Table of Contents

List of acronyms

ML	Machine Learning
AI	Artificial Intelligence
IoT	Internet of things
SMOTE	Synthetic minority oversampling technique
WEKA	Waikato environment for knowledge analysis
SVM	Support vector machines
AUC	Area under curve
ROC	Receiver Operating Characteristic

CHAPTER 1 – Artificial intelligence in financial sales and marketing

Forecasting of sales and marketing initiatives to broader customer bases has become an increasingly important function in the financial sector. Use of AI to optimise marketing can increase customer interaction across marketing channels, improves market forecasting, and provides automation in defining target customers in the interest of increased likelihood of sales (Vla˘ci´c, et al., 2021).

CHAPTER 2 Dataset used for marketing analysis

The dataset chosen for this analysis is from the UCI machine learning repository. It is related to direct marketing campaigns of a Portuguese banking institution. Albeit being a large dataset, it closely matches the problem First Finance Company would wish to solve and is an empirical dataset.

Link:

https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

CHAPTER 3 – Properties of dataset

The marketing dataset contains 45211 instances of data with the following attributes:

Type	Description	Variable type	Description of attribute
attribute	age	numeric	Customer age
attribute	job	categorical	type of job ('admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
attribute	marital	categorical	('divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
attribute	education	categorical	('basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
attribute	default	categorical	has credit in default? ('no','yes','unknown)
attribute	balance	numeric	average yearly balance in euros
attribute	housing	categorical	has housing loan? ('no','yes','unknown')
attribute	loan	categorical	has personal loan? ('no','yes','unknown')
attribute	contact	categorical	contact communication type ('cellular','telephone')
attribute	day	numeric	last contact day of the month
attribute	month	categorical	last contact month of year ('jan', 'feb', 'mar', ..., 'nov', 'dec')
attribute	duration	numeric	last contact duration, in seconds
attribute	campaign	numeric	number of contacts performed during this campaign and for this client
attribute	pdays	numeric	number of days that passed after the client was last contacted from a previous campaign (999 means client was not previously contacted)
attribute	previous	numeric	number of contacts performed before this campaign and for this client
attribute	poutcome	categorical	outcome of the previous marketing campaign ('failure','nonexistent','success')
attribute	y	binary	has the client subscribed a term deposit? ('yes','no')

Table 1 – Dataset definition

The duration attribute will be removed from the model at the pre-processing stage as it is closely related with the output target (if duration = 0, y = “no”).

The data as imported into WEKA is shown in Appendix A, Figure 4. No missing values are seen in any attribute. After running a RemoveDuplicates filter to ensure no duplicate entries exist in the data, 45195 unique instances remain as shown in Appendix A, Figure 5.

(Zolanvari, et al., 2018) show the effect of class imbalance, stating that class imbalance may lead to an unrepresentative model. In the case of this dataset, it is seen that the class in severely imbalanced, (88.29%/11.7% [no/yes]) shown in Appendix A, Figure 6. Methods such as under-sampling (taking fewer instances in the majority class), or over-sampling (taking more samples from the minority class) exist, though this may lead to overfitting in the case of over-sampling, or loss of useful information in the case of under-sampling (Zolanvari, et al., 2018). The method chosen from (Danso, 2022) is the use of synthetic minority oversampling technique (SMOTE) to generate synthetic instances of the unbalanced class.

SMOTE generated 34 537 more instances in the dataset to bring the total instances for classification to 79732 instances, which are randomised for training using the Randomize function. The final class balance after SMOTE is 50%/49.94% [no/yes] as shown in Appendix A, Figure 7.

CHAPTER 4 – Approach for prediction model

The testing for algorithms used will be done in the Waikato environment for knowledge analysis (WEKA). WEKA provides a simple tool with a powerful set of pre-processing filters, visualizations, and machine learning algorithms.

(van Engelen & Hoos, 2020) define labelled data as a set of data points consisting of input x and a corresponding output value y. The dataset in use matches this description, having inputs for multiple attributes of the customer, mapped to an output or class (Was term loan accepted Y/N). We will thus use supervised learning methods to train the chosen algorithm on the dataset. (Yang & Shami, 2020) also note that hyper-parameter configuration has a direct impact on the model's performance. In the tests done in Chapter 5 below, some of these parameters will be assessed and various parameters used to find the highest F-measure.

CHAPTER 5 – Prediction models used

(Mahesh, 2018) notes that this type of dataset (labelled) is suited for supervised learning methods such as decision tress, support vector machines (SVM) with (Brownlee, 2021) also stating that backpropagation algorithms using multi-layer feedforward neural networks also work well with supervised learning methods. These algorithms will be interrogated.

Decision tree

Decision trees consist of a decision structure beginning with a root decision with branches and leaves (Mahesh, 2018). Each branch represents a value that the node can take (Mahesh, 2018). Some of the benefits are computational efficiency, and efficacy (Mitchell, et al., 2018), even with dealing with larger datasets.

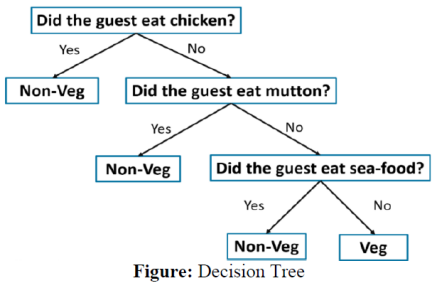

Figure: Decision Tree

Figure 1 – Simple decision tree (Mahesh, 2018)

Support Vector Machines

Support vector machines (SVM) are another popular supervised machine learning method used for classification and regression analysis. SVM's work by drawing margins between classes, and then maximising them (from the mean hyperplane), thereby minimising classification errors (Mahesh, 2018).

SVM's are also efficient, even in nonlinear classification and use what is called the kernel trick (calculating similarity measures in a high dimensional feature space) to map inputs into a high dimensional feature space (Mahesh, 2018).

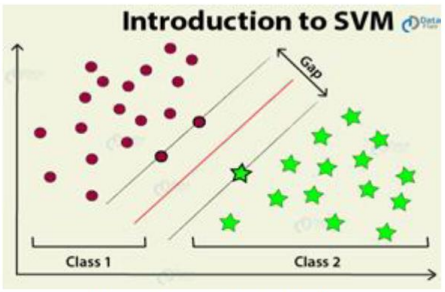

Figure: Support Vector Machine

Figure 2 – SVM classification (Mahesh, 2018)

Multilayer perceptron

In a supervised neural network, the output for the input nodes is already known as the data is labelled. The algorithm's predicted output is compared with the actual output, and the node weights altered accordingly (Mahesh, 2018). Two of the problems of neural networks is the large storage consumption and computational complexity of the network (Cheng, et al., 2018).

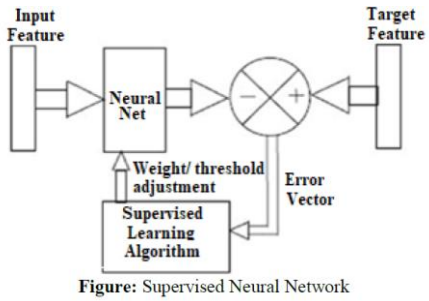

Figure: Supervised Neural Network

Figure 3 – Supervised neural network classifier (Mahesh, 2018)

CHAPTER 6 – Analysis of algorithm outputs

Decision tree

The first algorithm tested was a decision tree, J48 pruned decision tree. The decision tree algorithm was the most computationally efficient, building the model in approximately 4.85 seconds. A few hyperparameter tuning parameters were tested to find the best overall result. Default values of 3 is used for Numfolds (One third of the data used for pruning, two thirds used for growing the tree), seed value left at default 1 (used for reduced error pruning), and unpruned setting left at false for this trial. The Confidence factor, which represents confidence level of the data integrity was used at two values, 0.25 and 0.5. The higher the confidence factor, the less pruning will occur as it denotes higher confidence in the dataset (Mantovani, et al., 2018). The data was tested with 10 validation cross folds, and a 66% training 34% testing split. Parameters discussed to assess performance of the model are F-Measure and Area under curve (AUC) for Receiver Operating Characteristic (ROC). The F Measure is the harmonic mean of precision and recall, with each weighted the same. This is helpful in describing the performance and comparing models (Brownlee, 2020). A ROC curve shows the performance of a model (distribution of classifications) at all thresholds and is a graph of the true positive rate against the false positive rate (Martens, 2021).

Using the default parameters (C=0.25, Seeds = 1; numfolds = 3) gave the best balance of computational cost and highest F-Measure score (0.9/0.87 [no/yes]) with 89.87% correctly classified instances and 10.12% incorrectly classified instances, as shown in Appendix A Figure 8. Area under ROC curve is also high 92.6% as shown in Appendix B Figure 9.

An increased confidence factor of 0.5% (C=0.5) did not yield any discernible improvements to the classifier with an F-Measure score (0.899/0.896 [no/yes]) with 89.77% correctly classified instances and 10.22% incorrectly classified instances as shown in Appendix B Figure 10, and a lower ROC area of 92.1% as shown in Appendix B Figure 11.

Using a training split of 66% training and 34% testing with the default hyperparameter settings (C=0.25, Seeds = 1; numfolds = 3) produced an F-measure score of 0.893/0.888 [no/yes] with 89.08% correctly classified instances and 10.91% incorrectly classified instances as shown in Appendix B Figure 12 and corresponding ROC curve in Appendix B Figure 13.

Support Vector Machine

The LibSVM package was installed for the purposes of the testing in WEKA. The main hyperparameters that have been tuned are C and Gamma. The C parameter adds a penalty for each misclassified data point. If C is low, the penalty for misclassification is low, and the decision boundary with a large margin is chosen, thus a greater number of misclassifications is allowed (Yildirim, 2020). Gamma is used to influence the margins chosen to perform the kernel trick. Higher values of gamma result in tighter boundaries between margins meaning the fit of the margins in the high feature space is tighter between clusters of data (Yildirim, 2020). The SVM algorithm requires much higher computational cost than decision trees with the full dataset, with each iteration building the model in approximately 2316.12 seconds.

For test 1 default values of C and gamma were used (C=1.0, Gamma = 0). This resulted in a F-Measure score of 0.85/0.837 [no/yes], with 84.85% correctly classified

instances and 15.14% incorrectly classified instances. 10-Fold cross validation is used on the entire dataset for the test. Results are shown in Appendix B Figure 14 with a AUC of 0.831 (Appendix B Figure 15).

For test 2, the C parameter was increased to add increased penalty for misclassification (C=3.0, Gamma=0). This resulted in a F-Measure score of 0.85/0.847 [no/yes], with 84.85% correctly classified instances and 15.14% incorrectly classified instances (Appendix B Figure 16). The AUC is 0.848 (Appendix B Figure 17). This shows a marginal improvement over default values with a higher C parameter.

For Test 3, the C parameter was increased to 5.0 (C=5.0, Gamma=0). The results remained negligible, F-Measure score of 0.853/0.849 [no/yes], with 85.08% correctly classified instances and 14.91% incorrectly classified instances as seen in Appendix B Figure 18 and AUC Appendix B Figure 19. In testing, higher values of Gamma (Gamma=0.3~0.6) reduced the F-Measure to 0.772/0.62 [no/yes] with lower correctly classified instances at 71.52%. Increasing gamma (tightening hyperplane/s) between 0.3~0.6 was found to make this models F-measure decrease, thus it will be left at 0.

Multi-layer Perceptron

For this algorithm to execute with an acceptable computational cost, the dataset was condensed using the RemoveStratifiedFolds filter. Since our data is well randomised, the dataset was split in 10 folds. Of these, fold 3 was used for training, and fold 7 for testing. Both folds contained the same class balance of the original set, and was condensed to 7973 instances as shown in Appendix C Figure 20.

The major hyperparameters in the multi-layered perceptron are the learning rate, momentum, and number of hidden layers (Weissbart, 2020). The learning rate (LR) determines the step size at each iteration (Epoch) while moving to a minimum loss function (Wikipedia, 2022). Momentum (M) increases the size of steps taken toward the minimum loss function.

For test 1, default values were executed for baselines (LR=0.3, M=0.2, Epoch=500, hidden layers=a). This resulted in 1 hidden layer with 25 nodes. An F-Measure score of 0.767/0.787 [no/yes], with 77.75% correctly classified instances and 22.25% incorrectly classified instances. This is shown in Appendix C Figure 21. The AUC is 0.854 with these settings (Appendix C Figure 22)

Many further tests were performed using differing values on LR, M and number of nodes. The best results gathered were L=0.5, M=0.2, Number of hidden nodes=5. These hyperparameters resulted in an F-Measure of 0.8/0.799 [no/yes] with 79.91 correctly classified instances, and 20.08% incorrectly classified instances as in Appendix C Figure 23. The AUC is 0.872 with these parameters as shown in Appendix C Figure 24.

Results between algorithms

Algorithm	F-Measure (n)	F-Measure (y)	F-Measure (Ave)	Correctly Classified	Incorrectly Classified	AUC (ROC)
Support Vector machine	0.853	0.849	0.851	85.08%	14.91%	0.851
Multi-layered Perceptron	0.8	0.799	0.799	79.91%	20.08%	0.872
Decision tree	0.901	0.896	0.899	89.87%	10.12%	0.926

Table 2 – Highest score achieved for 3 tested algorithms

Support vector machine algorithm is computationally costly on this dataset. It is seen that changes to the C parameter and Gamma did not yield large increases in classification accuracy on this dataset. While the best F-Measure and accuracy is seen with increased C parameter with no change to Gamma, the results of the tuned hyperparameters is very similar to the default settings. The F-Measure and AUC in Table 1 is acceptable in isolation.

The multi-layered perceptron was equally costly on computation, requiring condensing of the dataset. The best hyperparameters found above yielded an F-Measure of 0.8/0.799 [no/yes] which is less accurate than the SVM. This is surprising as this network should be capable of better accuracies, perhaps with more testing time and the addition of multiple learning layers it can be improved.

The decision tree algorithm did a startling job as listed in Table 1, with a very high 0.926 AUC and lowest computation time by far. It performed the best with default settings, and for this dataset, is the best suited for prediction with this dataset. Decision trees algorithm has shown a prediction accuracy of 89.87%.

CHAPTER 7 – Benefit to First Finance

The results of the testing show that highest classifications possible are an accuracy of approximately 89.87% in this testcase. This accuracy will allow First Finance Company to optimise sales and marketing initiatives to include the respondents with a much higher likelihood of taking term loans. Further to this, First Finance Company will benefit from savings in time and thus cost, and can easily be enable differing marketing

content to be automated and sent to the correct customers enabling agents to further enhance offerings.

CHAPTER 8 - References

Brownlee, J., 2020. *A Gentle Introduction to the Fbeta-Measure for Machine Learning.* [Online]

Available at: https://machinelearningmastery.com/fbeta-measure-for-machine-learning/#:~:text=The%20F%2Dmeasure%20is%20calculated,model%20and%20in%20comparing%20models.

[Accessed 27 05 2022].

Brownlee, J., 2021. *How to Code a Neural Network with Backpropagation In Python (from scratch).* [Online]

Available at: https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/

[Accessed 23 05 2022].

Cheng, J. et al., 2018. Recent Advances in Efficient Computation of Deep Convolutional Neural Networks. *Frontiers of Information Technology & Electronic Engineering,* 19(1), pp. 64-77.

Danso, S., 2022. *UAI_PCOM7E March 2022: Unit 11 seminar recording now available.* England: s.n.

Mahesh, B., 2018. Machine Learning Algorithms -A Review. *International Journal of Science and Research (IJSR),* 9(1), pp. 381-386.

Mantovani, R. G. et al., 2018. An empirical study on hyperparameter tuning of decision trees. *arXiv preprint arXiv:1812.02207,* pp. 4-36.

Martens, F. K., 2021. *POLYGENIC RISK PREDICTION OF COMMON DISEASES.* 1 ed. Enschede: Ipskamp Printing.

Mitchell, R., Adinets, A., Rao, T. & Frank, E., 2018. XGBoost: Scalable GPU Accelerated Learning. *arXiv preprint arXiv:1806.11248,* pp. 1-5.

van Engelen, J. E. & Hoos, H. H., 2020. A survey on semi-supervised learning. *Machine Learning,* 109(1), pp. 373-440.

Vlaˇci´c, B., Corbo, L., Silva, S. C. e. & Dabi´c, M., 2021. The evolving role of artificial intelligence in marketing: A review and research agenda. *Journal of Business Research,* 128(1), pp. 187-203.

Weissbart, L., 2020. *Performance Analysis of Multilayer Perceptron in Profiling side-channel analysis.* Cham, Springer.

Wikipedia, 2022. *Learning rate.* [Online] Available at: https://en.wikipedia.org/wiki/Learning_rate [Accessed 25 05 2022].

Yang, L. & Shami, A., 2020. On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing,* 415(1), pp. 295-316.

Yildirim, S., 2020. *Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters.* [Online] Available at: https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167 [Accessed 28 05 2022].

Zolanvari, M., Teixeira, M. A. & Jain, R., 2018. *Effect of Imbalanced Datasets on Security of Industrial IoT Using Machine Learning.* Miami, 2018 IEEE International Conference on Intelligence and Security Informatics (ISI).

CHAPTER 9 Appendix A – Data Pre-processing

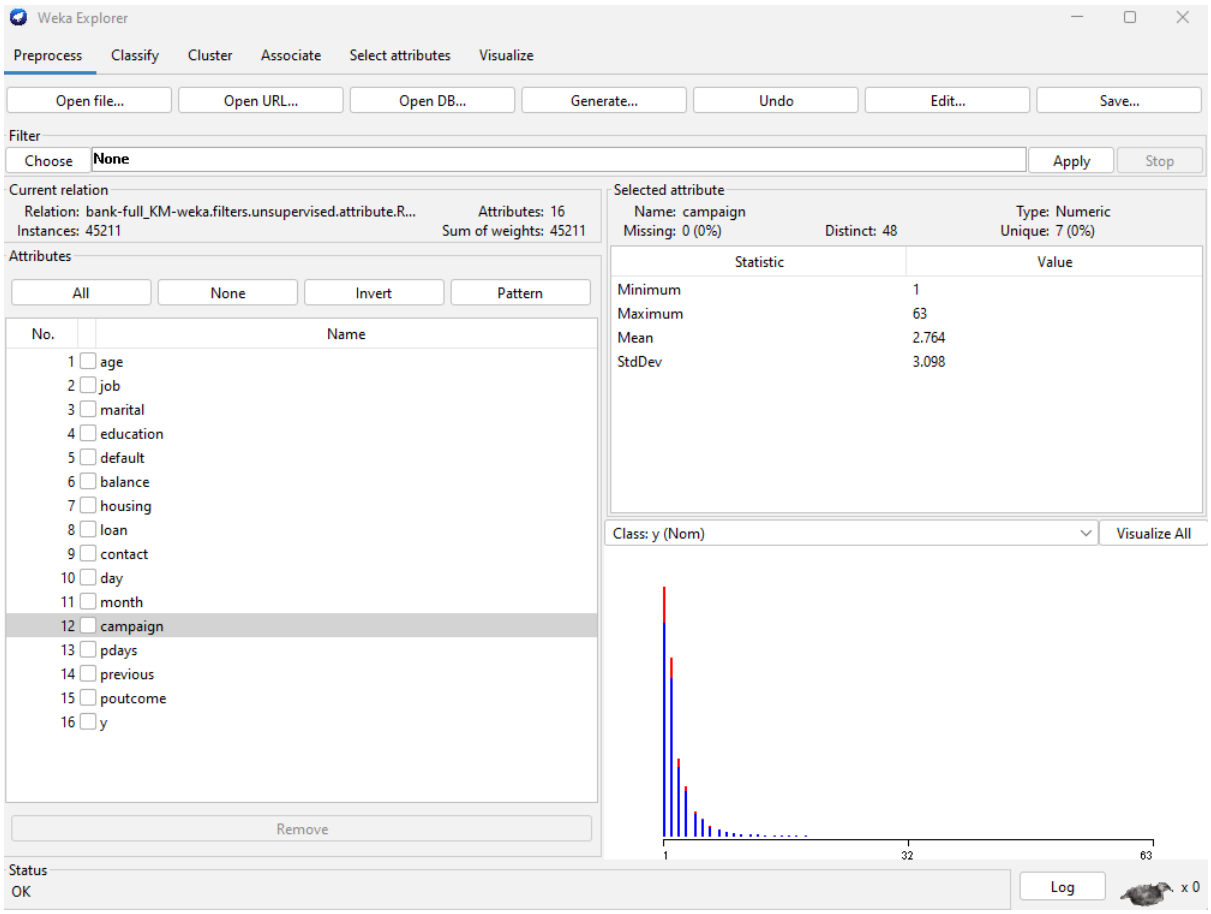

Figure 4 – Input data (Duration removed)

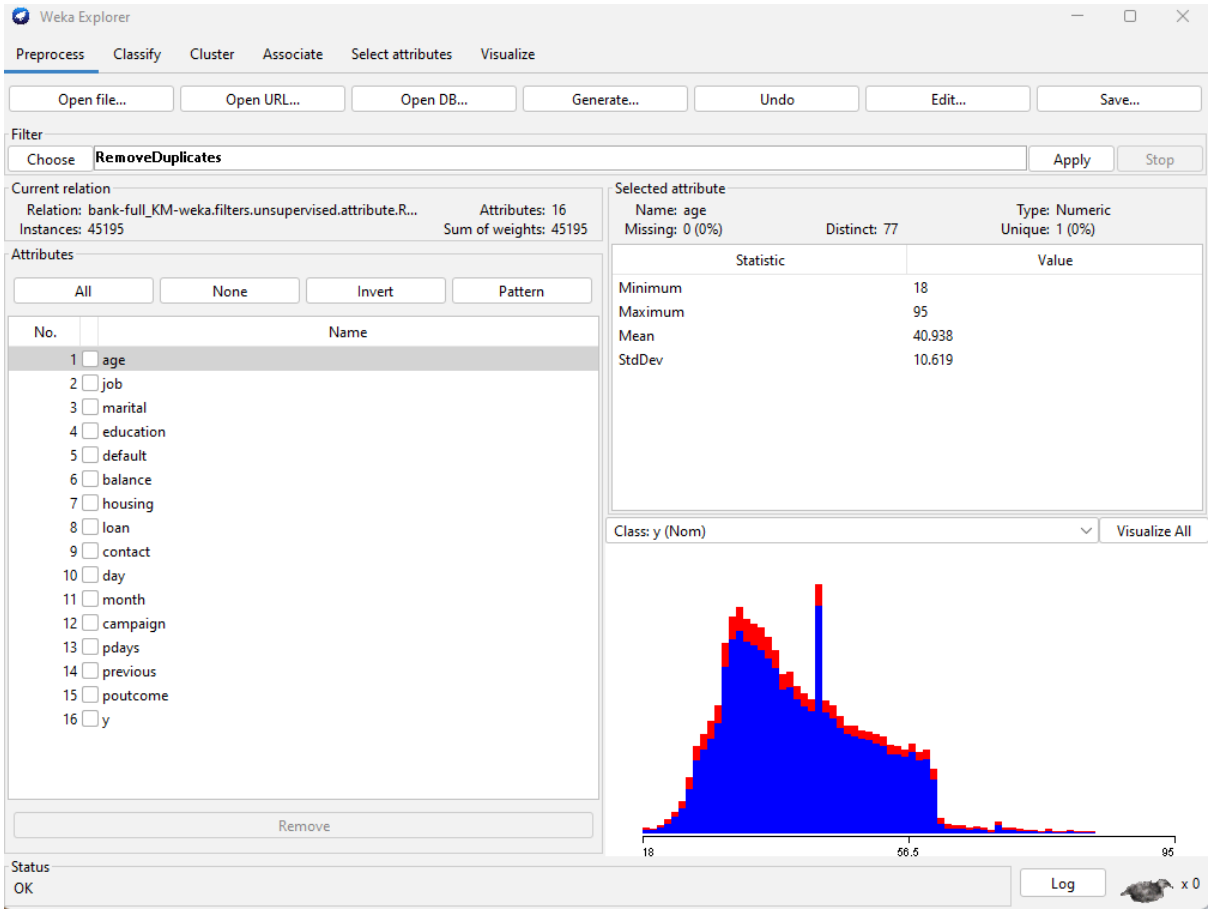

Figure 5 – Duplicates removed – (45195 instances)

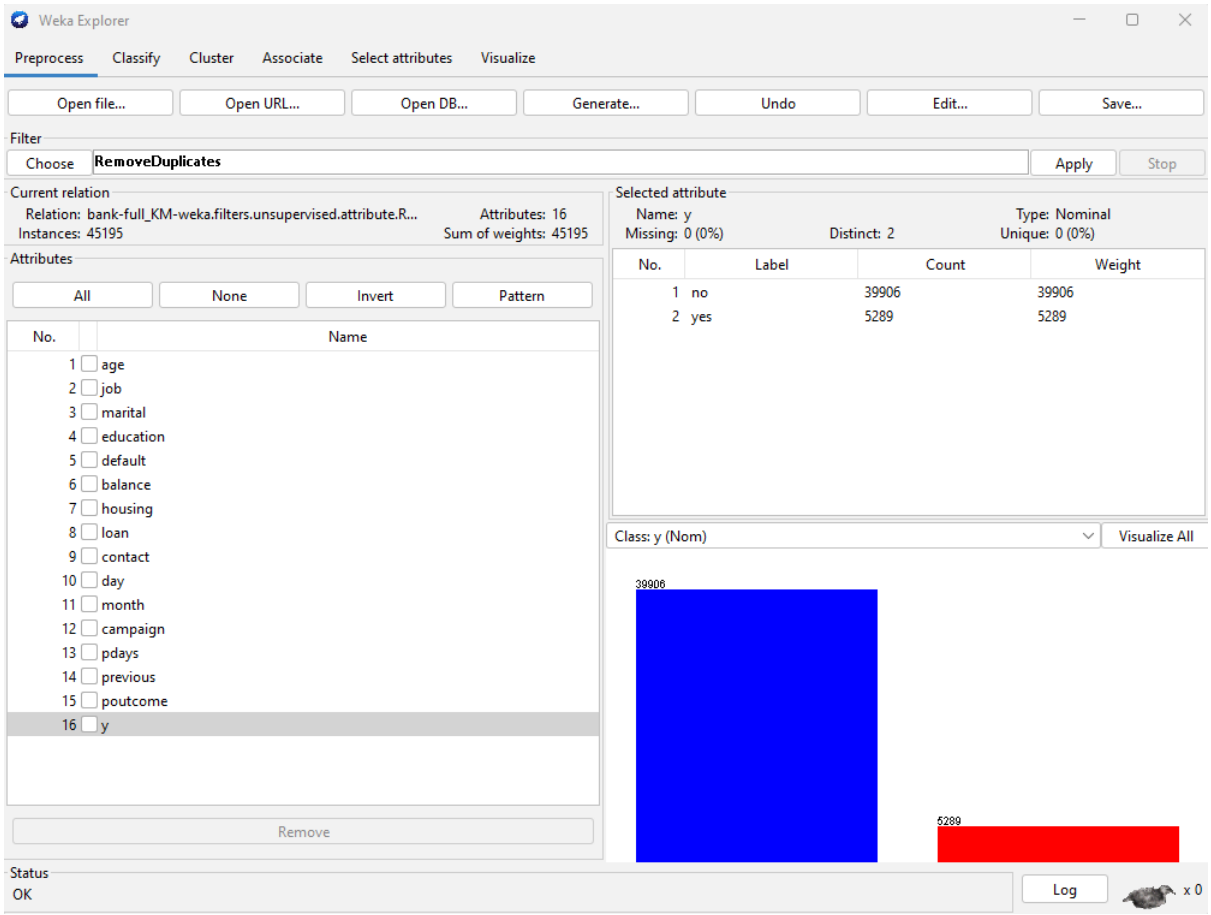

Figure 6 – Class balance

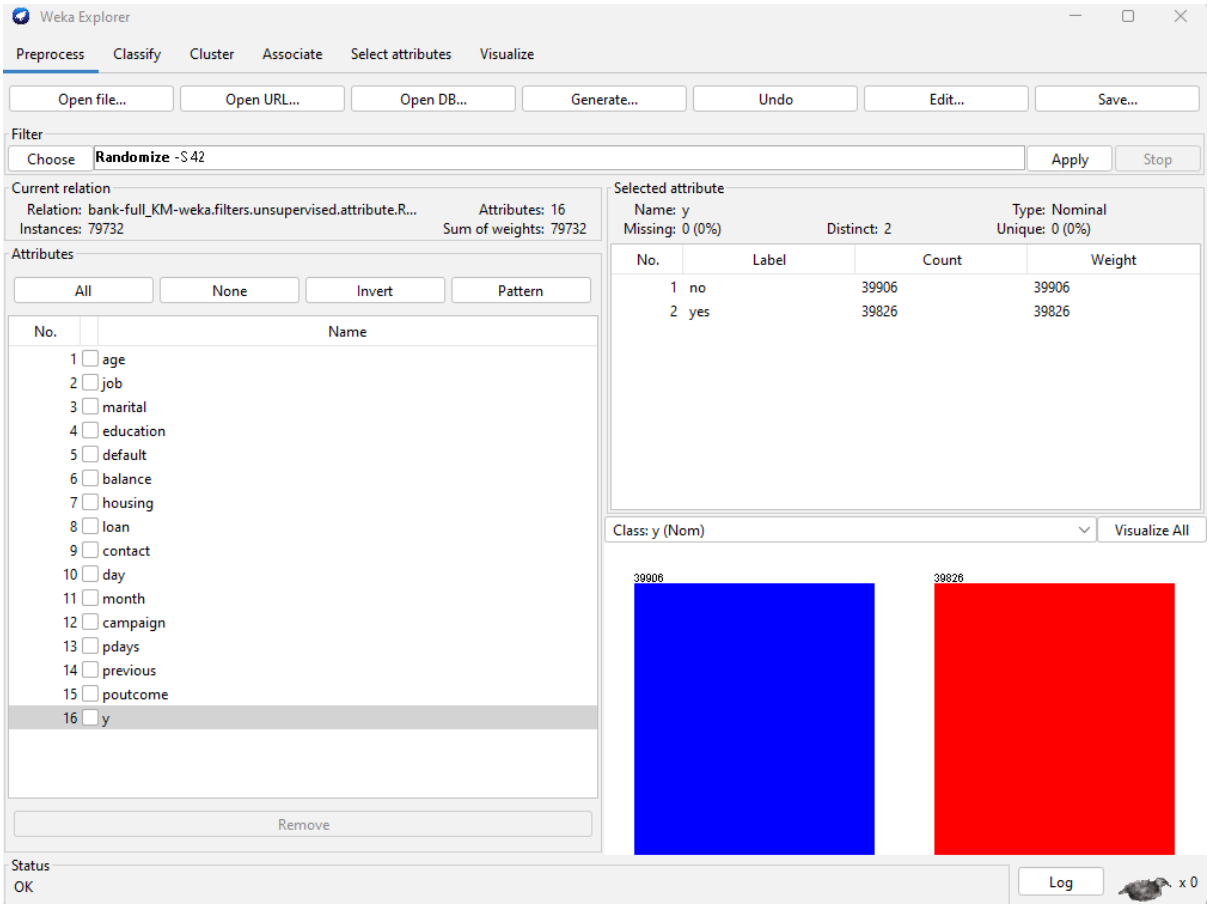

Figure 7 – SMOTE and Randomize applied

CHAPTER 10 Appendix B – Decision tree results

```
Time taken to build model: 5.69 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       71662               89.8786 %
Incorrectly Classified Instances      8070               10.1214 %
Kappa statistic                          0.7976
Mean absolute error                      0.1487
Root mean squared error                  0.2956
Relative absolute error                 29.7364 %
Root relative squared error             59.1177 %
Total Number of Instances            79732

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,921    0,123    0,882      0,921    0,901      0,798    0,926     0,887     no
               0,877    0,079    0,917      0,877    0,896      0,798    0,926     0,921     yes
Weighted Avg.  0,899    0,101    0,900      0,899    0,899      0,798    0,926     0,904

=== Confusion Matrix ===

     a     b   <-- classified as
 36739  3167 |     a = no
  4903 34923 |     b = yes
```

Figure 8 – Confidence factor 0.25

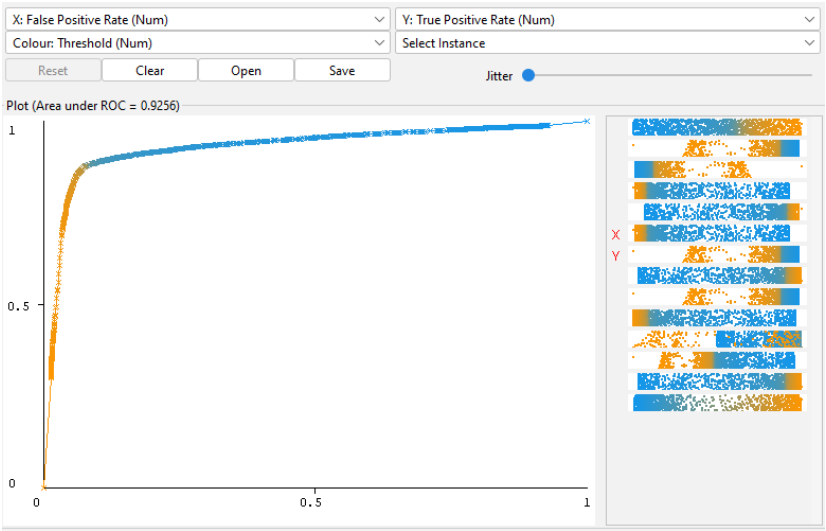

Figure 9 – ROC curve - Confidence factor 0.25

```
Time taken to build model: 5.16 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       71581               89.777  %
Incorrectly Classified Instances      8151               10.223  %
Kappa statistic                          0.7955
Mean absolute error                      0.1335
Root mean squared error                  0.2985
Relative absolute error                 26.7093 %
Root relative squared error             59.7004 %
Total Number of Instances            79732

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,910    0,114    0,889      0,910    0,899      0,796    0,921     0,878     no
                 0,886    0,090    0,907      0,886    0,896      0,796    0,921     0,912     yes
Weighted Avg.    0,898    0,102    0,898      0,898    0,898      0,796    0,921     0,895

=== Confusion Matrix ===

     a     b   <-- classified as
 36300  3606 |     a = no
  4545 35281 |     b = yes
```

Figure 10 – Confidence factor 0.5

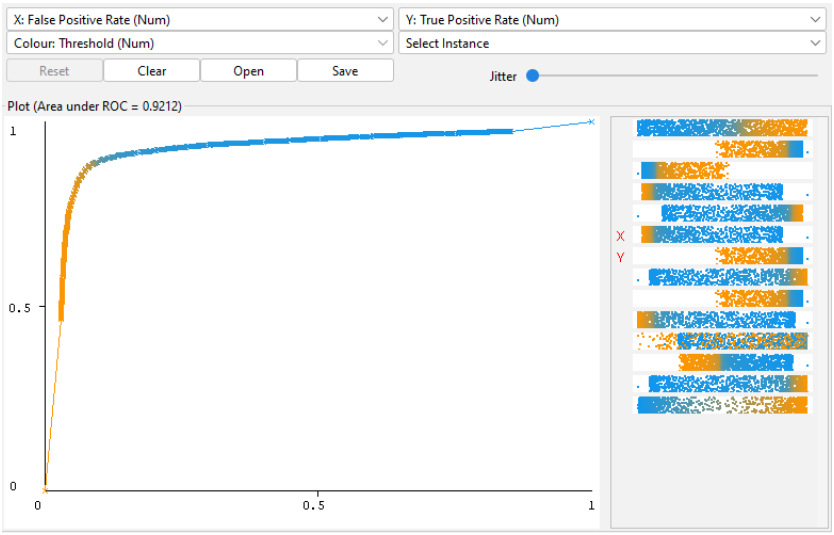

Figure 11 – ROC Curve - Confidence factor 0.5

```
Time taken to build model: 4.84 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.02 seconds

=== Summary ===

Correctly Classified Instances       24151               89.0885 %
Incorrectly Classified Instances      2958               10.9115 %
Kappa statistic                          0.7817
Mean absolute error                      0.1582
Root mean squared error                  0.3056
Relative absolute error                 31.6436 %
Root relative squared error             61.1103 %
Total Number of Instances            27109

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,903    0,121    0,884      0,903    0,893      0,782    0,920     0,884     no
               0,879    0,097    0,899      0,879    0,888      0,782    0,920     0,911     yes
Weighted Avg.  0,891    0,109    0,891      0,891    0,891      0,782    0,920     0,897

=== Confusion Matrix ===

     a     b   <-- classified as
 12372  1330 |     a = no
  1628 11779 |     b = yes
```

Figure 12 – 66% training split

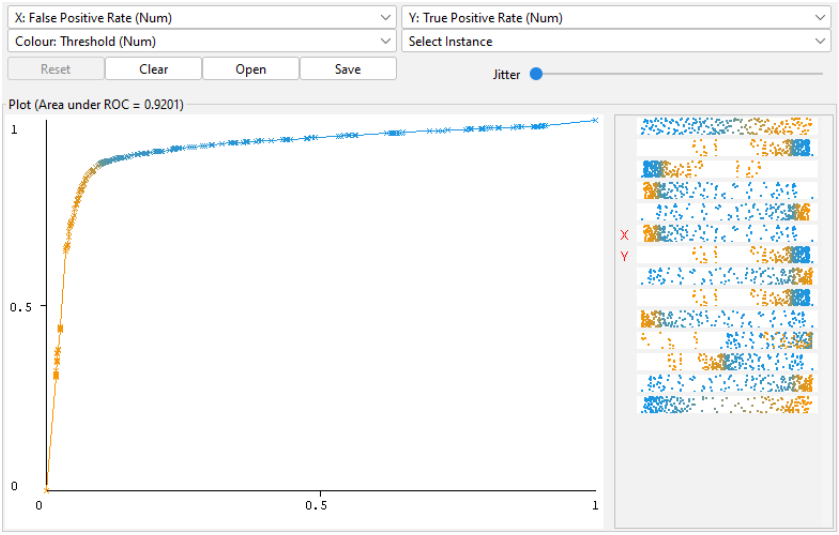

Figure 13 – ROC Curve - 66% training split

CHAPTER 11 Appendix C – SVM Results

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8-6" -seed 1
Relation:     bank-full_KM-weka.filters.unsupervised.attribute.Remove-R12-weka.filters.unsupervised.instance.RemoveDuplicates-weka.filters.supervised.instance.SMOTE-C2-K5-P653.0-S1-weka.filters.unsupervised.instance.Randomize-S42
Instances:    79732
Attributes:   16
              age
              job
              marital
              education
              default
              balance
              housing
              loan
              contact
              day
              month
              campaign
              pdays
              previous
              poutcome
              y
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 2316.12 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       66233               83.0695 %
Incorrectly Classified Instances     13499               16.9305 %
Kappa statistic                          0.6614
Mean absolute error                      0.1693
Root mean squared error                  0.4115
Relative absolute error                 33.861  %
Root relative squared error             82.2933 %
Total Number of Instances            79732

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
                 0,842     0,180     0,824     0,842     0,833     0,662     0,831     0,773     no
                 0,820     0,158     0,838     0,820     0,829     0,662     0,831     0,777     yes
Weighted Avg.    0,831     0,169     0,831     0,831     0,831     0,662     0,831     0,775

=== Confusion Matrix ===

    a     b   <-- classified as
 33587  6319 |     a = no
  7180 32646 |     b = yes
```

Figure 14 – SVM results for default values (10 fold cross validation)

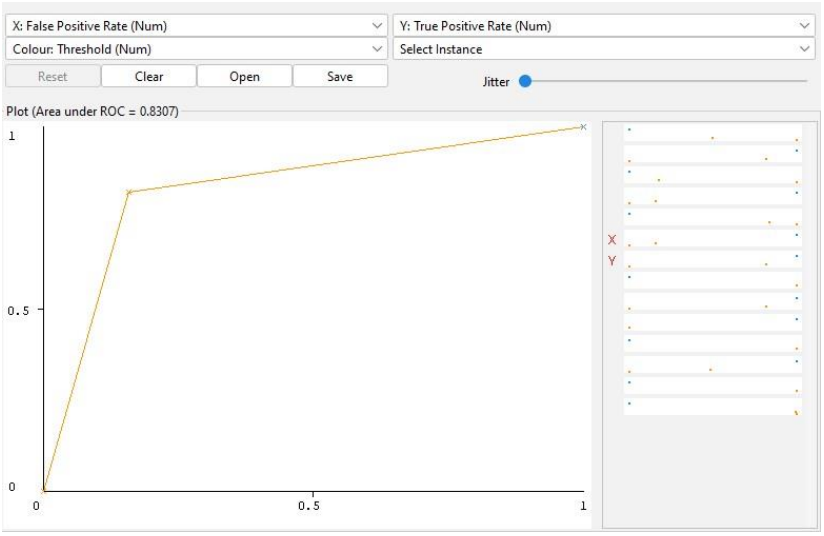

Figure 15 – ROC Curve for SVM - (10 fold cross validation)

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8-6" -seed 1
Relation:     bank-full_KM-weka.filters.unsupervised.attribute.Remove-R12-weka.filters.unsupervised.instance.RemoveDuplicates-weka.filters.supervised.instance.SMOTE-C2-K5-P653.0-S1-weka.filters.unsupervised.instance.Randomize-S42
Instances:    79732
Attributes:   16
              age
              job
              marital
              education
              default
              balance
              housing
              loan
              contact
              day
              month
              campaign
              pdays
              previous
              poutcome
              y
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 2759.24 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       67653               84.8505 %
Incorrectly Classified Instances     12079               15.1495 %
Kappa statistic                          0.697
Mean absolute error                      0.1515
Root mean squared error                  0.3892
Relative absolute error                 30.299  %
Root relative squared error             77.8448 %
Total Number of Instances            79732

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,860    0,163    0,841      0,860    0,850      0,697    0,848     0,793     no
                 0,837    0,140    0,856      0,837    0,847      0,697    0,848     0,798     yes
Weighted Avg.    0,849    0,152    0,849      0,849    0,848      0,697    0,848     0,796

=== Confusion Matrix ===

     a     b   <-- classified as
 34303  5603 |     a = no
  6476 33350 |     b = yes
```

Figure 16 – SVM results for C=3.0 (10 fold cross validation)

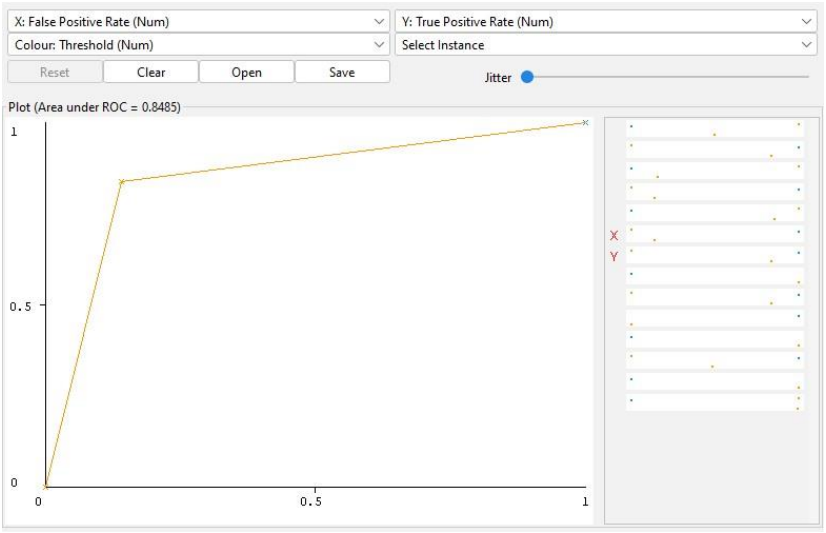

Figure 17 – ROC Curve – C=3.0

```
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 2933.59 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       67839               85.0838 %
Incorrectly Classified Instances     11893               14.9162 %
Kappa statistic                          0.7017
Mean absolute error                      0.1492
Root mean squared error                  0.3862
Relative absolute error                 29.8325 %
Root relative squared error             77.2431 %
Total Number of Instances            79732

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,862     0,160     0,843      0,862    0,853      0,702    0,851     0,796     no
               0,840     0,138     0,859      0,840    0,849      0,702    0,851     0,801     yes
Weighted Avg.  0,851     0,149     0,851      0,851    0,851      0,702    0,851     0,799

=== Confusion Matrix ===

     a     b   <-- classified as
 34405  5501 |     a = no
  6392 33434 |     b = yes
```

Figure 18 – ROC Curve – C=5.0

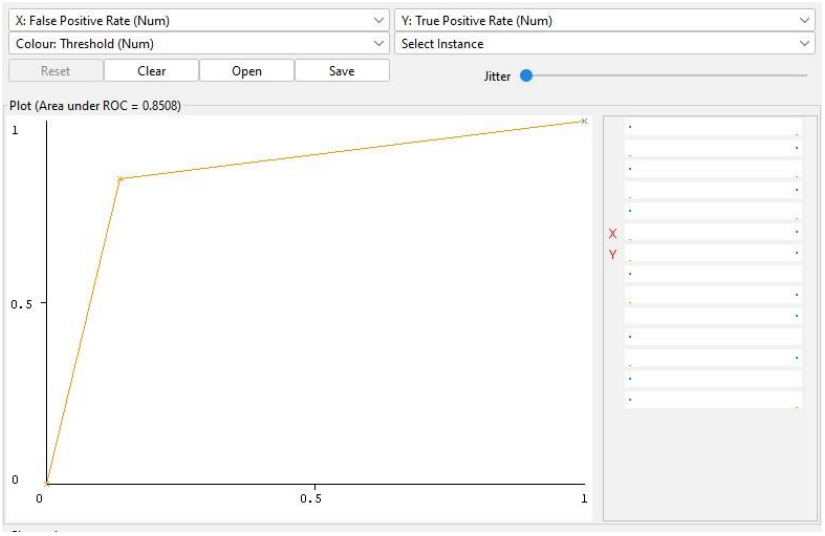

Figure 19 – ROC Curve – C=5.0

CHAPTER 12 Appendix C – Multi-layered perceptron

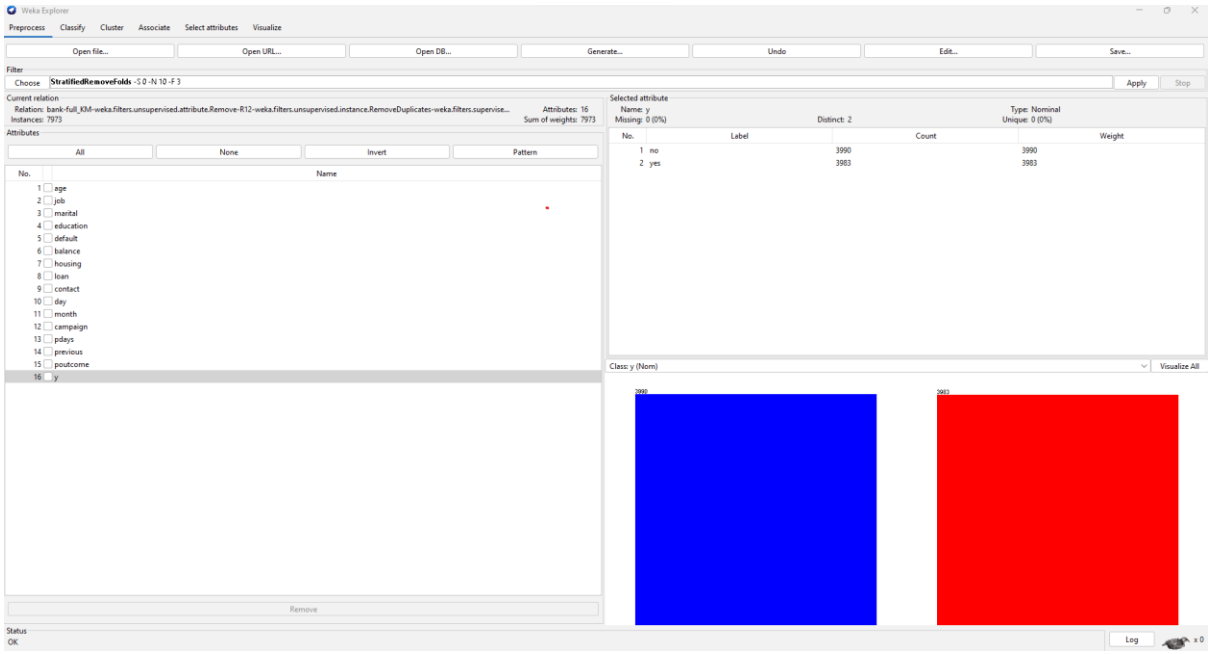

Figure 20 – Reduced dataset with class distribution unaffected (7973 Instances)

```
Time taken to build model: 57.58 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.12 seconds

=== Summary ===

Correctly Classified Instances        6199               77.7499 %
Incorrectly Classified Instances      1774               22.2501 %
Kappa statistic                          0.555
Mean absolute error                      0.243
Root mean squared error                  0.4148
Relative absolute error                 48.5919 %
Root relative squared error             82.9629 %
Total Number of Instances            7973

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,733    0,178    0,805      0,733    0,767      0,557    0,854     0,841     no
               0,822    0,267    0,755      0,822    0,787      0,557    0,854     0,851     yes
Weighted Avg.  0,777    0,222    0,780      0,777    0,777      0,557    0,854     0,846

=== Confusion Matrix ===

    a    b   <-- classified as
 2927 1064 |    a = no
  710 3272 |    b = yes
```

Figure 21 – Multi-layered perceptron - default settings

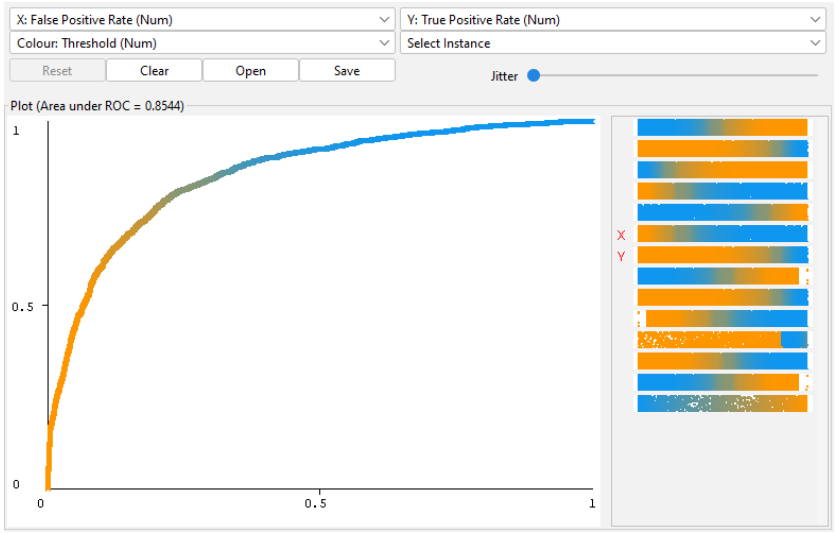

Figure 22- ROC Multi-layered perceptron - default settings

```
Time taken to build model: 12.65 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.05 seconds

=== Summary ===

Correctly Classified Instances        6372               79.9197 %
Incorrectly Classified Instances      1601               20.0803 %
Kappa statistic                          0.5984
Mean absolute error                      0.2645
Root mean squared error                  0.384
Relative absolute error                 52.9059 %
Root relative squared error             76.8075 %
Total Number of Instances             7973

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,800    0,202    0,799      0,800    0,800      0,598    0,872     0,859     no
                 0,798    0,200    0,799      0,798    0,799      0,598    0,872     0,872     yes
Weighted Avg.    0,799    0,201    0,799      0,799    0,799      0,598    0,872     0,866

=== Confusion Matrix ===

    a    b   <-- classified as
 3193  798 |    a = no
  803 3179 |    b = yes
```

Figure 23 – Multi-layered perceptron – LR=0.5, M=0.2, Hidden Layers=5

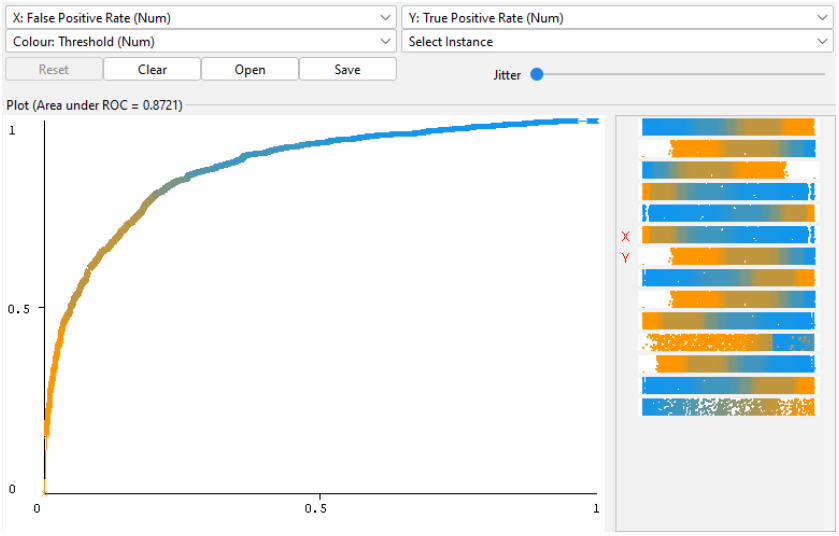

Figure 24 - ROC Multi-layered perceptron – C=0.5, M=0.2, Hidden layers=5