.b) Credit Card Fraud. Available from: https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud [Accessed 27 November 2022].

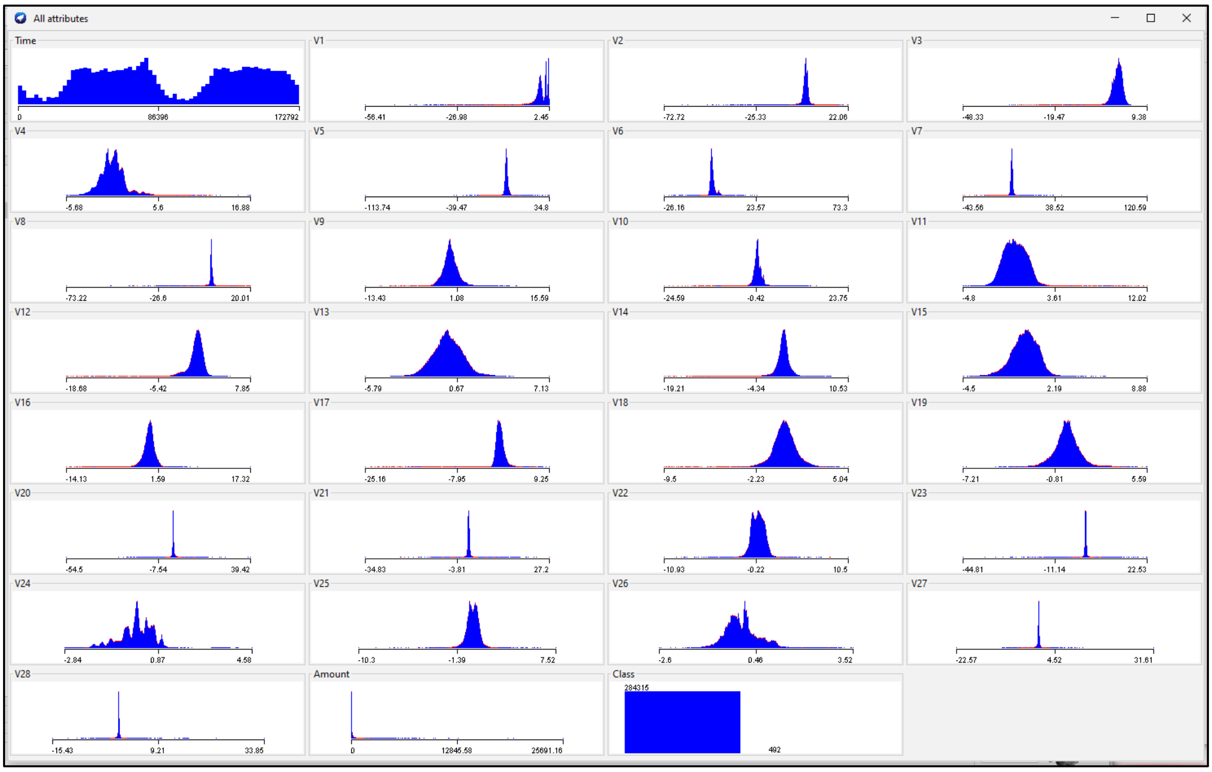
Powers, D.M. (2020) *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation*. arXiv preprint arXiv. 2010-16061.

Sokolova, M., Japkowicz, N. & Szpakowicz, S. (2006) ‘Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation’, *Australasian Joint Conference on Artificial Intelligence*. Hobart, Australia, 4-8 December. Berlin, Heidelberg: Springer. 1015-1021.

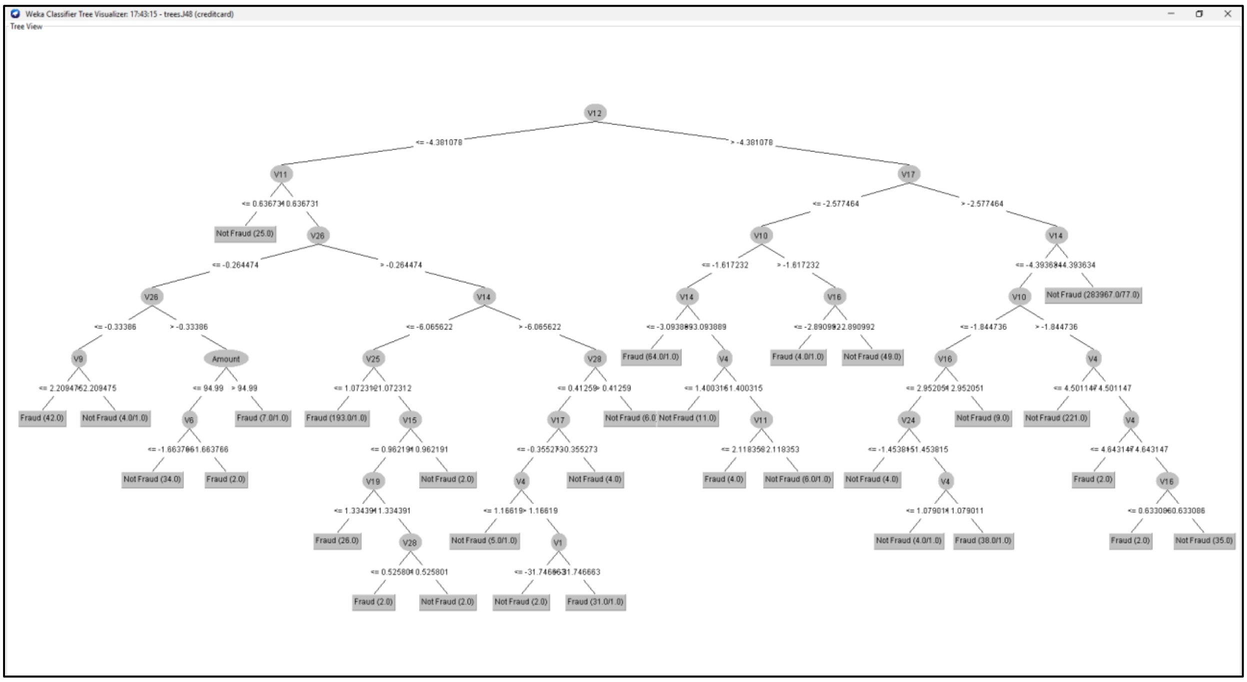
Suhaimi, N.A.D. & Abas, H. (2020) A Systematic Literature Review on Supervised Machine Learning Algorithms. *Perintis E-Journal* 10(1): 1-24.

UK Finance (2021) *Fraud – The Facts 2021: The Definition Overview of Payment Industry Fraud*. Available from: https://www.ukfinance.org.uk/system/files/Fraud%20The%20Facts%202021-%20FINAL.pdf [Accessed 25 November 2022].

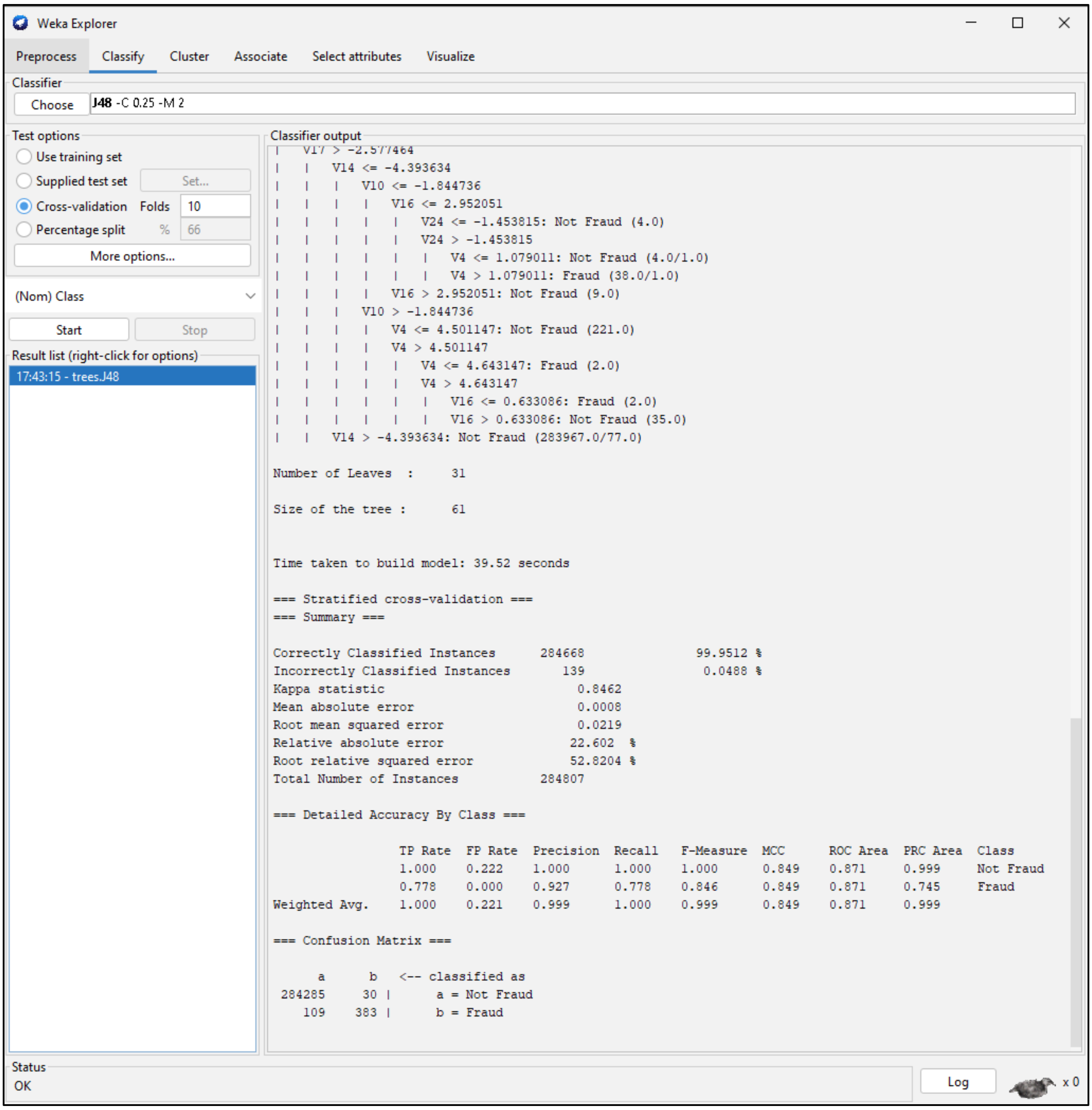
Appendix A – WEKA Screen Shots using Primary Dataset



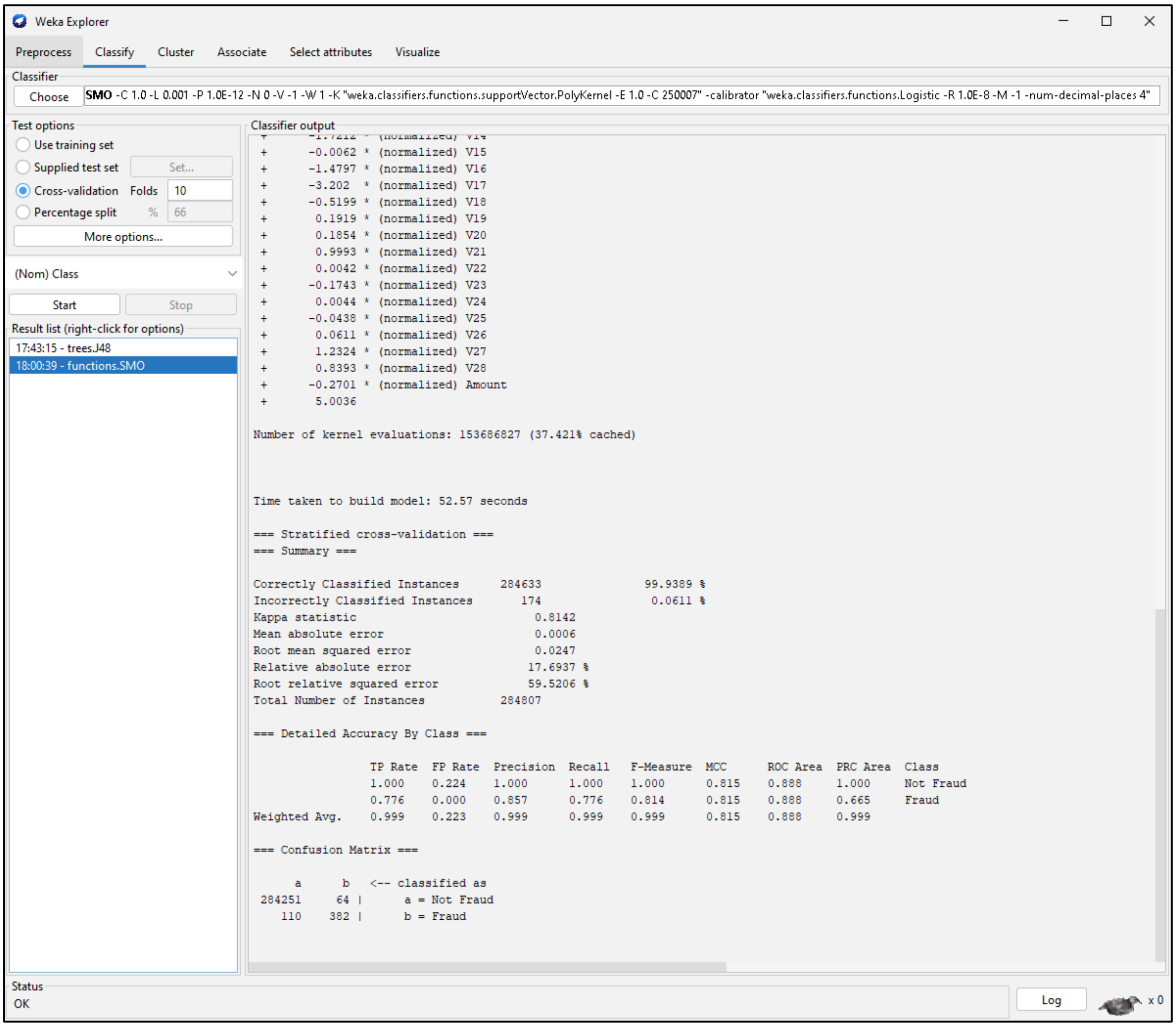
All Data Visualised



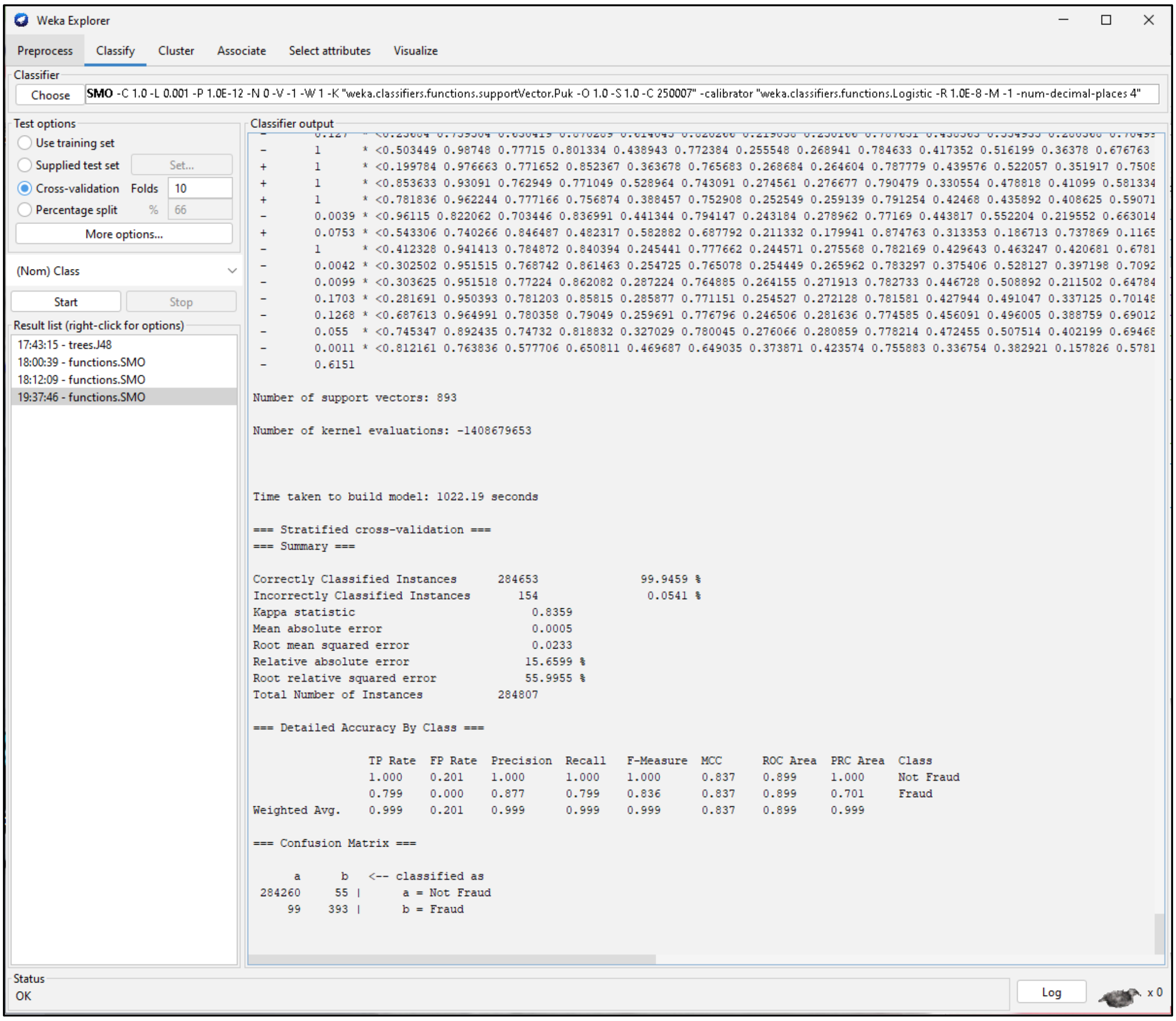
Decision Tree Visualisation



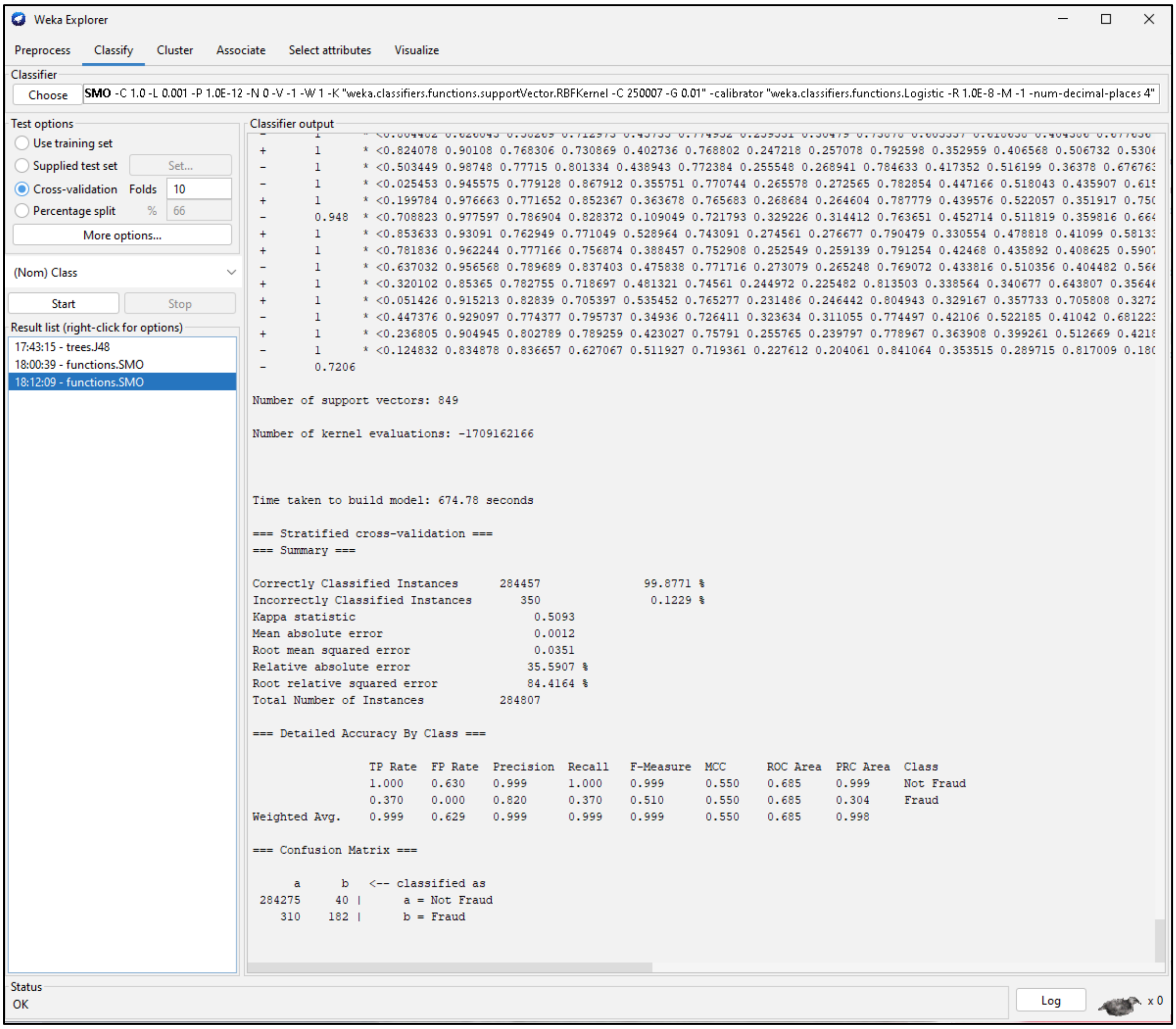
Decision Tree Results



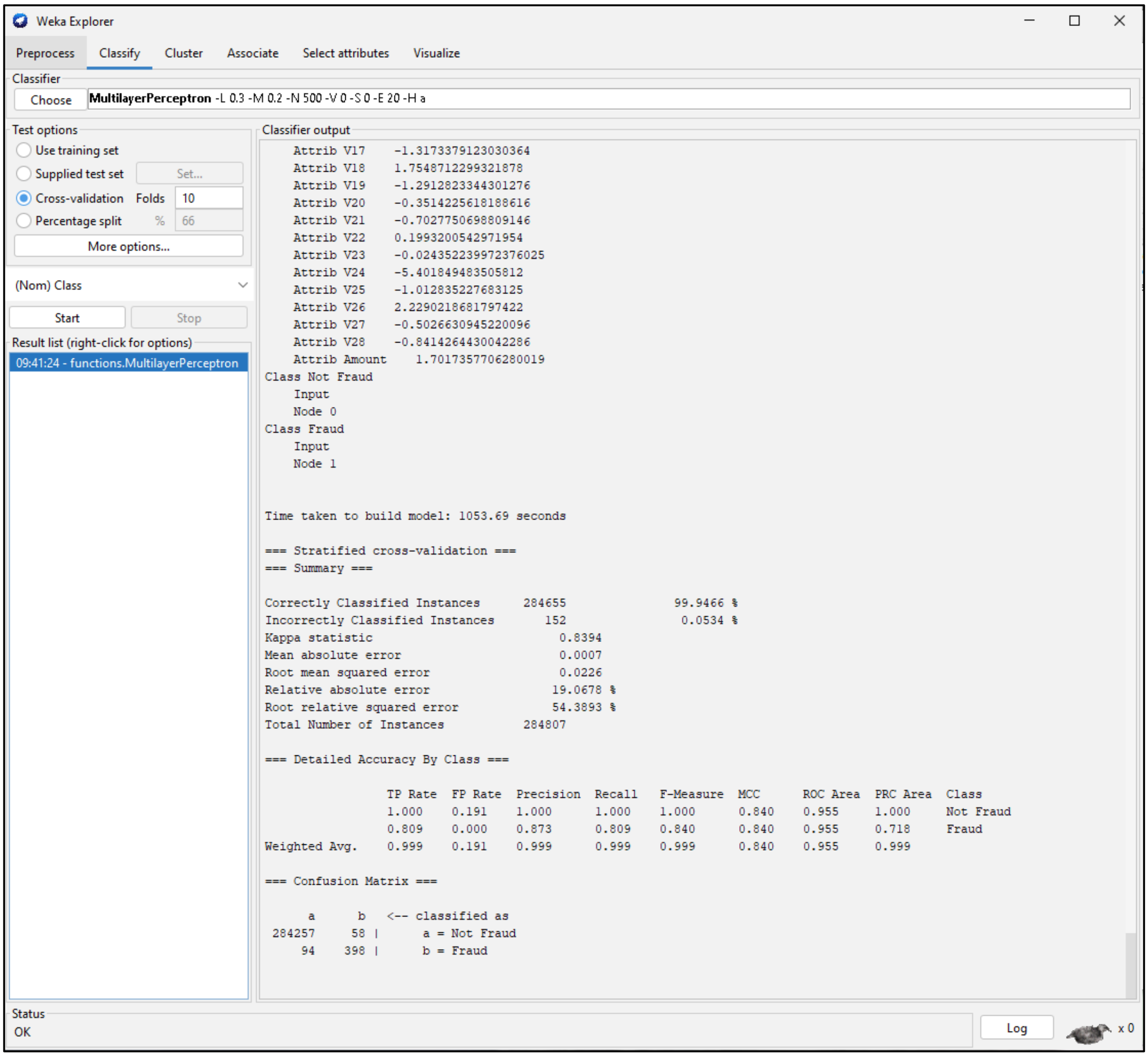
Support Vector Machine with Polynomial Kernel Results



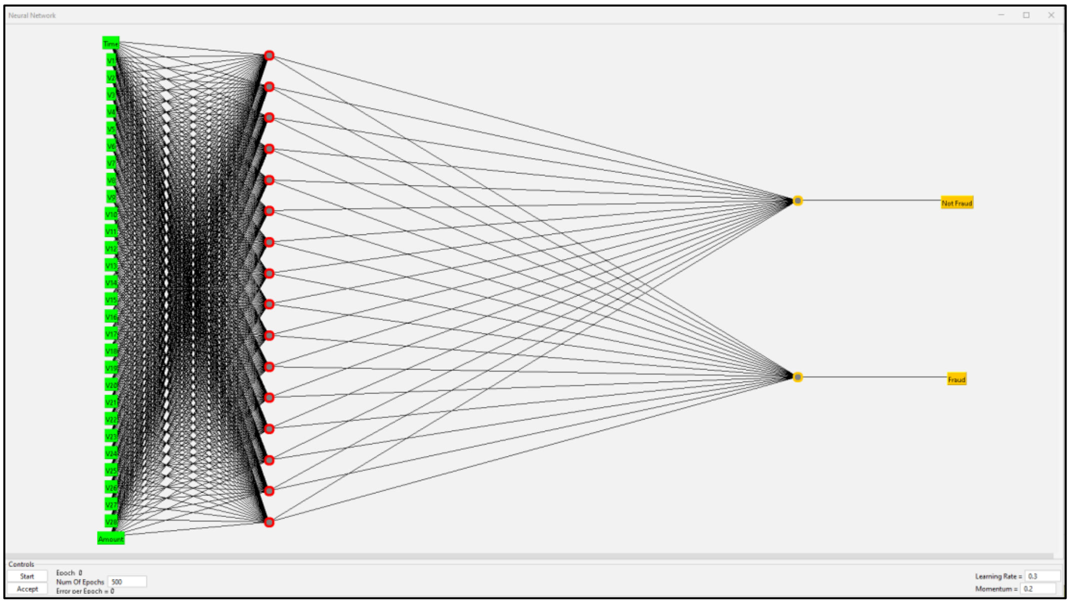
Support Vector Machine with Pearson VII Universal Kernel Results



Support Vector Machine with Radial Basis Function Kernel Results

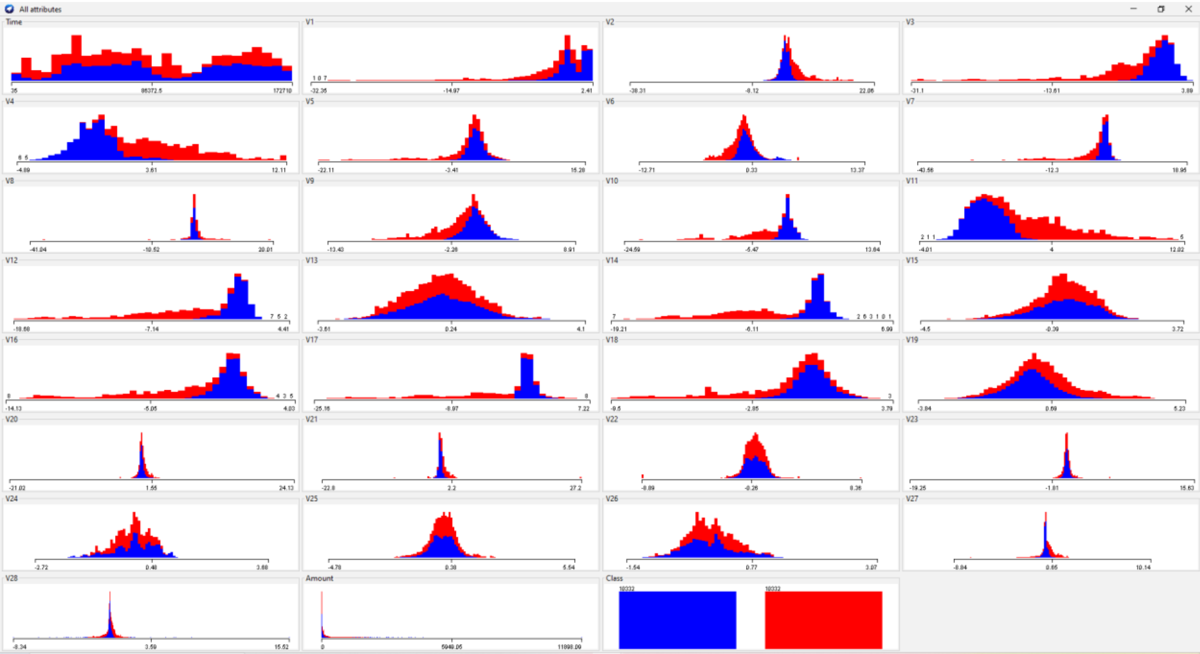


Artificial Neural Network Results

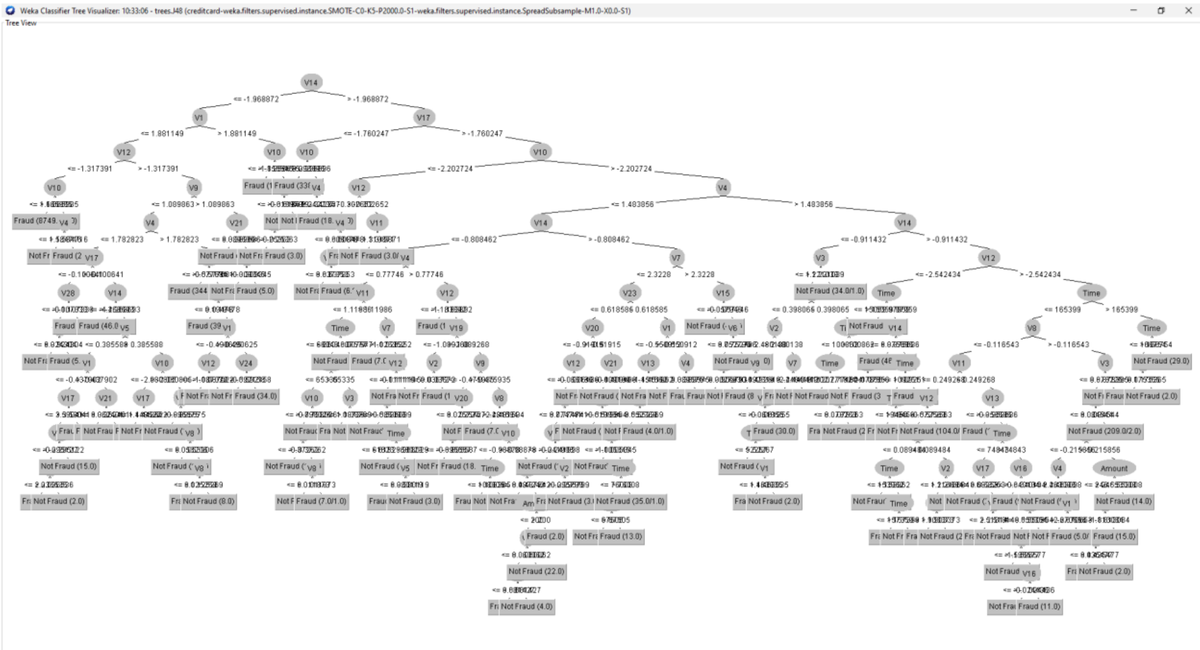


Artificial Neural Network Visualisation

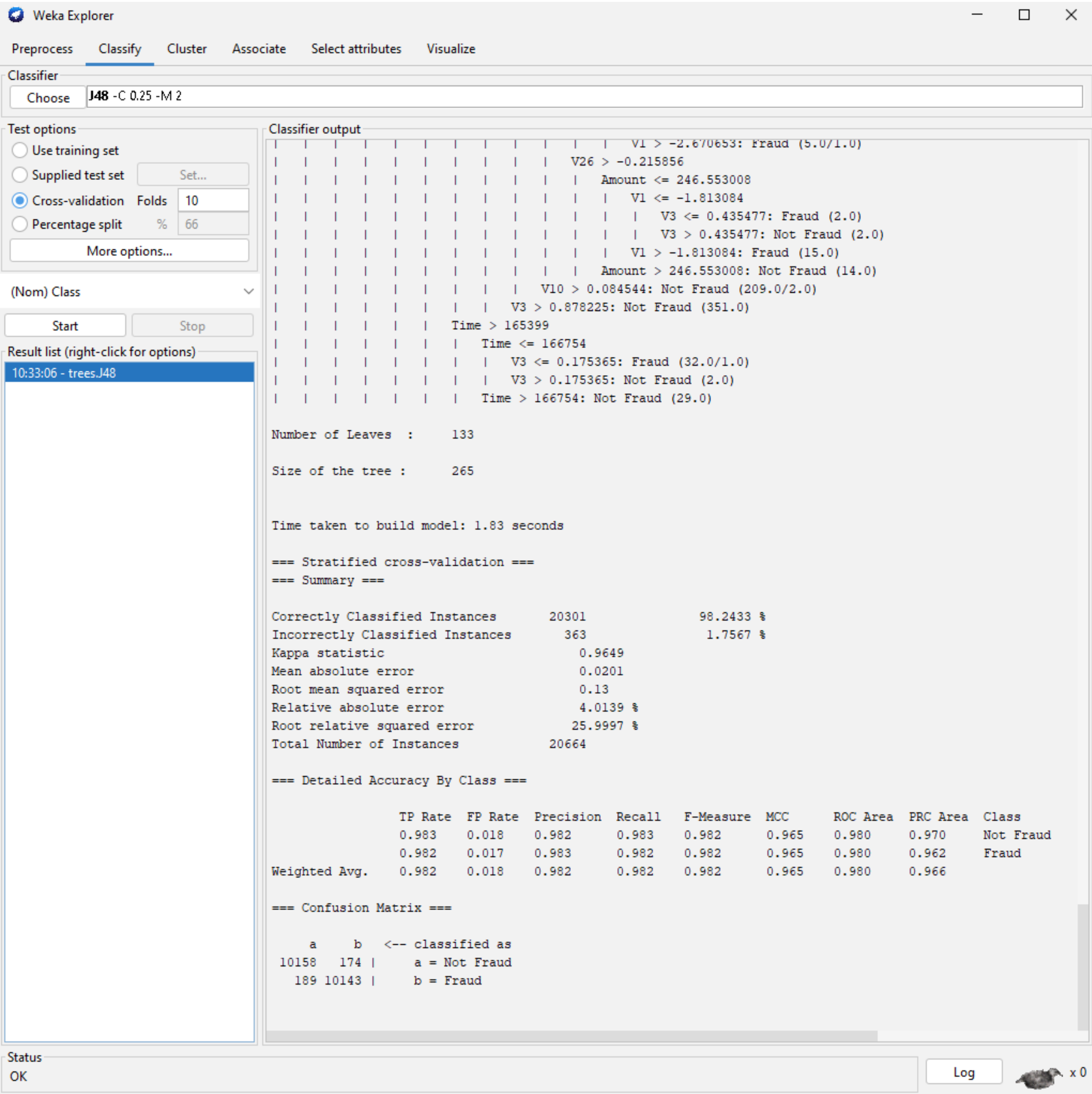
Appendix B – WEKA Screen Shots using Balanced Dataset



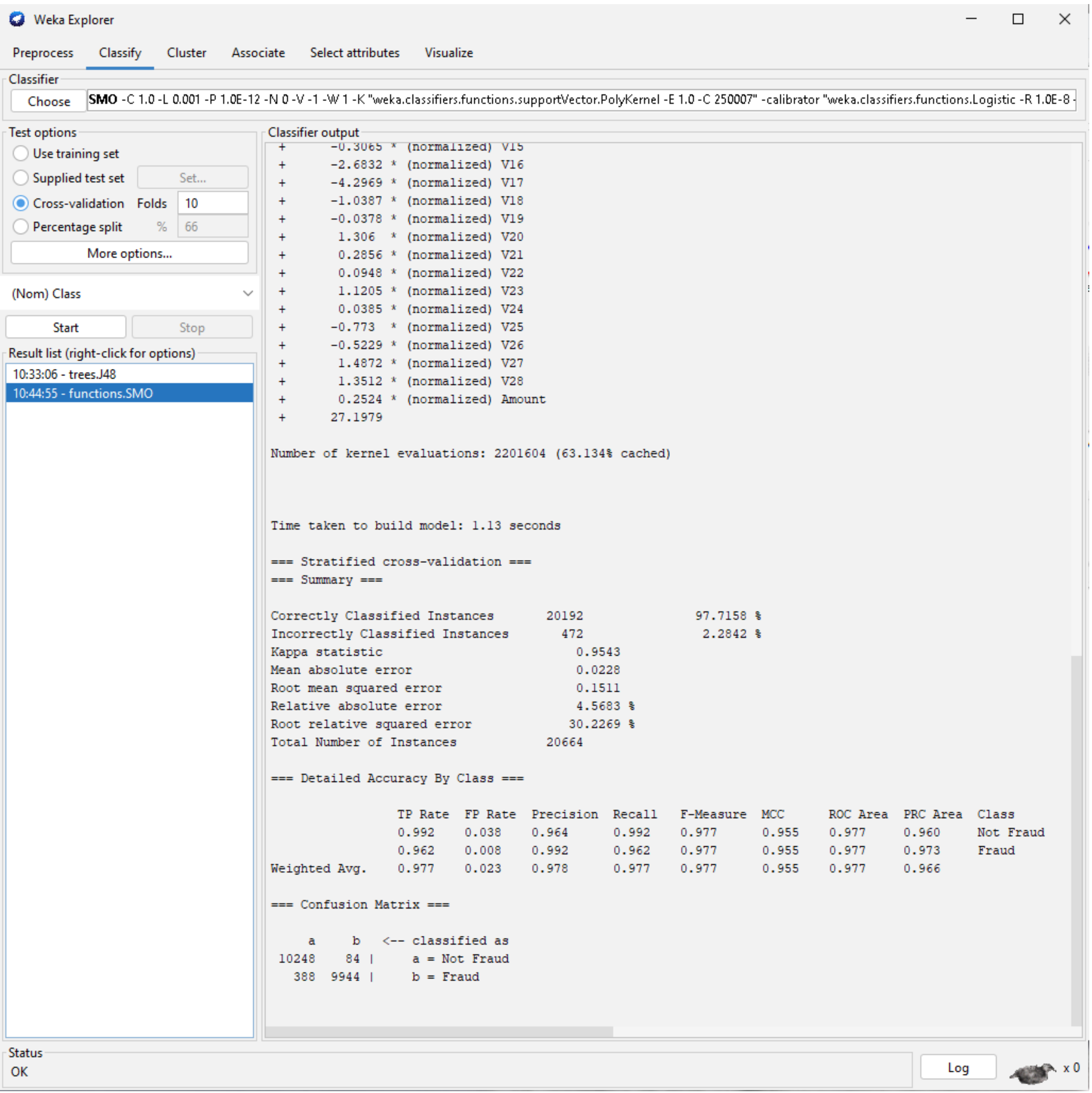
All Data Visualised



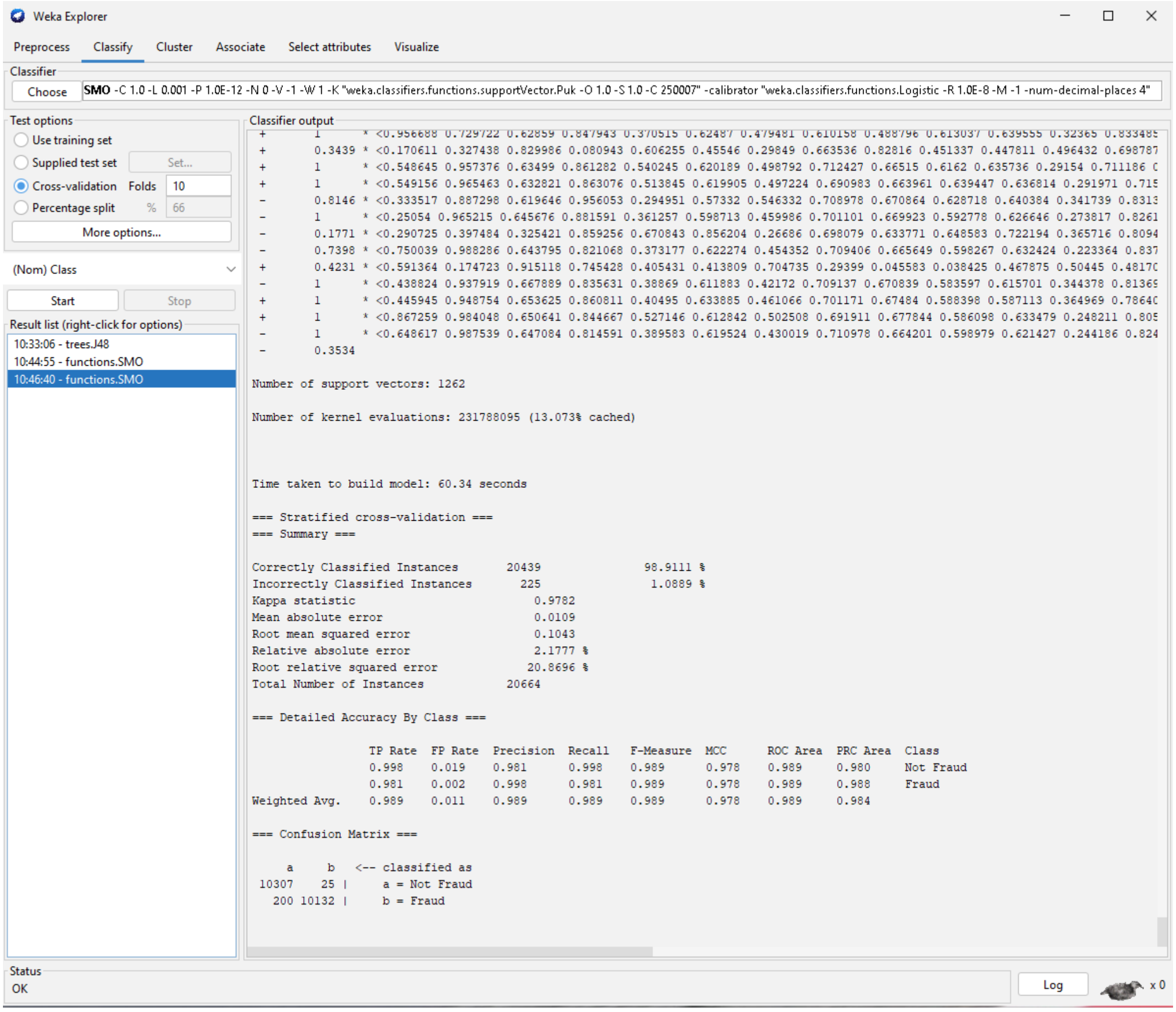
Decision Tree Visualisation



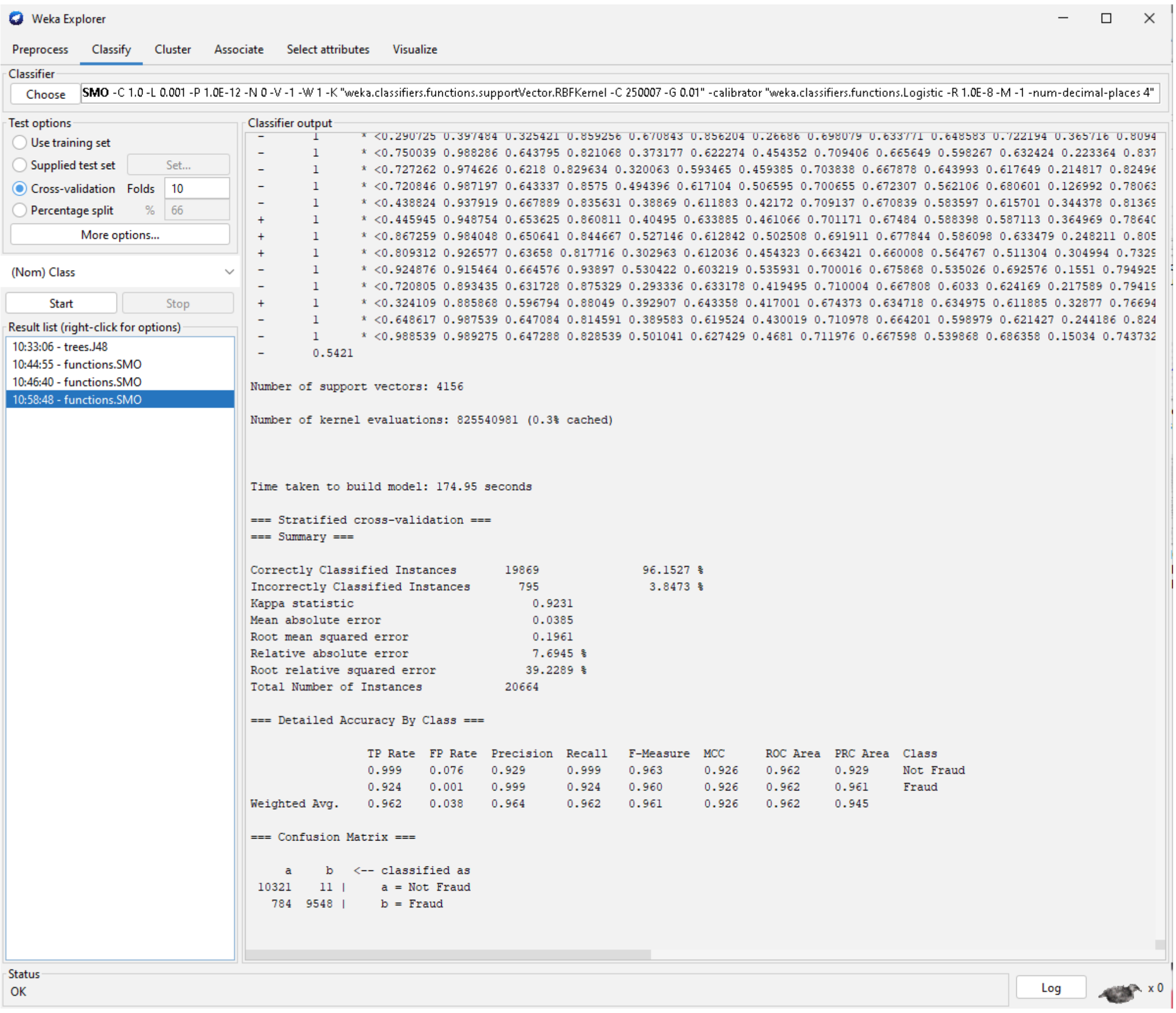
Decision Tree Results



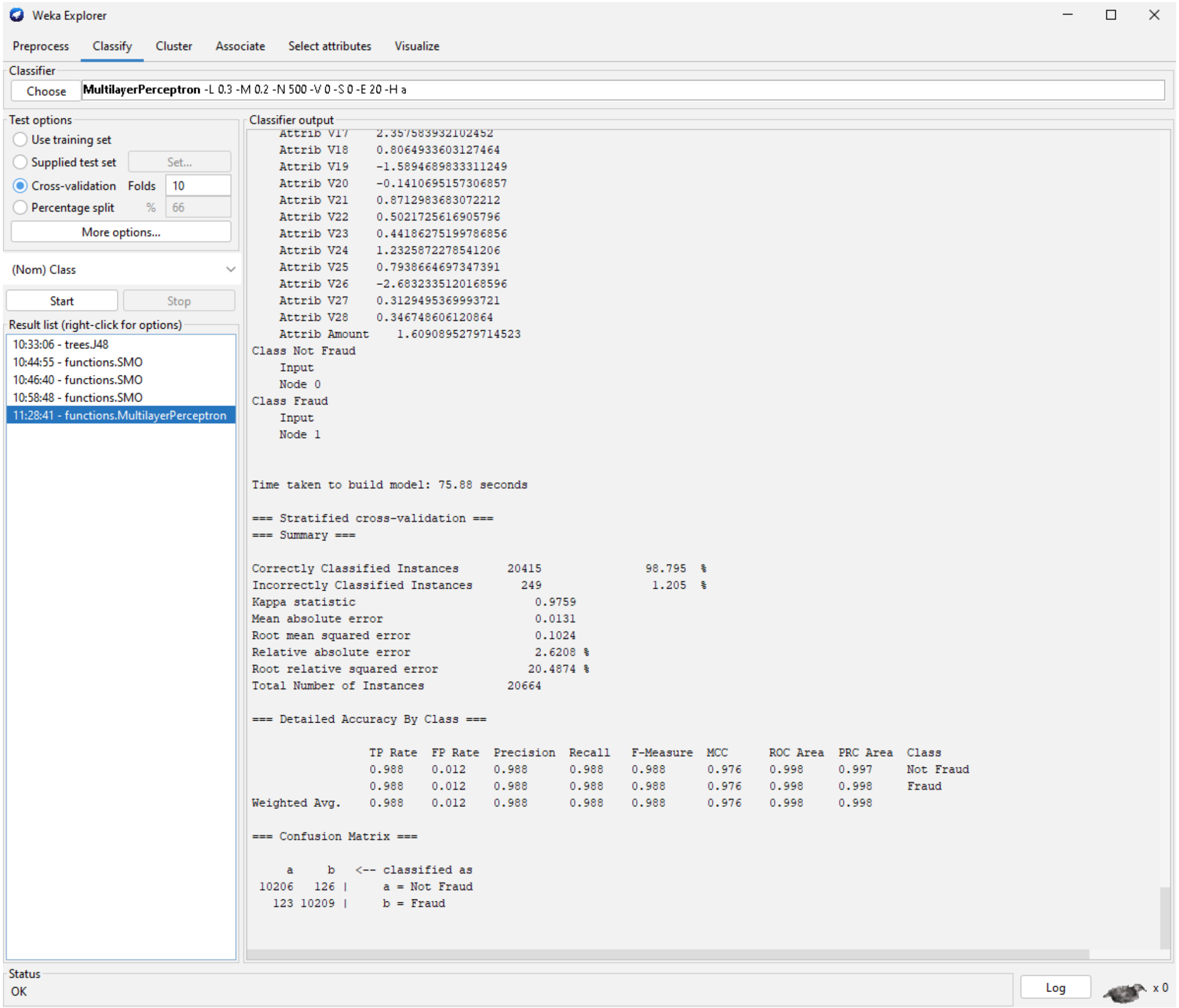
Support Vector Machine with Polynomial Kernel Results



Support Vector Machine with Pearson VII Universal Kernel Results

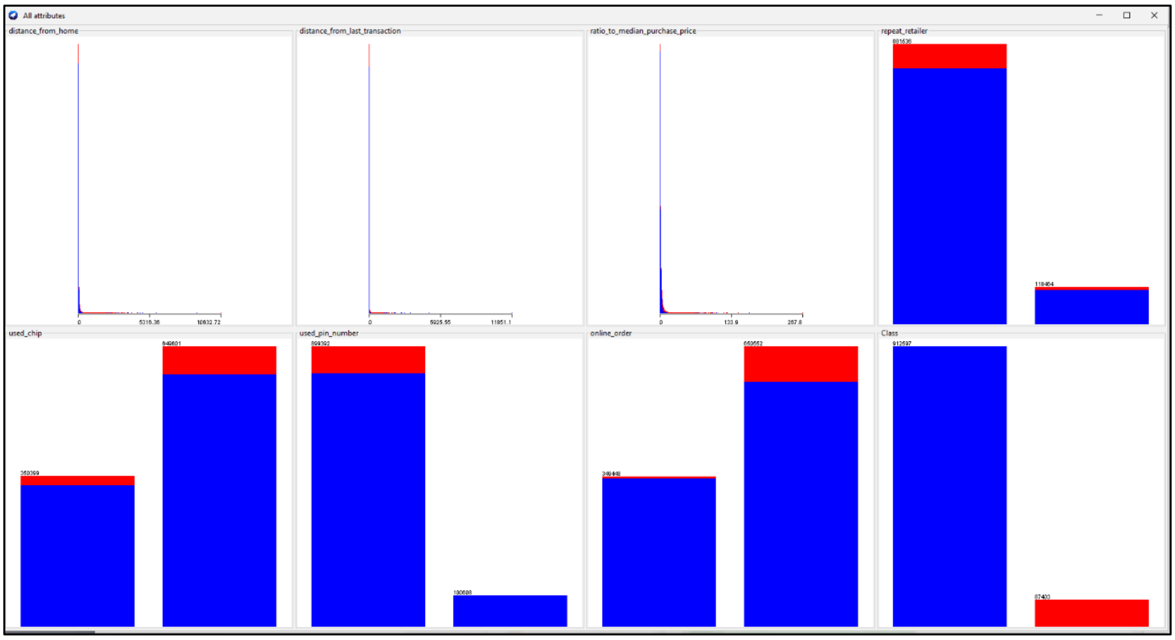


Support Vector Machine with Radial Basis Function Kernel Results

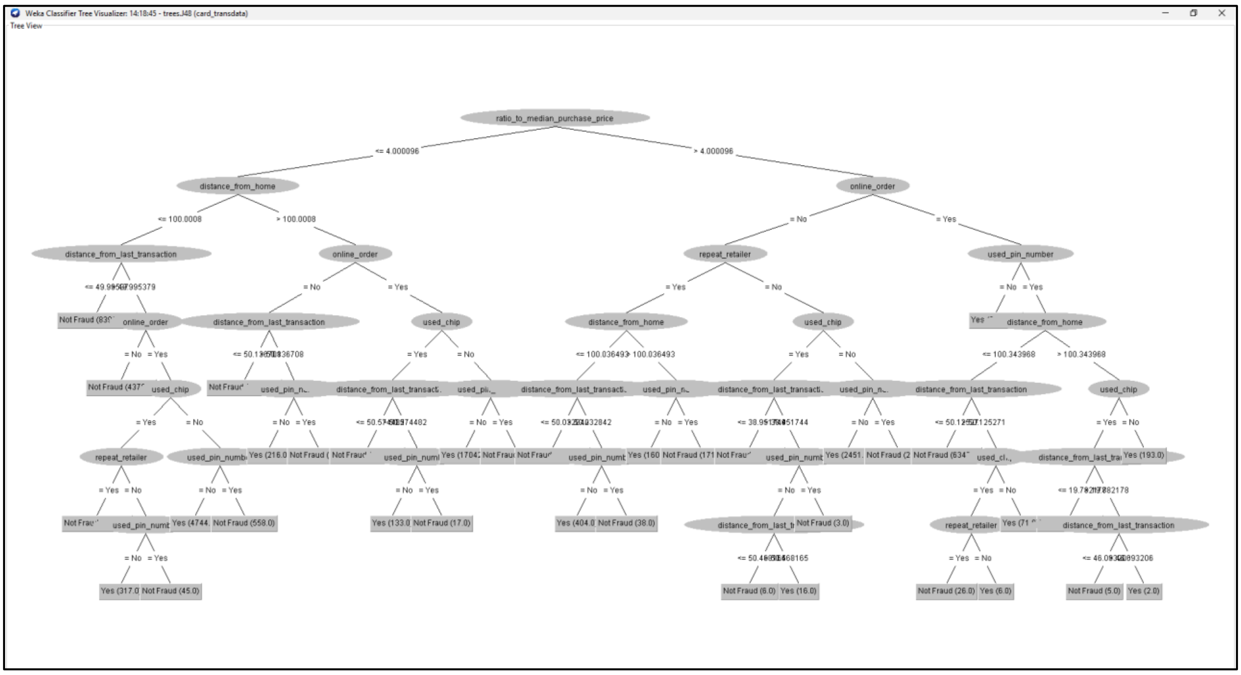


Artificial Neural Network Results

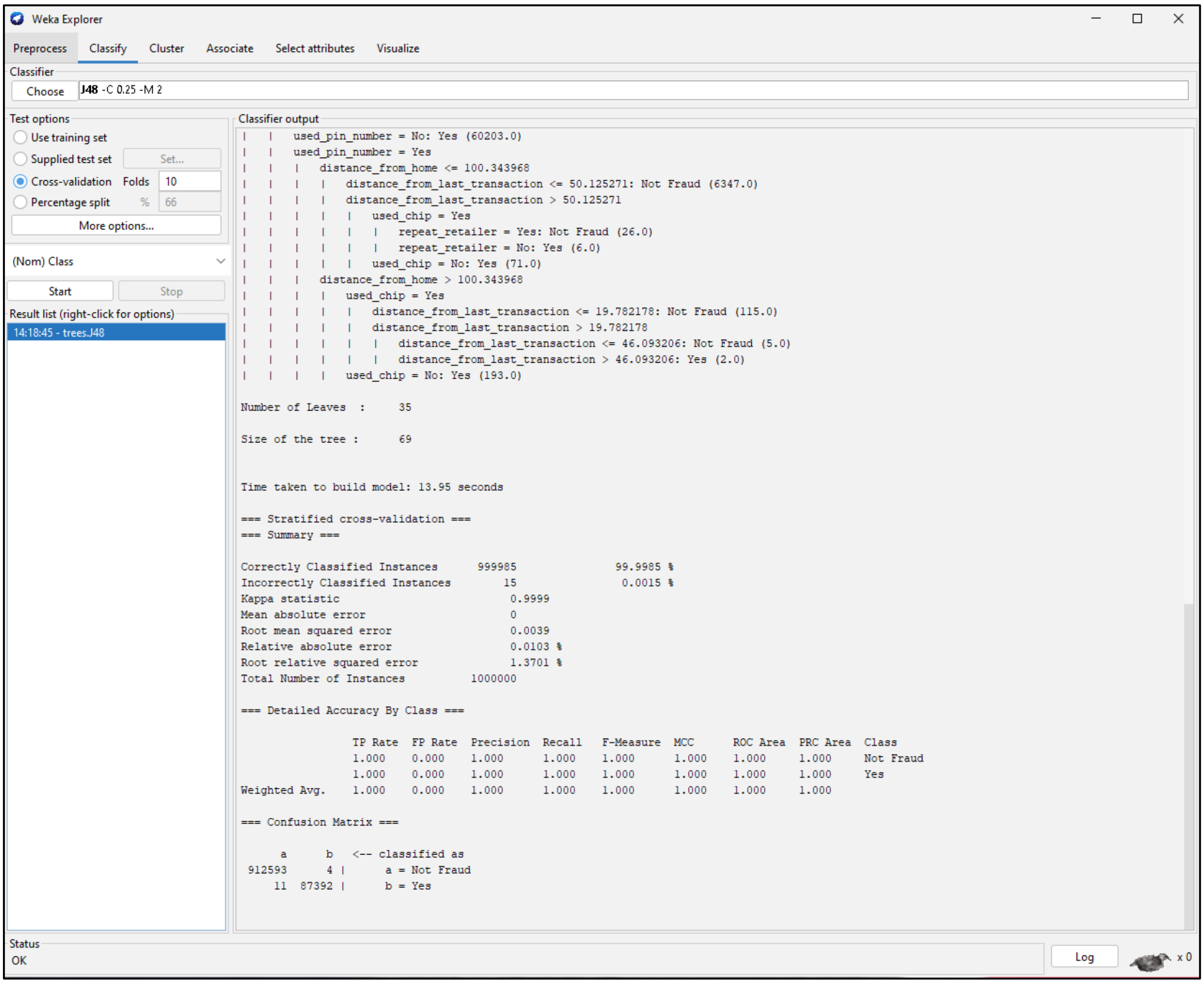
Appendix C – WEKA Screen Shots using Simulated Dataset, DT Only



All Data Visualised



Decision Tree Visualisation



Decision Tree Results