CUSTOMER DEFAULT PREDICTION IN BANKING

Using Machine Learning Algorithms Decision Tree and MLP

APRIL 9, 2018

ITNPBD6 2631564

Table of Contents:

Introduction

Bank data is growing rapidly as everything is digitized in the world today and every single transactions is captured by the system. These captured data are the historic data of their customer. Banks are providing lot of loans to their customers, but they face lot of dangers too with the current credit procedures followed. Banks faces a lot of risk today in lending a loan to their customer as the Defaulters rate is increasing. So the historic data is utilised by the banks today to foresee the behaviour of a particular customer over a period of time to take decision on his or her loan application.

In order to deploy the above method banks, need a perfect model which can do this process without manual intervention help the bank understand a customer's record and take decisions based on that. **To automate this procedure, an unbiased and a robust prediction model is built to recognize the customers fragments, those are qualified for Loan and mainly to save the bank from the risk of defaulters.**

In light of the above issue the onjective expects to taking care of an unpredictable issue of recognizing the customers to whom the loan can be endorsed insight of the past history of advances and distinctive others factors which will aid the banks to assume more educated praise scoring choices and diminish the defaulters rate later on.

Task: So, It is important to build a model to predict if the person is eligible for loan or not ,in other words if the customer is more likely to default the loan or not ,based on the various factors from the historic data available.

Buliding a predictive model to help the banks in predicting the probability of a customer defaulting in the future based on the historic data is the ultimate objective of the project . For which **Structured methodology** is followed with the help of **Weka** , in which the **historic data** of the customers are **cleaned , splitted for training**,**validation and testing**, **independency are checked using SPSS ,classification** algorithms were used like **J48, MLP**,**base line accuracy** for the available data set is calculated, models were repeatedly tested and trained with **Hold out method** and **K-fold cross validation** methods ,**significant features are selected** and the models were re evaluated for **accuracy and precision comparison** based on the **confusion matrix**, model errors are address with the **ROC** ,also the final model is treated with **cost sensitive learning** to increase the model efficiency.Individual model outputs are captured and compared with each other to get the best model to predict the Defaulter.

Data Summary

We have managed to get a data set of **2000 customers of a bank** in support of the objective to find the loan defaluters . There were 14 different variables in the data set with a mixture of numeric and nominal variables. The Target variable was nominal is nature with only two values **“Defaluted” and “Paid”**. There were **13 independent variables** in the data set to explain the customer’s history with the bank . Those independent variables provided various important imformations about the customer in detail . Indepent variables were a mixture of nominal and numeric data. List of independent variables found in the data set are – Customer Id,Gender, Age, CCJ,Years at address, Employment status,Loan amount, Income,Own Home,Country,Post Code,Current Debt,Fictional surname. Amongst these 13 independent variables ,

Postcode was alphanumeric in nature ,**Fictional Surname** is a text or string data type .

Numeric variables: Age,Years at address,Current Debt,Income,Loan Amount,CCJ's

Nominal variables: Gender,Employment status,Country,Own Home,

Customer ID – It is the unique Identification number allotted by the bank for their individual customers.

Age -By name itself it gives the age of the particular customer.

Gender – It says the sex of the customer amongst two values “M- Male” or “F-Female”

Fictinal Surname – This gives the Surname of the customer.

Country – It tells the country to which the customer belongs to .

Post code – This gives the post code of the area in which the customer is living in the country .

Employment Status - This gives the current status of emplyment of the customer.It has four values in the dataset – Employed ,Unemployed,Self Employed ,Retired.

Own Home – This gives the information about the living home status , it takes three values in the dataset – Rent, Own ,Mortgage

Years at address – It gives the number of years a particular customer is llliving in that address.

Current Debt – It gives the current liabilities or obligations that are in due corresponding to the customer .

Income – This gives the income of the customer per annum.

Loan amount – This tells the amount of loan given by the bank to that particular customer.

County Court Judgement (CCJ) – A CCJ is passed out by a County Court. In the event that a man neglects to reimburse an obligation, the organization or individual they owe cash to can go to the court to attempt and get it back.

Variable	Data Type	Values/Range
Customer Id	Numeric	555574 to 1110985
Age	Continuous Numeric	17 to 89
Gender	Nominal	"F" and "M"
Fictional Surname	String	No Range and Multiple values
Country	String	UK
Post Code	Alphanumeric	No Range and Mutiple values
Employment Status	Nominal	Employed, Unemployed,Self Employed,Retired
Own Home	Nominal	Own Home, Rent , Mortgage
Year at Address	Numeric	1 to 71
Current Debt	Continuous Numeric	0 to 9980
Income	Continuous Numeric	3000 to 54500
Loan Amount	Continuous Numeric	13 to 54455
CCJ	Nominal	0,1,2,3
Outcome	Nominal	"Defaulted" and "Paid"

Data Preparation

Customer ID,Surname,Postcode these three variables are poor choices for any machine learning model because firstly,they have no strong relationship with the target variable , secondly the have great range or many different values which cannot be used to categorise the customers based on that and also these diffent values increases the complexity of the model .Thirdly there is very less scope for generalising these variables.Considering all the above factors stated above these variables(Customer Id,Surname,Postcode) are removed from the dataset.Now the data set has 10 variables including the target variable.

Gender has many values like Female , Male , F , M, N, D,H,0,1 etc .In general, Gender shoud have two values Male and Female, but there are lot of other values in the dataset.Under the values F and M there are 1985 records (Maximum percentage of the whole dataset)-that forms the base of the gender types.Converting 4 records under Female to “F “and Converting 3 records under Male to “M”.Removing the records (8 records) corresponding to othe types under gender like –N,D,H,0,1 now 1992 records are in the dataset under F and M category .

Years at address – Scatter plot and Histograms for the data showed the outliers clearly .Outliers spotted - 250,300,410,560 (A person living for these many years is humanly impossible and it is shown clearly by the plots).These 4 records are removed from the dataset.Now the size of the data set has 1988 records

Country -There are four countries - UK,Germany,Spain,France.More than 95% of the customers belongs to UK and Other three countries have two records each ,Removed those 6 records from the data set and now the dataset has the customers belonging to UK .Now the data set has 1982 records ater removing 6 records corresponding to other three countries.

Most of the **Income** of the customers in the dataset ranges between 3000 to 54500 GBP per annum.Only two customers seem to have extreme values - 220000 ,180000 GBP per annum.These are clearly outliers in the data set ,removed these extreme valuesa and now the data set has 1980 records.

CCJ's -Most of the customer records have the CCJ's between 0 to 3 ,Only two records had CCJ values 10 and 100 –which are extreme values .These two records are removed from the data set and now the data set has 1978 records.

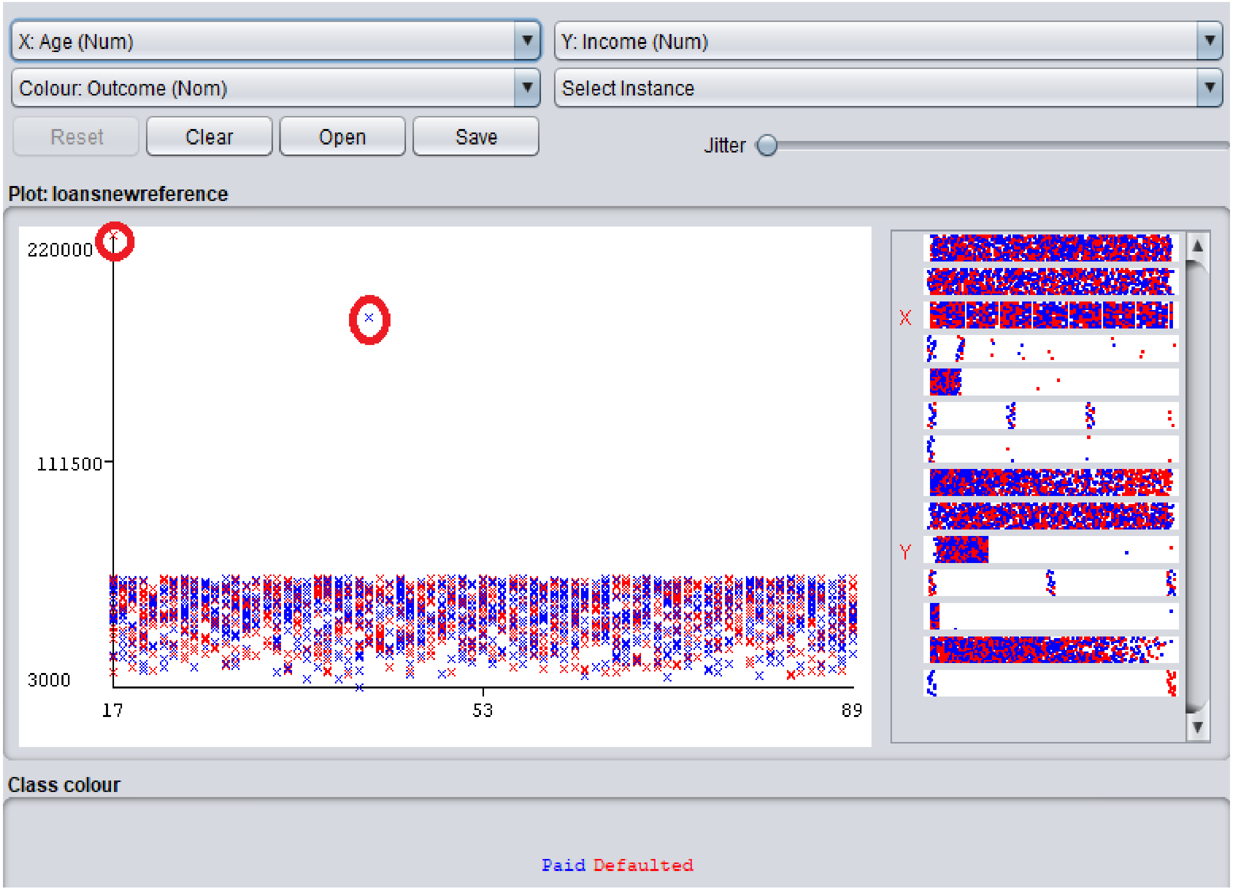
One Example scatter plot and histogram corresponding to Income data is shown below with and without the spotted outliers.

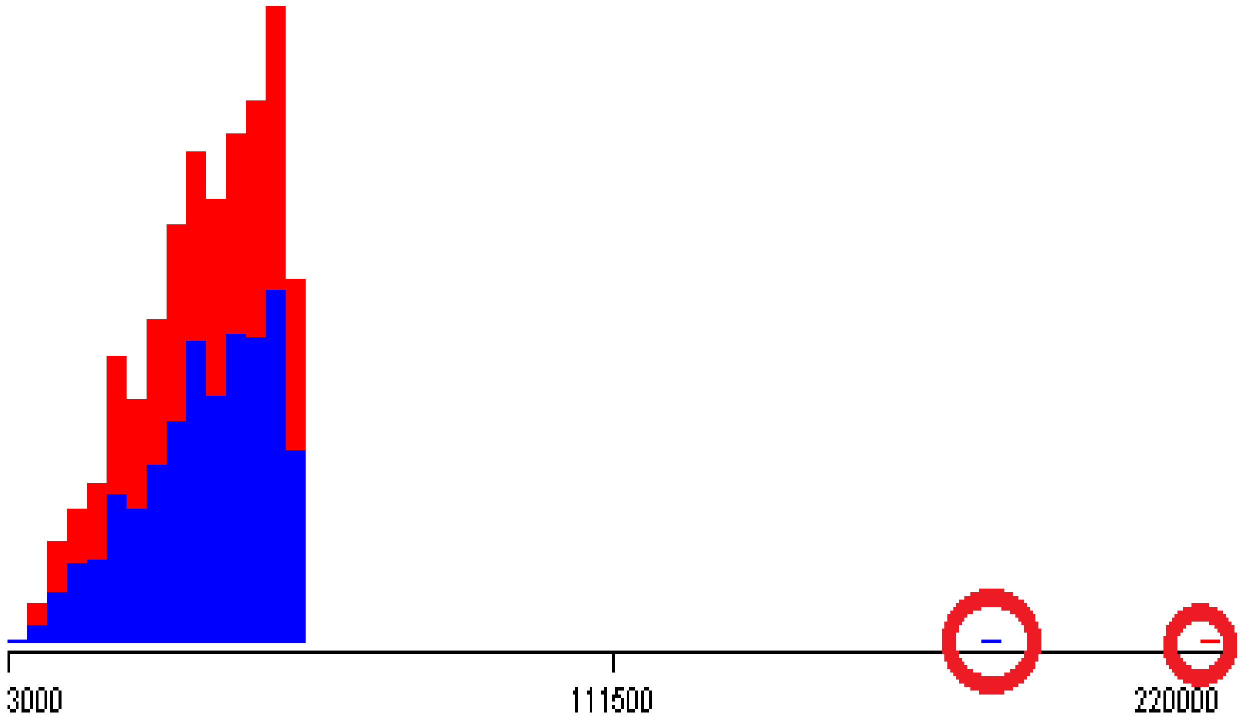
Outliers Spotted and Removed in the Data Set:

- 8 records under other categories variable(Gender) like H ,D,N,1,0 - Considering them as outliers.
- Years at address - 4 records corresponding to values (250,300,410,560)
- removing 6 records corresponding to other three countries -Spain,France,Germany
- Removed two extreme values in the income variable -220000 ,180000
- Two records had CCJ values 10 and 100 - etreme values

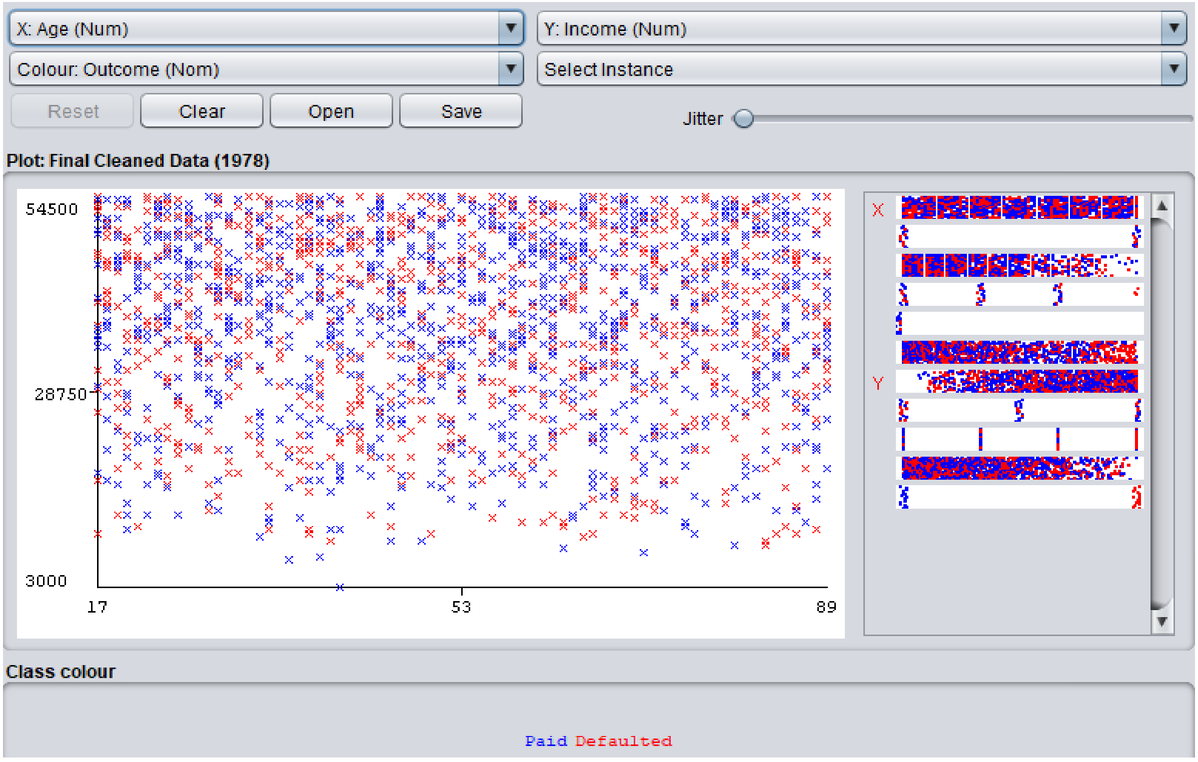
Totally 22 extreme values are spotted and removed from the data set ,22 is less than 2 % of the whole dataset, so this should not affect the quantity of the dataset required to classify the defaulters and non defaulters, since we have enough number of records to use for the modelling.

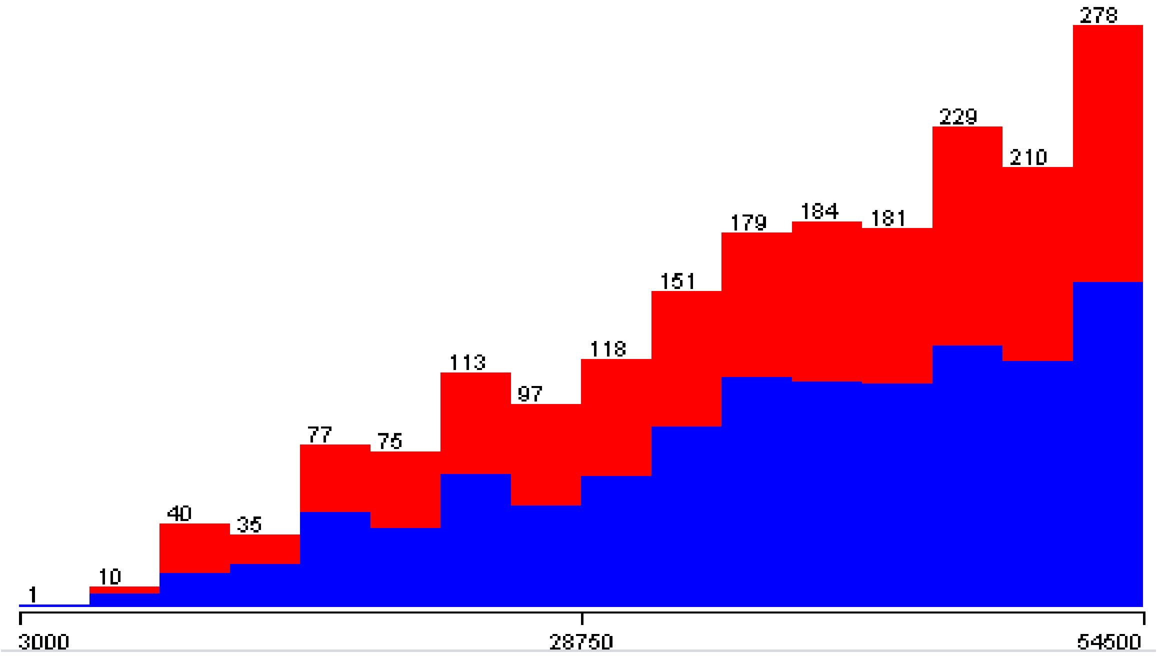
Income Data with spotted Outliers:





Income Data after removing the outliers :





(For all the spotted outliers for other variables in the scatter plot and histograms please check the appendix)

Modelling

After the detailed sweep through the data set we could see that the Target variable is nominal with two distinct classes – “Defaulted” and “Paid”. Since we have two distinct classes in the dependent variable we can classify the customers to one among the two classes based on the data we have. A planned approach for the model design is very important to get the best estimate of the model to classify the customers. Based on the target, we can choose two different algorithms to classify the customers- **Decision Tree and Multilayer Perceptron.** These two algorithms are used in the model design in the further sections, the models based on these algorithms are compared based on their individual accuracy. Individual algorithm’s assumptions are discussed and data set is rechecked based on them before feeding into the model. A structured methodology is followed for each model based on each chosen machine learning algorithms as listed below:

Assumptions of Algorithms

Decision Tree (J48): Decision tree is a classic Supervised learning which can be used to build a predictive model in which the dependent variable is driven from the various independent variables. Decision tree comes into the play when the target variable is discrete- nominal is nature. **$(x,Y) = (x_1,x_2,x_3,\ldots..x_k,Y)$,** where x is the vector of all the predictor variables, Y is the target variable

Assumptions of Decision tree:

- Decision tree implicitly performs feature selection
- Variable transformation and normalisation is not required for decision tree.
- Multicollinearity is not a problem for decision a tree model.

MLP: It is the free forward Artificial Neural Network, with Supervised learning and backpropagation techniques. It has input layers, hidden layers and output layer.

Assumptions of MLP

- Input layer is linear in nature
- Hidden layers are always non linear
- Muticollinearity is an issue MLP and the activation function(sigmoid)

Sigmoid function (activation function) – $S(x) = 1/1+e^{-x} = (e^x/e^x + 1)$

From the above understanding on the assumptions of the algorithms chosen we need to check for the multicollinearity for the case of MLP as shown in the below section,

Multicollinearity check based on the assumptions:

From the assumptions of Decision tree we know that multicollinearity should not affect these models, **Multi-Layer Perceptron is nonlinear and the multicollinearity does have an effect on the model accuracy,** so multicollinearity should be checked amongst the independent variables and removed. To be precise this multicollinearity and increasing standard errors makes the independent variables statistically insignificant when they are actually not.

One approach to quantify multicollinearity is the **Variance Inflation factor (VIF),** which checks how much the difference of an expected regression coefficient increments if the predictors are associated.

Muticollinearity check has been performed through SPSS and the VIF values for all the combinations of the independent variables are checked, one example is shown below,

Coefficients[a]

Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta	t	Sig.	95.0% Confidence Interval for B Lower Bound	Upper Bound	Collinearity Statistics Tolerance	VIF
1	(Constant)	.819	.489		1.675	.094	-.140	1.778		
	EmpStatus	-.190	.838	-.005	-.227	.821	-1.833	1.453	.995	1.005
	CurrentDebt	1.163E-5	.000	.032	1.433	.152	.000	.000	.997	1.003
	Income	-9.321E-7	.000	-.010	-.407	.684	.000	.000	.807	1.239
	OwnHome	.469	.659	.016	.712	.476	-.823	1.761	.993	1.007
	Age	-.002	.002	-.038	-1.296	.195	-.005	.001	.587	1.704
	Year@address	.003	.002	.041	1.383	.167	-.001	.007	.586	1.706
	Gender	.036	.048	.017	.746	.456	-.059	.131	.997	1.003
	LoanAmount	.001	.001	.032	1.265	.206	.000	.002	.813	1.229

a. Dependent Variable: CCJ

CCJ is one among the predictors which is taken as an dependent variable against the other variables for the Multicollinearity diagnostics. And the output in SPSS confirmed that all the values against the predictors are around 1 which confirms that there is no multicollinearity amongst the independent variables. The same procedure has been followed for all the combinations of predictors and the results showed that there is no multicollinearity issue amongst the independent variables in predicting the target.

Splitting Data

It is a good practice to split the whole dataset into Train, Test and Validation tests. We can split the good proportions of data to train and test sets to help the model better in classifying the customers and testing the same model. Since we have validations options in Weka, we need to separately segregate data for validation and use that for the k-fold cross validation of the model. Total number of records on the whole data set = 1978 (after data preparation and removing outliers)

80% data as train set – 1582 records,10% data as Cross Validation set – 198 records and another 10% as test set -198 records. 80-20 is a good split. In Weka, we can use the whole training set in the model also we can split the data internally with Weka for training and then the same model can be tested with the supplied test set.

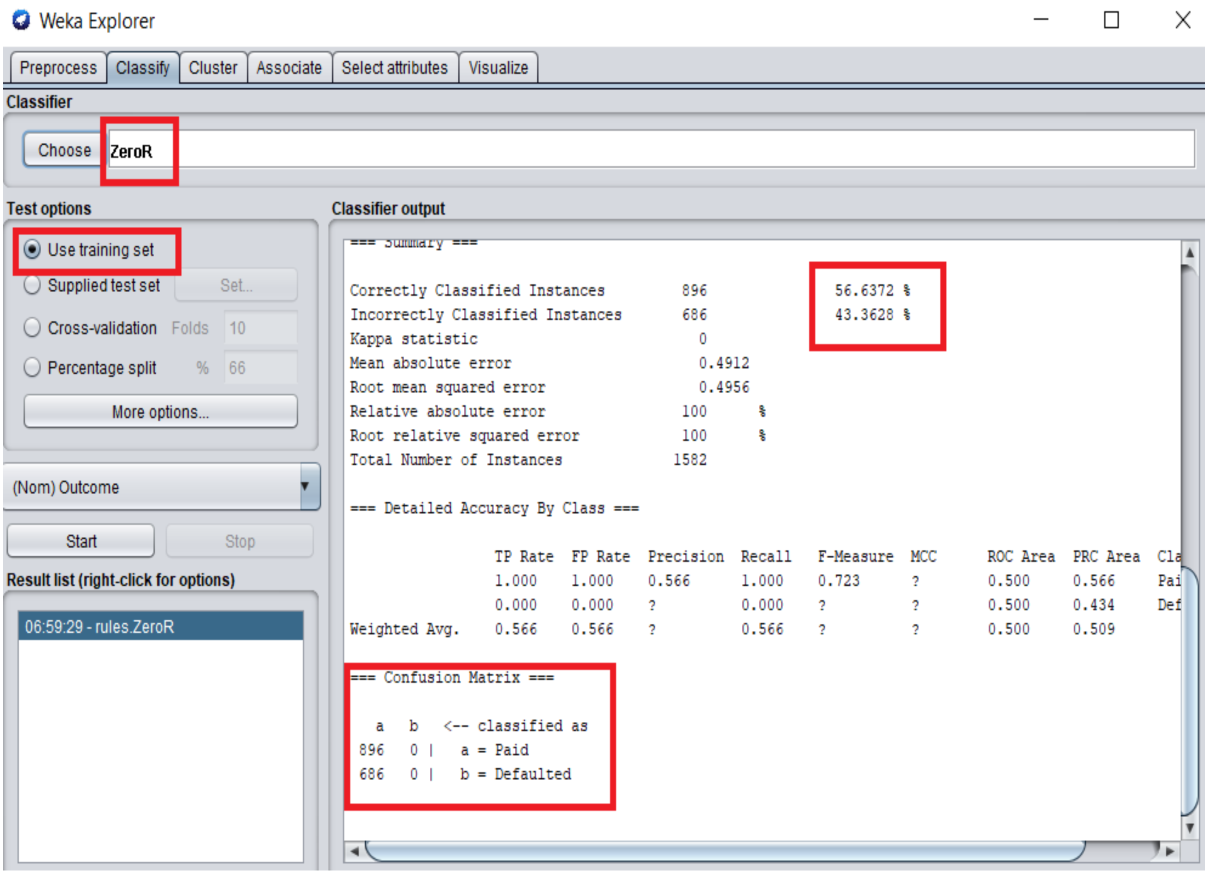
Resample filter is used in splitting the data into train and test set with no replacement and with invert selection option in order to ensure the same data points are not chosen in both train and test set.

Baseline accuracy using ZeroR classifier in weka.

Base line accuracy is the measure of basic accuracy value for a basic model like ZeroR in Weka for that particular dataset. This baseline accuracy value can be used to compare the other models built with different algorithms. Since we can fix the baseline accuracy, the performance of other models can be evaluated based on this. Baseline accuracy acts as the threshold value, and all the model accuracy should be above this threshold value. It is important to check the Kappa static, which gives an normalised accuracy value by the baseline. Data set has 1582 records with **Paid – 896, Defaulted – 686**, based on this we can guess the baseline accuracy and compare with the ZeroR model accuracy.

Guessed baseline accuracy based on the Negative result (Paid) = 896/1582 *100 = 56%

Checking the base line accuracy using the ZeroR classifier over the training data in Weka

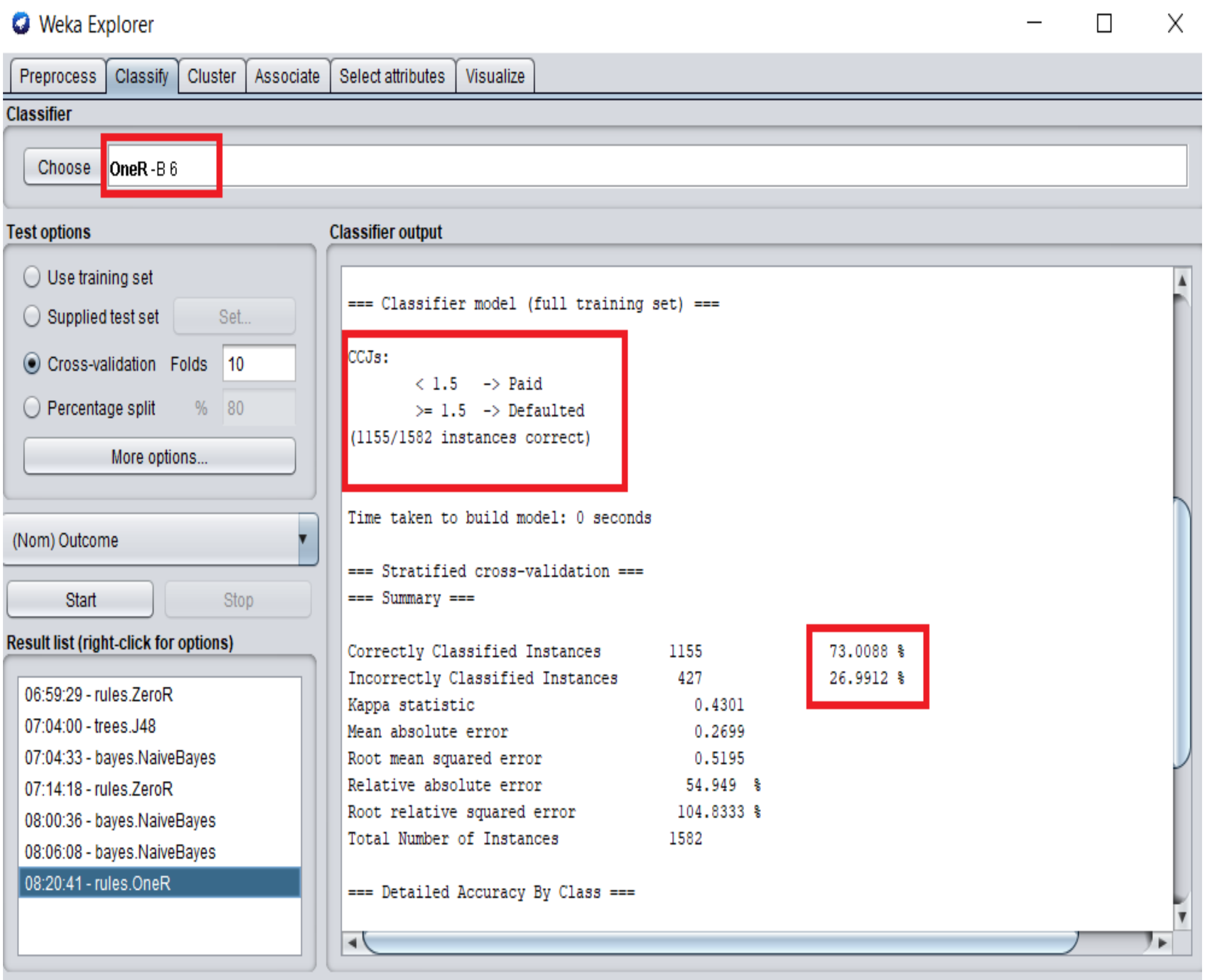


From the above screenshot we can see that the basic ZeroR model in Weka has an accuracy around 56% as guessed earlier manually. Kappa statistic is 0 in the above model. (Kappa statistic should be above 0 for all the new models). We can see that the above model shows that there are 896 Paid customers and 686 Defaulted customers in the Dataset.

Now the baseline accuracy for the model estimate over the take dataset is 56%, so we expect all the other models to perform over this accuracy value. And any Kappa Statistic value above 0 show some progress on the model performances.

Rule classifier: OneR classifier in Weka gives the significant rule based on which the data is mainly classified. **It Creates a basic rule based on the most significant variable that has high information gain.** Classifier output of OneR shows the rule based on the variable CCJ, which is the significant variable based on which the customers are classified. We can see that this rule has classified 1155 instances correctly out of 1582 of the training data. This rule based on CCJ classifies more than 70% of the customers correctly.

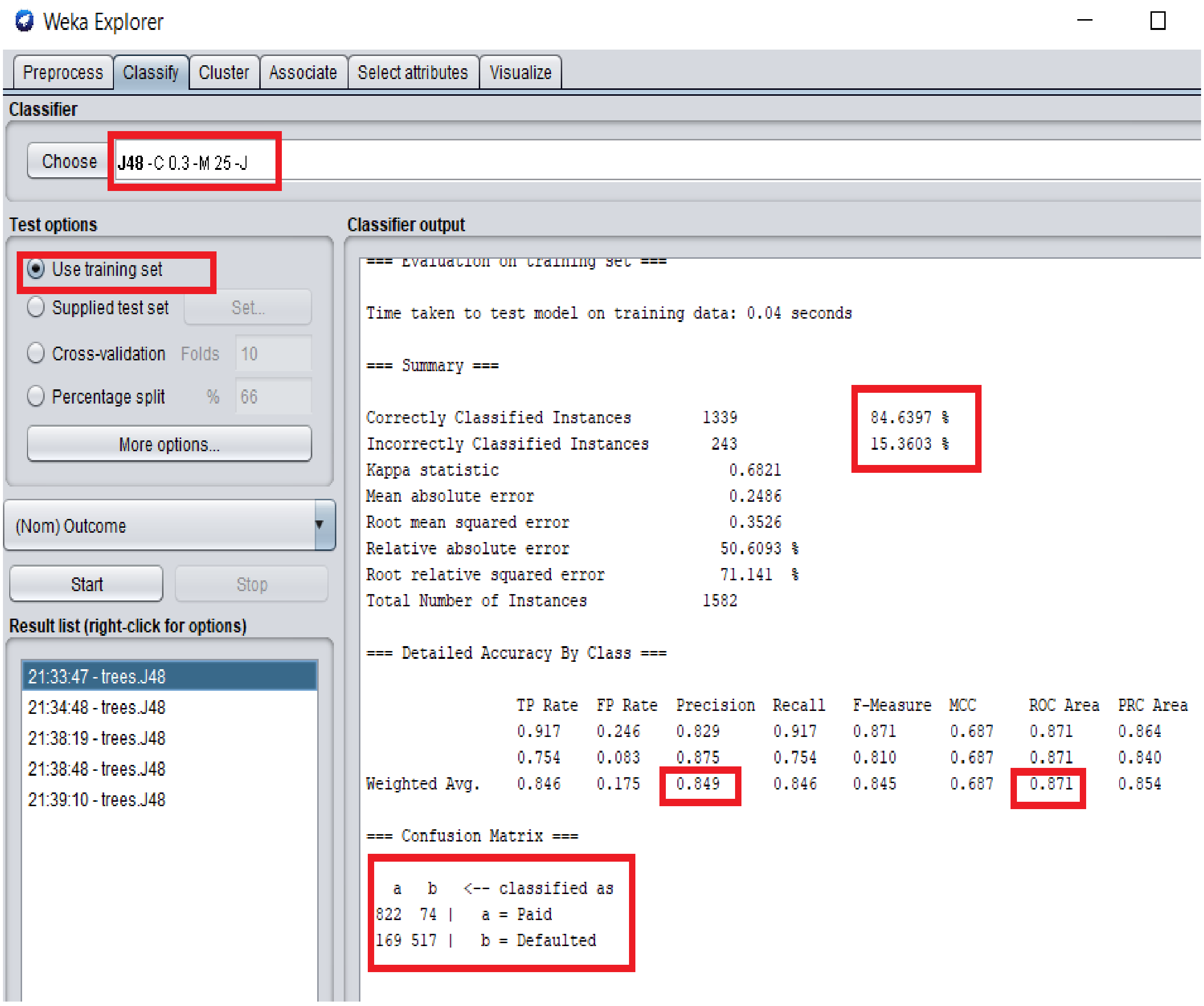
Any classification is based on a rule in which the significant variables are included, OneR gives the rule with the most significant variable to classify the customers, if that variable is removed in the dataset then it finds the next significant variables to classify. **MinBucketSize is used with the standard values in OneR to avoid over fitting.** Hence a check has been placed to avoid the over fitting of the models.

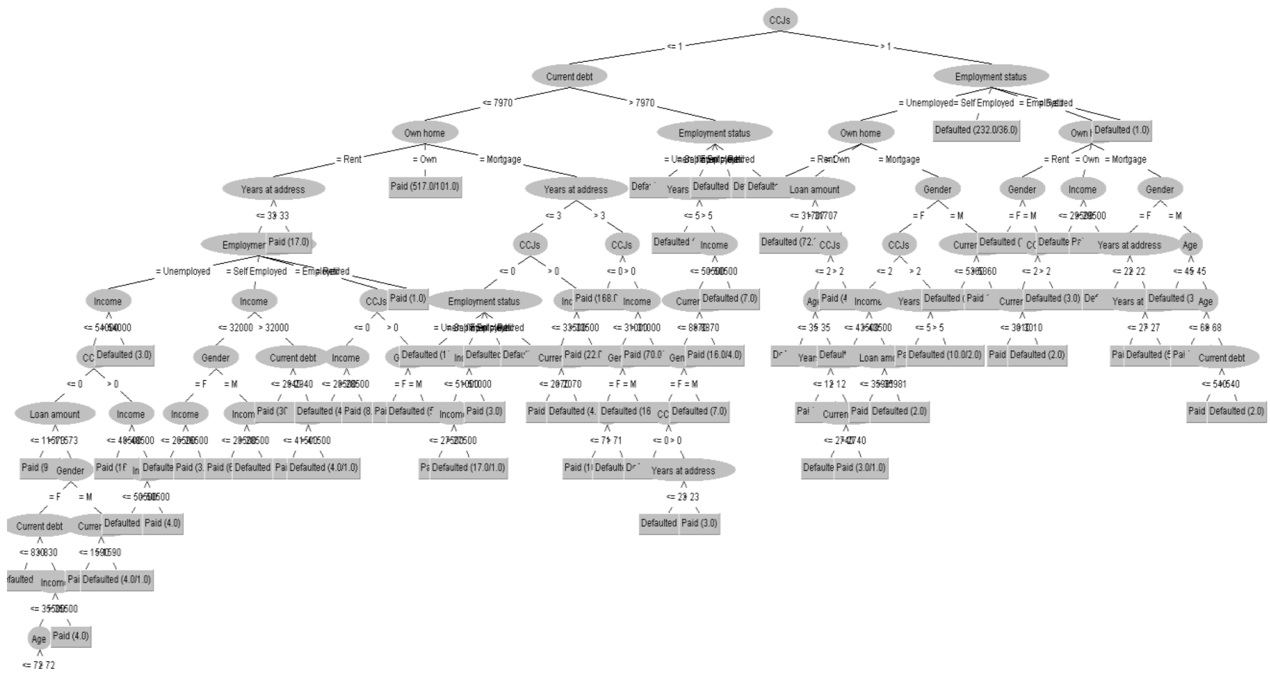


Use Training set:

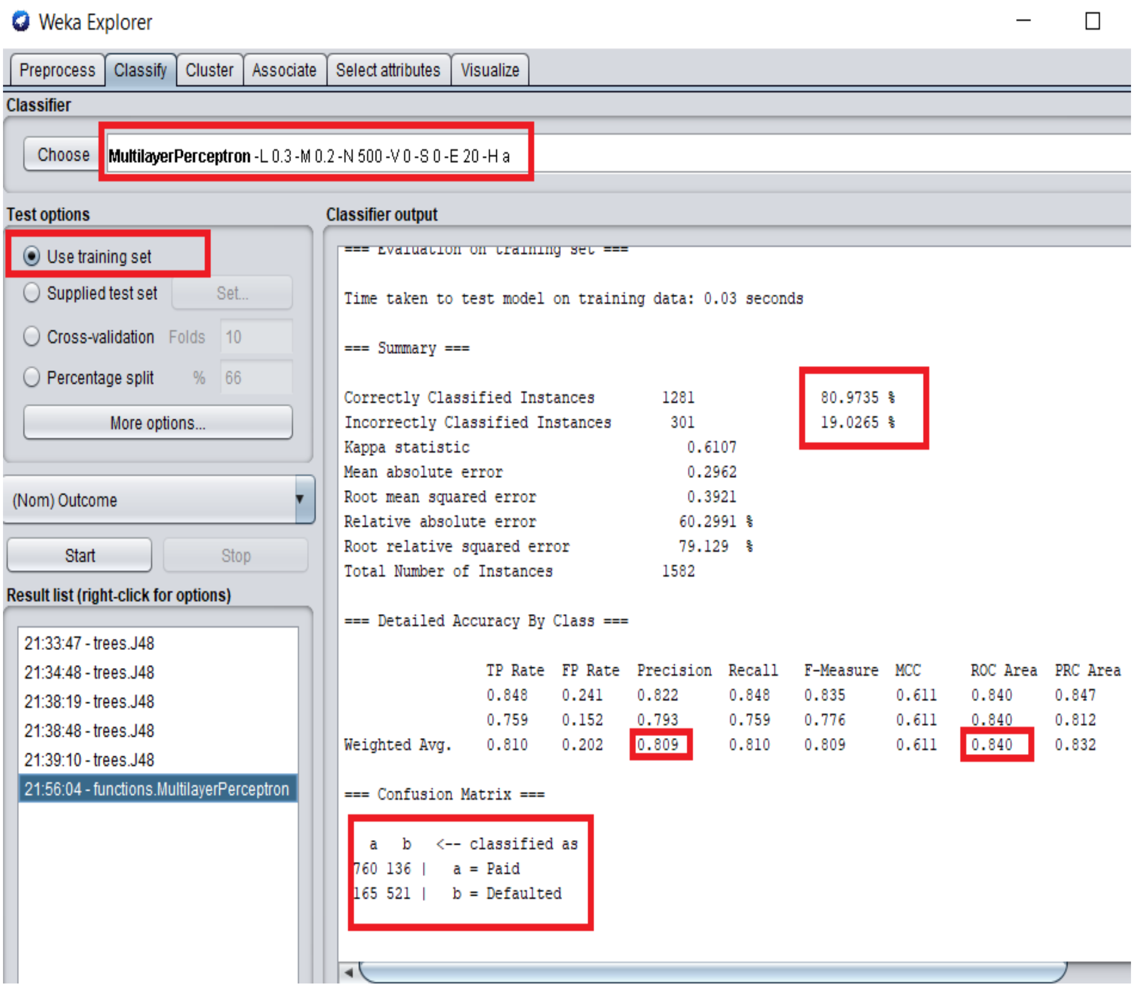
With this test option is Weka we can train the model with all the data points in the training set. With this the models are checked with the default hyper parameters and then those values are changed one after the other Until the best model is achieved with better accuracy, precision and Defaulters prediction rate compared to all the combinations of the hyper parameters tried earlier with the same model.

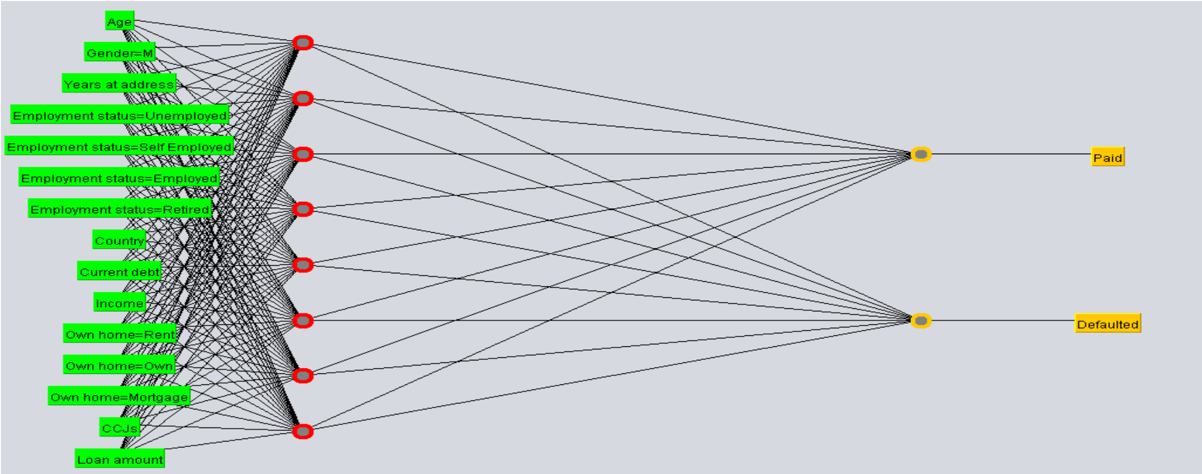
Training the J48 classifier model with all the basic hyper parameters and the whole training set is shown below with the accuracy, precision and structure of the tree,





Training the MLP model with all the basic hyper parameters and the whole training set is shown below with the accuracy, precision,





From the above screenshots we can see that, since all the data points (1582) are utilised in training so both the model performance seems to be high, but without evaluating the model with different hyper parameters and with other better methods of training we can't take a decision in choosing the best model. One other reason to re-evaluate with other methods is that, when the whole training set is used the data fed into the model is not randomised, it is important to check the model performance with stochastic data points and see the model efficiency.

Hyper parameters:

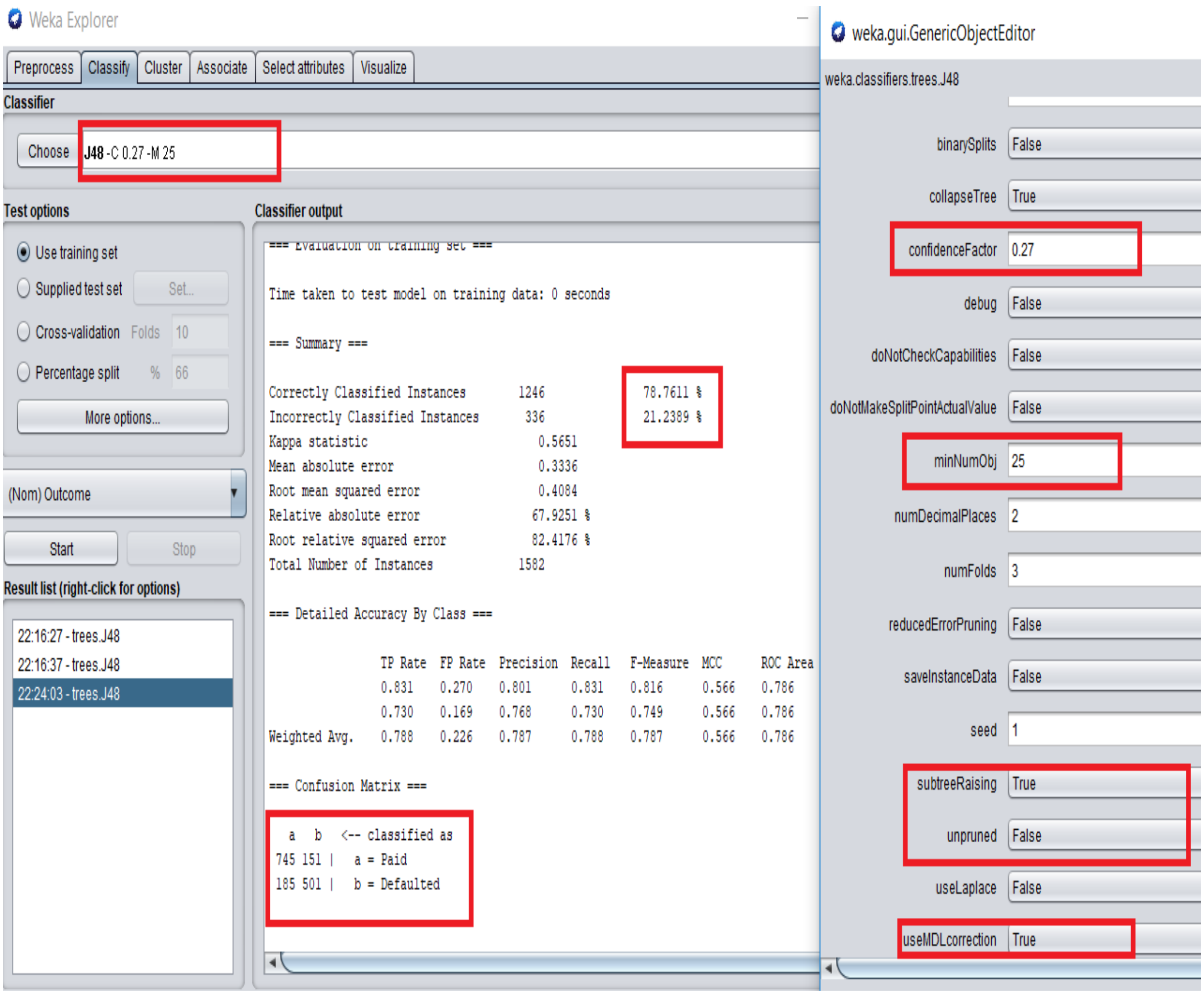
Both the Models have various hyper parameters, they are all checked in different combinations and evaluated based on the results of the models obtained and the best hyper parameters are chosen for the model to perform with better accuracy in predicting the defaulters and to optimise the solution in such a way it is understandable.

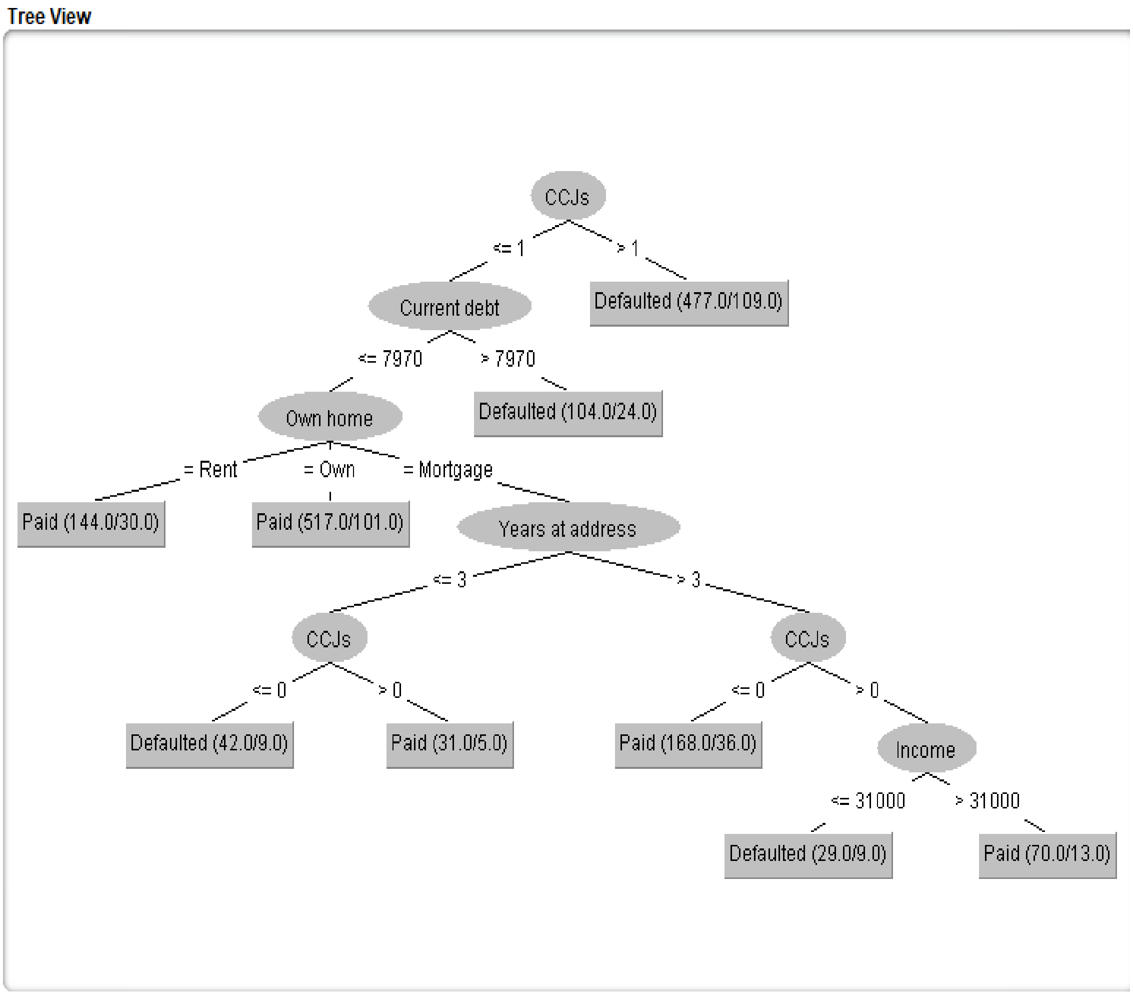
J48 hyper parameters: In J48 classifier few hyper parameters had significant effect on the model performance and the tree structure. With the basic **minNumObj** = 2 the structure of the tree was very complex, so various other values were tried and after minNumObj = 25 there was no significant changes in the tree, but the tree was simple and it showed the split based on the significant factors which we are looking for .

Confidence factor is a parameter to test the effectiveness of the post pruning in the J48 classifier model, it helps the tree in classifying the training set correctly. Default value in Weka is 0.25, this is changed with different values to check the changes in the model accuracy of J48 Classifier. Also the **Unpruned** option is J48 classifier is kept False. (Since we wanted the tree to be pruned). Confidence factor was changed with different values above and below 0.25 (like 0.30,0.5,0.1,0.05), but after all the options 0.27 gave us the better model accuracy and precision compared to other values with more than 500 defaulters predicted correctly. In fact few values did not make any changes to the model accuracy or the structure of the tree either. Also the **UseMDLcorrection** hyper parameter is Changed to True since we want a balance tree. The developed model and the tree structure are shown below with the hypermeters chosen,

J48 classifier model:

minNumObj = 25, Confidence Factor = 0.27, Unpruned = False, UseMDLcorrection = True, subtree raising = True,



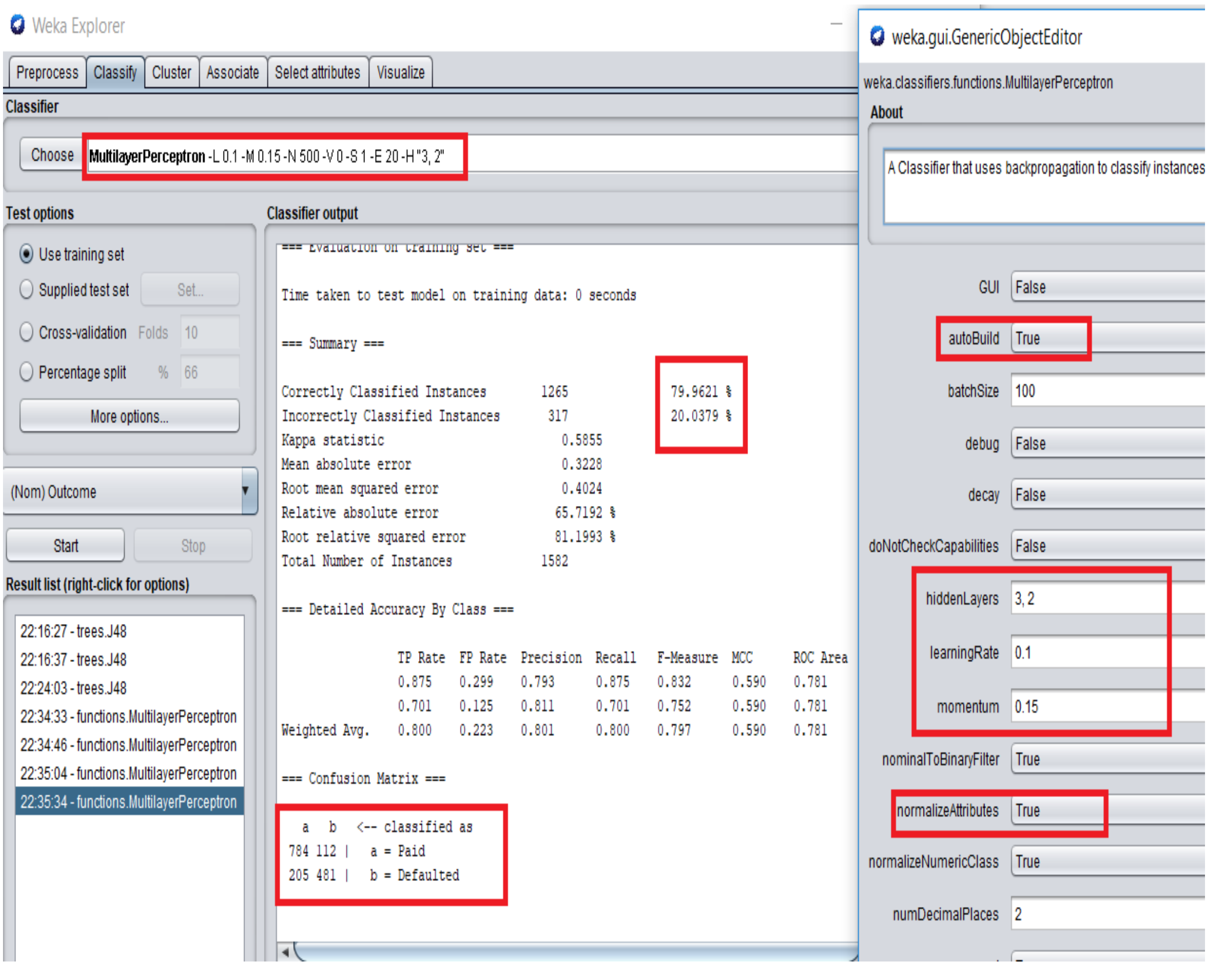


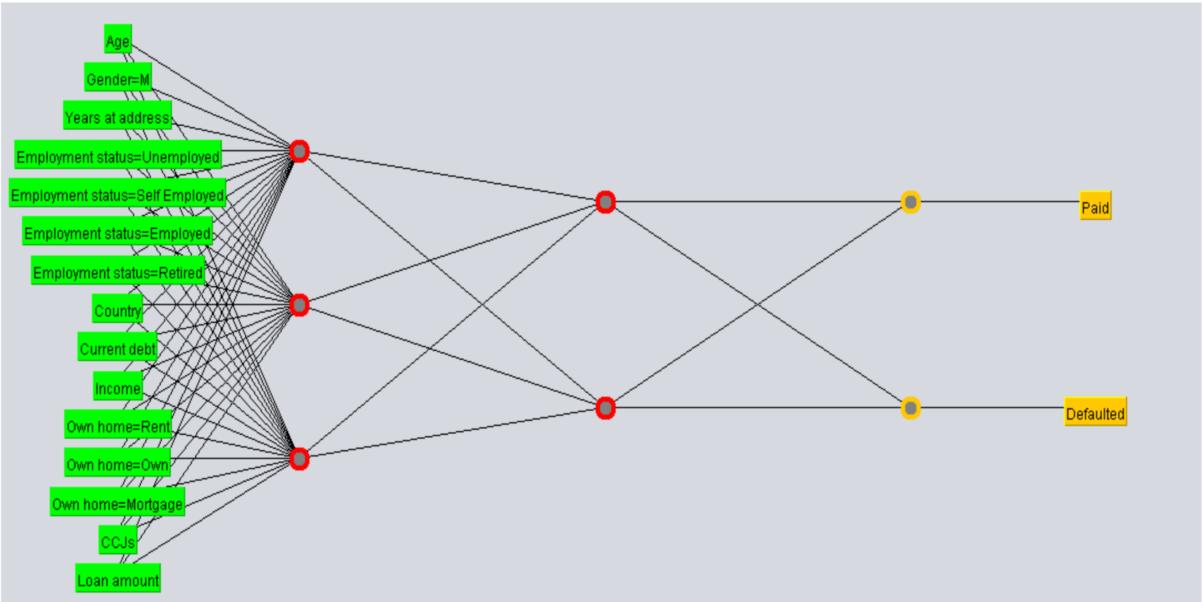
MLP Hyper parameters:

Hidden layers is a very important hyper parameter of MLP model , initially the model performed with default value (a) , but different combinations of the hidden layers with number of perceptron values like (2 , 3, (2,1), (3,3) ,etc) were changed and the model efficiency was compared every time based on the precision and the defaulters prediction, out of which (3,2) , **Learning rate** was changed with different values like(0.35,0.4,0.25 etc) but learning rate = 0.1 and **momentum** = 0.15 had better performance rate .**Normalize attributes** is kept True.

The developed model and the network structure are shown below with the hypermeters chosen,

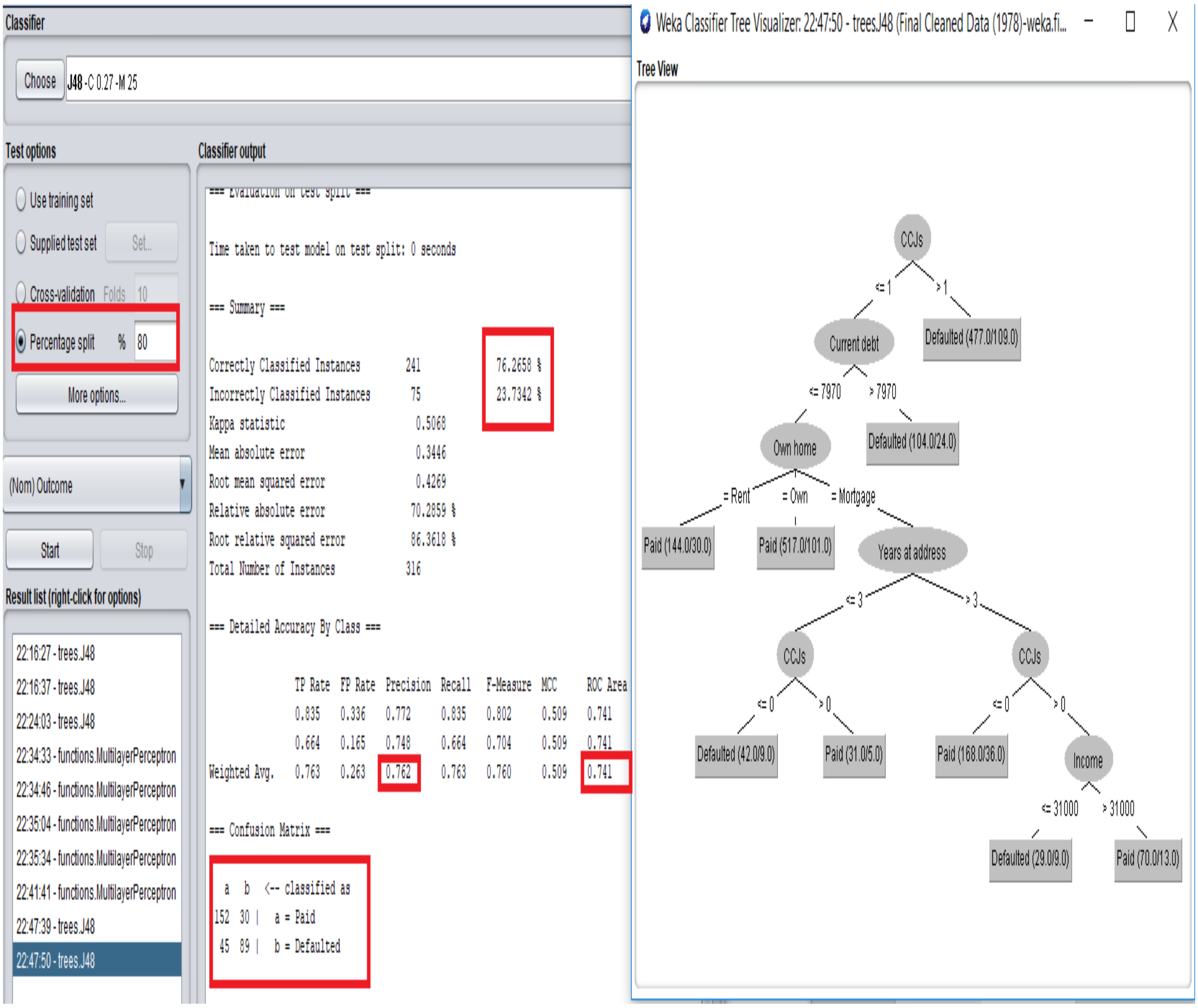
MLP model: Hidden Layers = (3,2) , Leaning Rate = 0.1 , Momentum = 0.15 , Normalize attributes = True , Auto Build = True , GUI = True


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 1 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
22:16:27 - trees.J48
22:16:37 - trees.J48
22:24:03 - trees.J48
22:34:33 - functions.MultilayerPerceptron
22:34:46 - functions.MultilayerPerceptron
22:35:04 - functions.MultilayerPerceptron
22:35:34 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on training data: 0 seconds
=== Summary ===
Correctly Classified Instances 1265 79.9621 %
Incorrectly Classified Instances 317 20.0379 %
Kappa statistic 0.5855
Mean absolute error 0.3228
Root mean squared error 0.4024
Relative absolute error 65.7192 %
Root relative squared error 81.1993 %
Total Number of Instances 1582
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area
0.875 0.299 0.793 0.875 0.832 0.590 0.781
0.701 0.125 0.811 0.701 0.752 0.590 0.781
Weighted Avg. 0.800 0.223 0.801 0.800 0.797 0.590 0.781
=== Confusion Matrix ===
a b <-- classified as
784 112 | a = Paid
205 481 | b = Defaulted
weka.gui.GenericObjectEditor
weka.classifiers.functions.MultilayerPerceptron
About
A Classifier that uses backpropagation to classify instances
GUI False
autoBuild True
batchSize 100
debug False
decay False
doNotCheckCapabilities False
hiddenLayers 3, 2
learningRate 0.1
momentum 0.15
nominalToBinaryFilter True
normalizeAttributes True
normalizeNumericClass True
numDecimalPlaces 2


Age
Gender=M
Years at address
Employment status=Unemployed
Employment status=Self Employed
Employment status=Employed
Employment status=Retired
Country
Current debt
Income
Own home=Rent
Own home=Own
Own home=Mortgage
CCJs
Loan amount
Paid
Defaulted

Now the hyper parameters are chosen for both the models, now they must be evaluated with the % split and then cross validated.

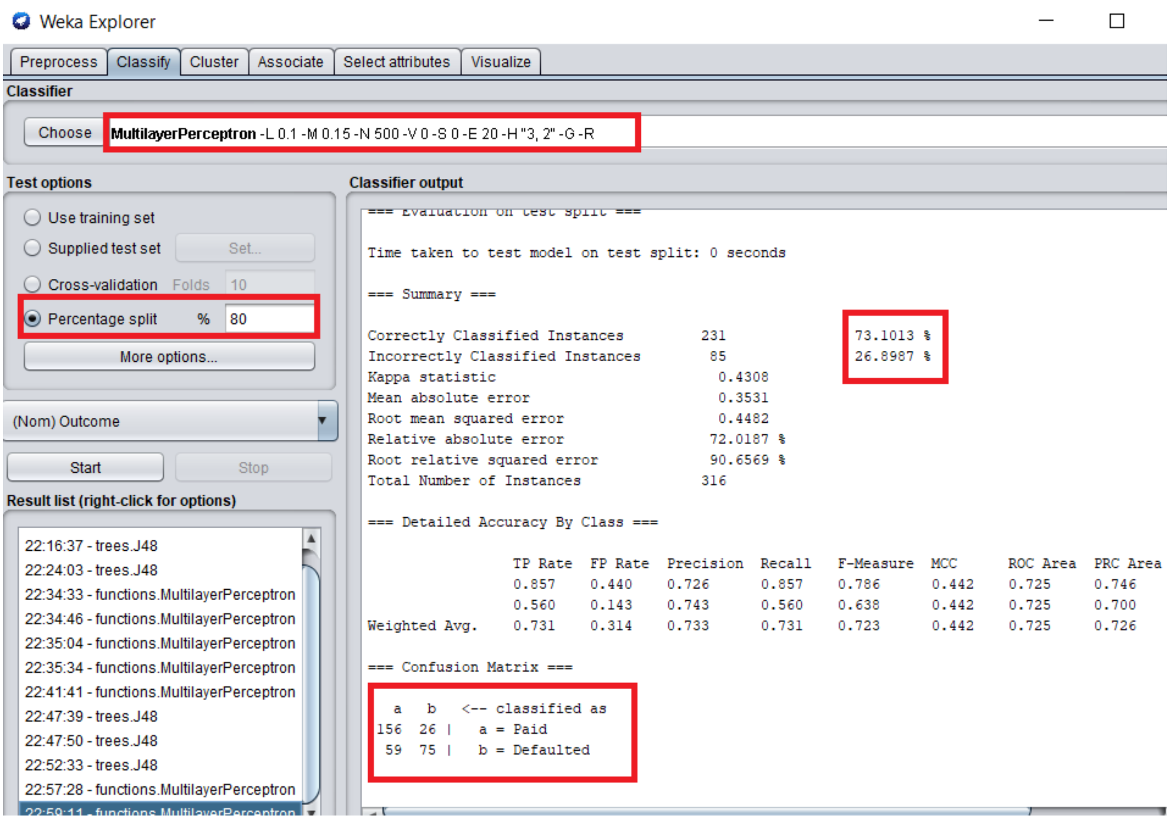
Percentage Split: In this test option of Weka, the supplied training set is split based on the percentage specified ,80 % split gave better results compared to other values. So, 80% of data in training set is used to train the model and the same is tested with the other 20% of the remaining training set and the result is recorded as shown below,



It is a good approach to evaluate the model with random data feed into the model using the seed option in Weka, so the same model is processed with seed values 1 to 10 and the results are recorded, from which the mean accuracy of the model is calculated as shown below,

J48	seed	Accuracy (Xi)	Xi-Mean	(Xi-Mean)^2	Precision	ROC Area	Paid	Defaulted
% split = 80	1	76.26	0.063	0.0040	0.76	0.74	152	89
Minnumobj = 20	2	78.16	1.963	3.8534	0.78	0.75	144	103
Confidence factor = 0.27	3	76.26	0.063	0.0040	0.76	0.79	158	83
Unpruned = False	4	74.68	-1.517	2.3013	0.74	0.73	132	104
Before Feature Selection	5	78.16	1.963	3.8534	0.78	0.76	151	96
	6	79.11	2.913	8.4856	0.79	0.77	150	100
	7	75.63	-0.567	0.3215	0.75	0.78	139	100
	8	75.94	-0.257	0.0660	0.75	0.77	144	96
	9	75.31	-0.887	0.7868	0.75	0.76	147	91
	10	72.46	-3.737	13.9652	0.72	0.74	147	82
	Sum	761.97		33.64	7.58	7.59	1464.00	944.00
	Mean	76.20			0.758	0.759	146	94
	Variance	3.74						
	SD	1.93						

From the above we infer that the mean accuracy of the J48 Classifier model is 76% with the % split (Holdout method), also the model variance is 3.74. Now the same procedure is carried out with MLP model.



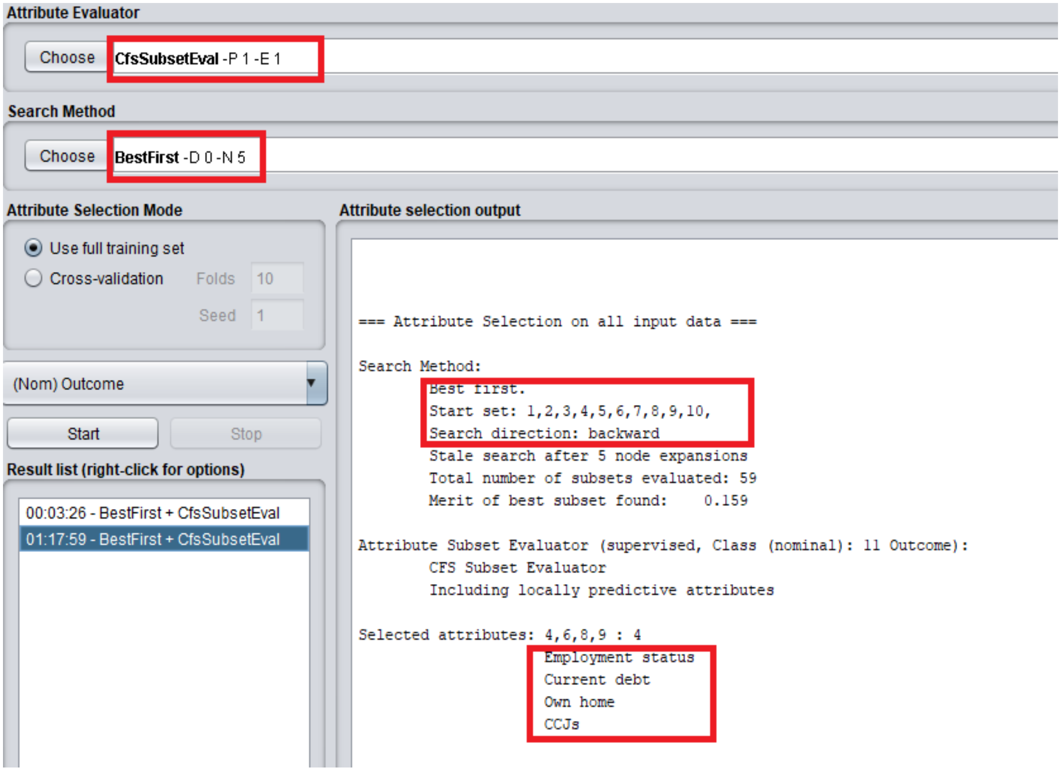
MLP	seed	Accuracy (Xi)	Xi-Mean	(Xi-Mean)^2	Precision	ROC Area	Paid	Defaulted
Hidden layers = (3,2)	1	73.1	-2.37	5.6122	0.73	0.72	156	75
Learning rate = 0.1	2	76.26	0.79	0.6257	0.77	0.77	150	91
Momentum = 0.15	3	76.26	0.79	0.6257	0.76	0.75	160	81
Before Feature Selection	4	75.94	0.47	0.2218	0.76	0.75	140	100
Hold out method	5	77.53	2.06	4.2477	0.77	0.77	150	95
	6	76.89	1.42	2.0192	0.76	0.76	151	92
	7	76.26	0.79	0.6257	0.76	0.77	151	90
	8	72.78	-2.69	7.2307	0.73	0.74	155	75
	9	73.41	-2.06	4.2395	0.73	0.74	148	84
	10	76.26	0.79	0.6257	0.76	0.76	156	85
	Sum	754.69		26.07389	7.53	7.53	1517	868
	Mean	75.47			0.753	0.753	152	87
	Variance	2.90						
	SD	1.70						

From the above we infer that the mean accuracy of the MLP model is 75% with the % split (Holdout method), also the model variance is 2.90.

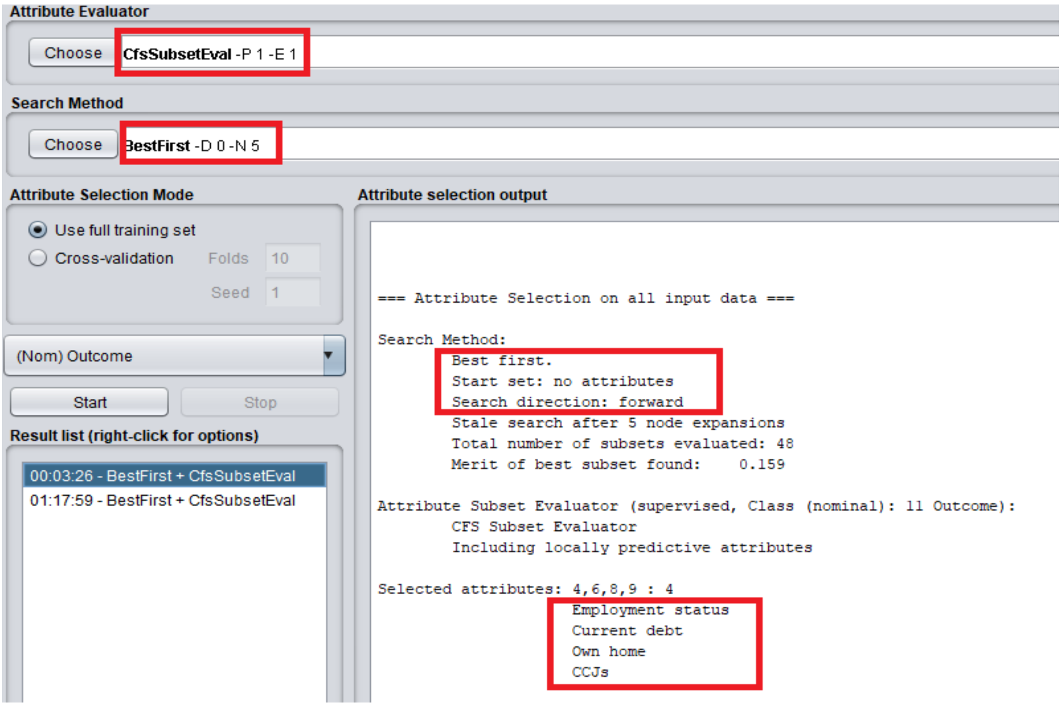
Feature Selection: As we know 11 attributes from the data set are used in this customer classification task, but there should always be some significant attributes that are chosen by the models to classify. It is important to understand that at this point of time to know the best fit model with only those significant attributes in both the developed models (J48, MLP).Also from the bank's perspective , it is important to take a decision(Loan Sanction) based on the most significant attributes, which is very important and helps the business and banks to better estimate the customers based on those specific attributes.Weka is powerful enough to make the work easy with its own features, Attributes are selected based of their significant relationship with the target variable.

CfsSubsetEval: Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. And with the help of a search method –**Best first** Weka shows the selected features. We have used both **Backward and Forward** approach to find the selected features as shown below,

Feature Selection: Backward approach, Best First,

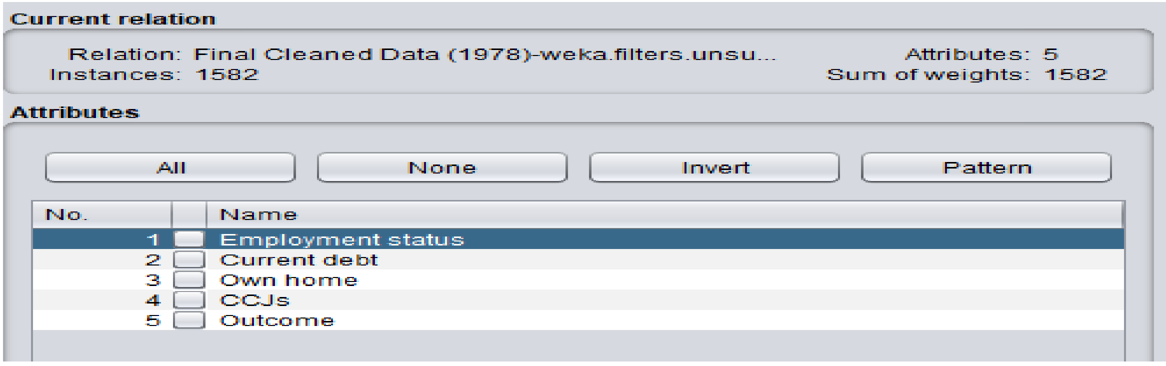


Feature Selection: Forward Approach, Best first

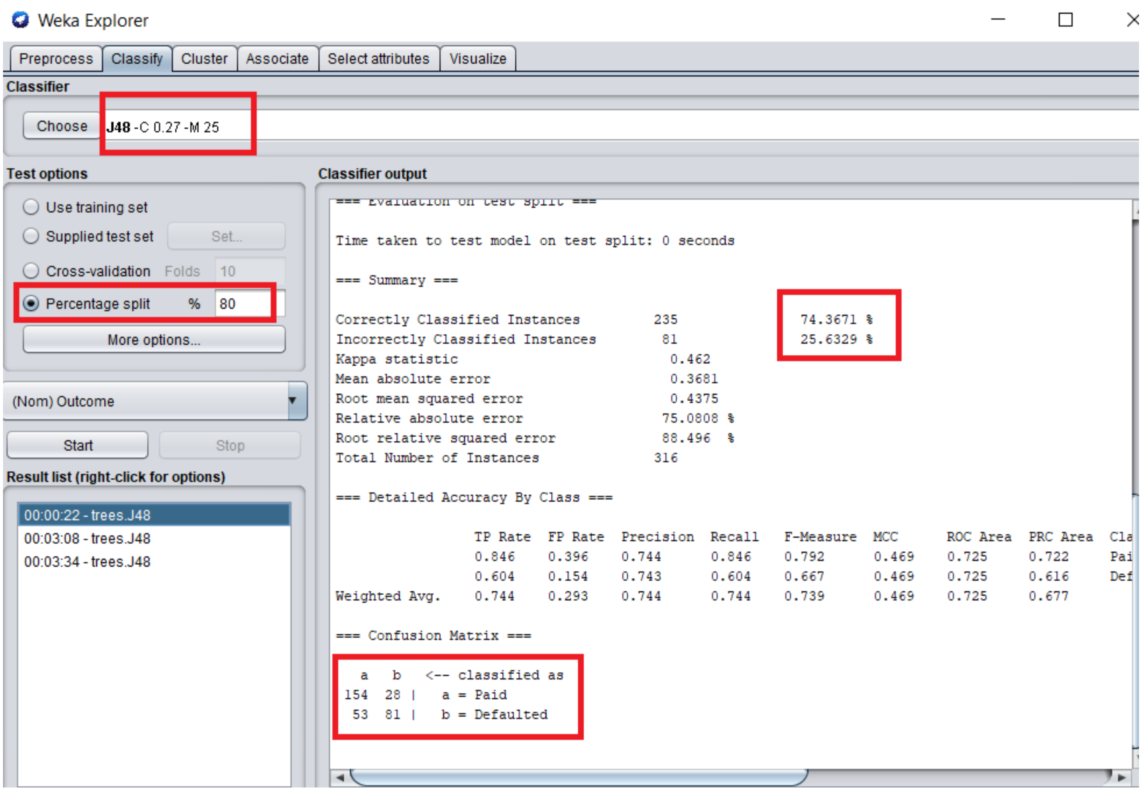


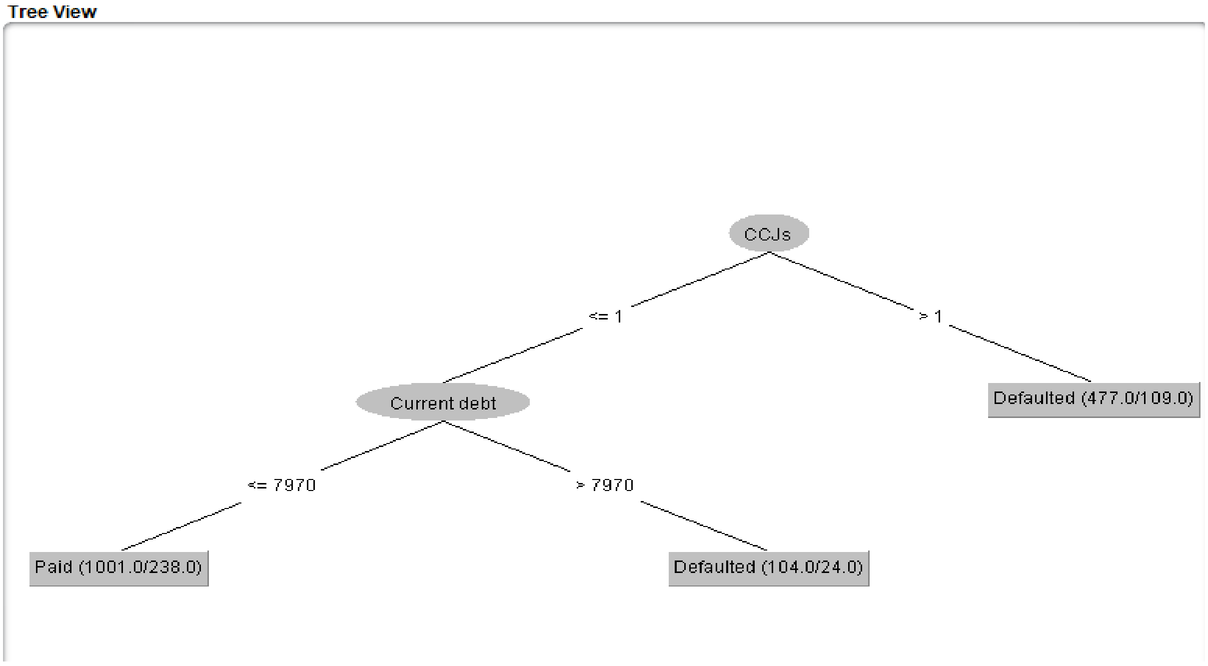
From the above feature selection approach we can see that Weka has chosen Four main features out of 11 , they are **CCJ, Current Debt ,Own Home , Employment status** – we can relate now to the OneR rule stated earlier , **CCJ is the main feature to classify the customers.**

AttributeSelectedClassifier is another way to cross check the selected attributes corresponding to the chosen algorithms, after checking specific to the models we could confirm that those fours features are invariably the same for all the models, this model specific feature selections screenshots are attached in the appendices for reference. Other features apart from the selected four can be removed from the data set and the models are tested again,



Now after the feature selection both the models need to be revaluated to see if there is any improvement in the model efficiency.



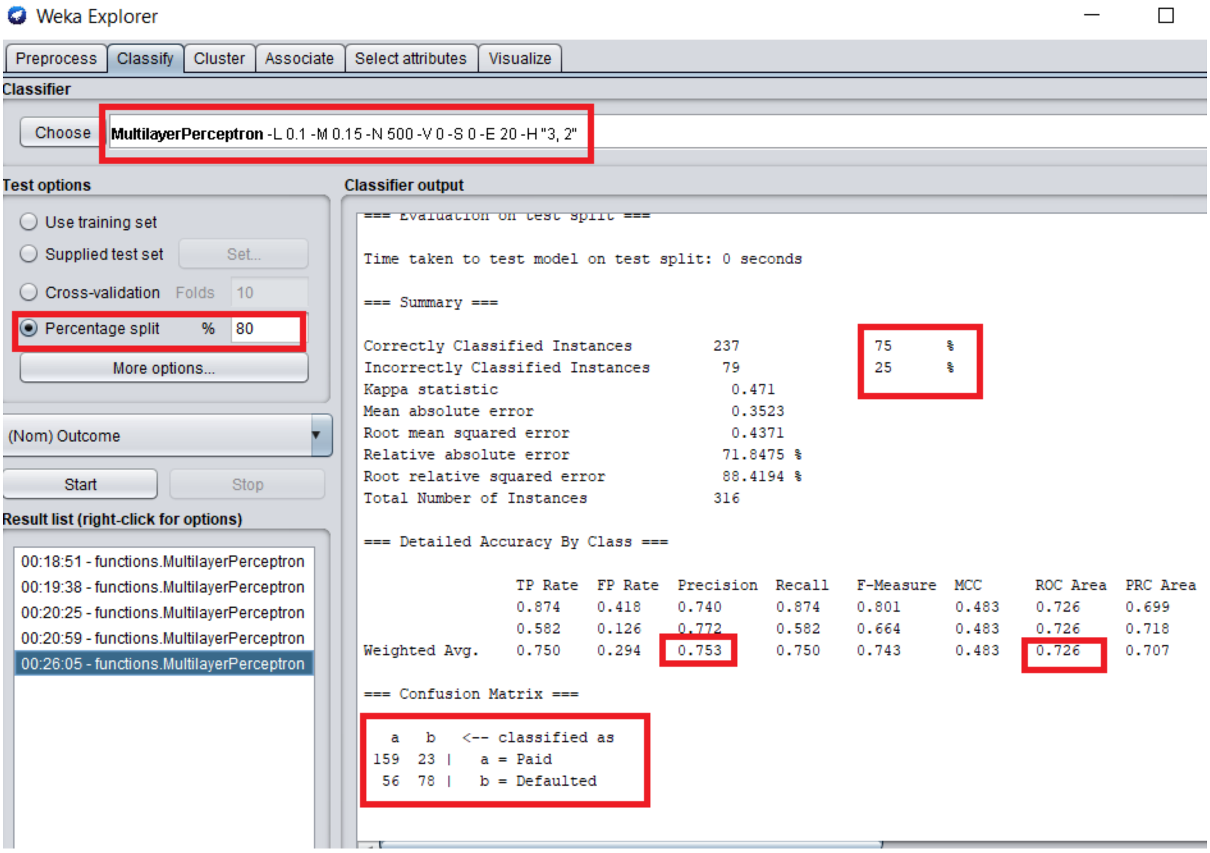


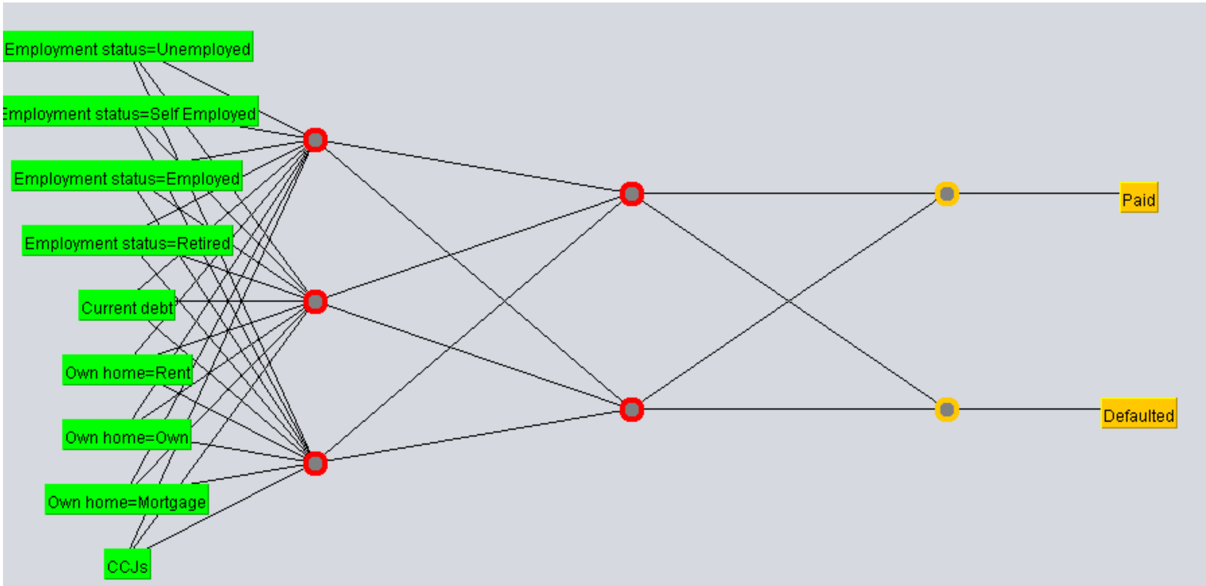
From the above structure of the J48 tree we can see that the model has chosen the most two significant attributes in predicting the defaulters, this is after the feature selection.

J48	seed	Accuracy (Xi)	Xi-Mean	(Xi-Mean)^2	Precision	ROC Area	Paid	Defaulted
% split = 80	1	74.36	-2.31	5.3361	0.74	0.72	154	81
Minnumobj = 20	2	76.58	-0.09	0.0081	0.77	0.70	147	95
Confidence factor = 0.25	3	77.84	1.17	1.3689	0.78	0.79	168	84
Unpruned = False	4	75.94	-0.73	0.5329	0.76	0.75	141	99
After Feature Selection	5	78.16	1.49	2.2201	0.78	0.76	151	96
	6	75.94	-0.73	0.5329	0.76	0.74	143	88
	7	77.84	1.17	1.3689	0.77	0.76	151	95
	8	77.84	1.17	1.3689	0.78	0.76	140	95
	9	76.89	0.22	0.0484	0.76	0.73	152	85
	10	75.31	-1.36	1.8496	0.75	0.74	161	87
	Sum	766.7		14.63	7.65	7.45	1508.00	938.90
	Mean	77			0.77	0.75	151	94
	Variance	1.63						
	SD	1.28						

From the above table we can see that the mean accuracy of the J48 model after feature is 77% and 94 mean defaulters prediction. Model efficiency has been improved after feature selection.

Carrying out the same procedure with MLP after feature selection and the results are recorded as shown in the below section,





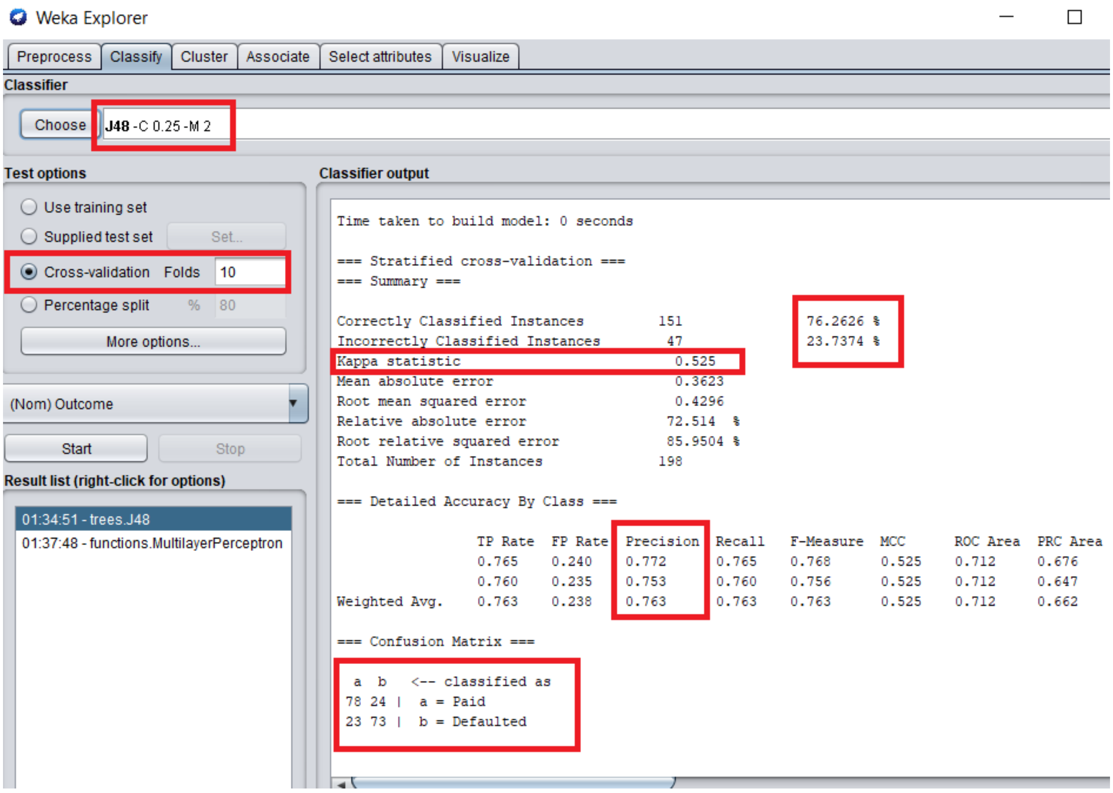
MLP	seed	Accuracy (Xi)	Xi-Mean	(Xi-Mean)^2	Precision	ROC Area	Paid	Defaulted
Hidden layers = (3,2)	1	75	-0.502	0.2520	0.75	0.73	159	78
Learning rate = 0.1	2	75.63	0.128	0.0164	0.76	0.76	149	90
Momentum = 0.15	3	75.94	0.438	0.1918	0.75	0.75	161	79
After Feature Selection	4	74.05	-1.452	2.1083	0.76	0.75	149	85
Hold out method	5	77.84	2.338	5.4662	0.78	0.78	155	91
	6	77.21	1.708	2.9173	0.78	0.78	159	85
	7	75.94	0.438	0.1918	0.76	0.76	138	79
	8	73.41	-2.092	4.3765	0.74	0.77	157	78
	9	75.00	-0.502	0.2520	0.75	0.74	157	86
	10	75	-0.502	0.2520	0.75	0.75	257	80
	Sum	755.02		16.0244	7.5800	7.5700	1641	889
	Mean	75.502			0.76	0.76	164	89
	Variance	1.78						
	SD	1.33						

From the above table we can see that the mean accuracy of the MLP model after feature selection is 75% and 89 mean defaulters prediction. Model efficiency has been improved after feature selection.

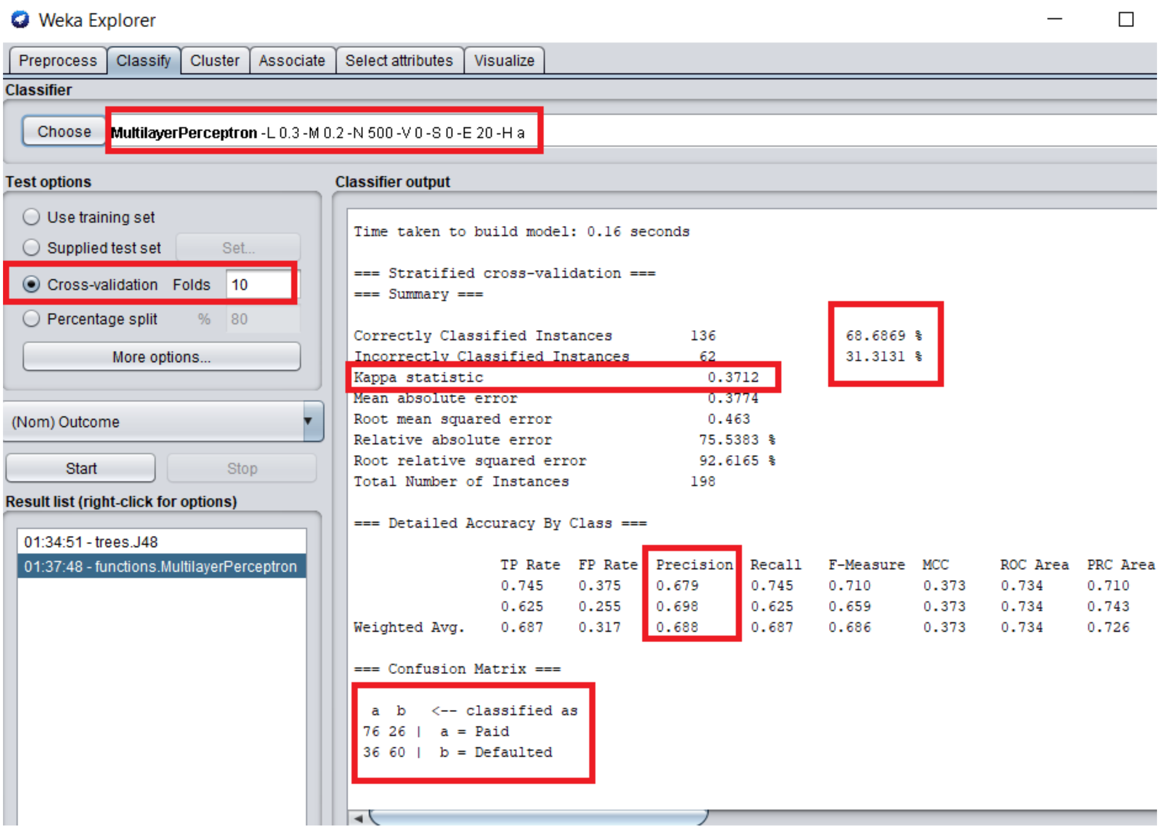
Also J48 classifier model has outperformed MLP in the above cases.

Now both these model needs to be cross validated with the selected features to arrive into the best fit model out of the two**. K-fold cross validation on the model should be evaluated with the initially created cross validation set (198 records).** Only those selected features are again used in the cross validation set for validation. For cross validation k=10 chosen, So both the models are evaluated with 10-fold cross validation and the results are compared ad shown in the below section,

J48 classifier model with 10-fold cross validation



MLP model with 10-fold cross validation



Summary table for 10-fold cross validation is shown below,

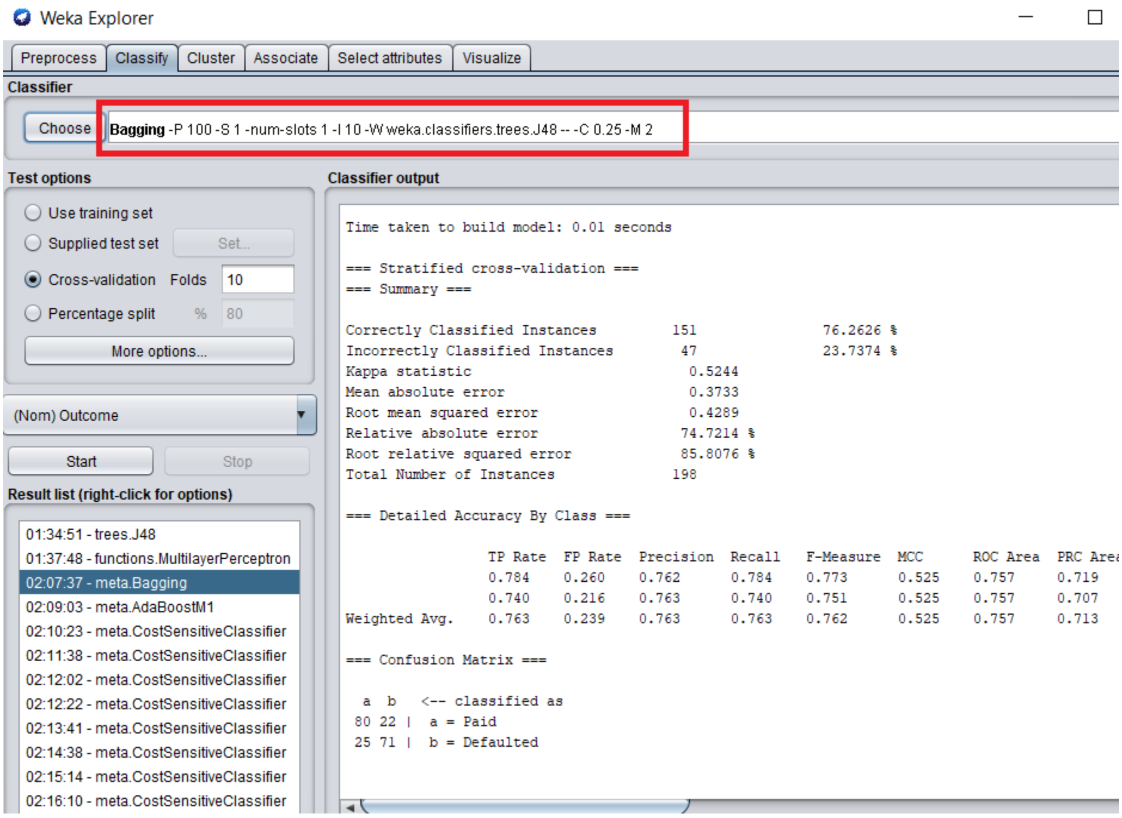
Models	Accuracy	Kappa Statistic	Precision	Paid (TP)	Defaulters (TN)	MSE
J48	76.26%	0.525	0.763	78	73	0.42
MLP	68.68%	0.3712	0.688	76	60	0.47

From the above summary it is evident that J48 classifier has outperformed the MLP model in all the cases with better accuracy, precision rate and kappa statistic and in classifying the customers with most significant attributes. From the tree structure obtained from J48 classifier after feature selection clearly shows that two attributes have the predicting rate. Mainly J48 classifier model has predicted maximum number of the Defaulters correctly, 73 times in 10-fold cross validation compared to MLP model, which is the main objective of the model we are looking for, So J48 stands out in better prediction of the Defaulters.

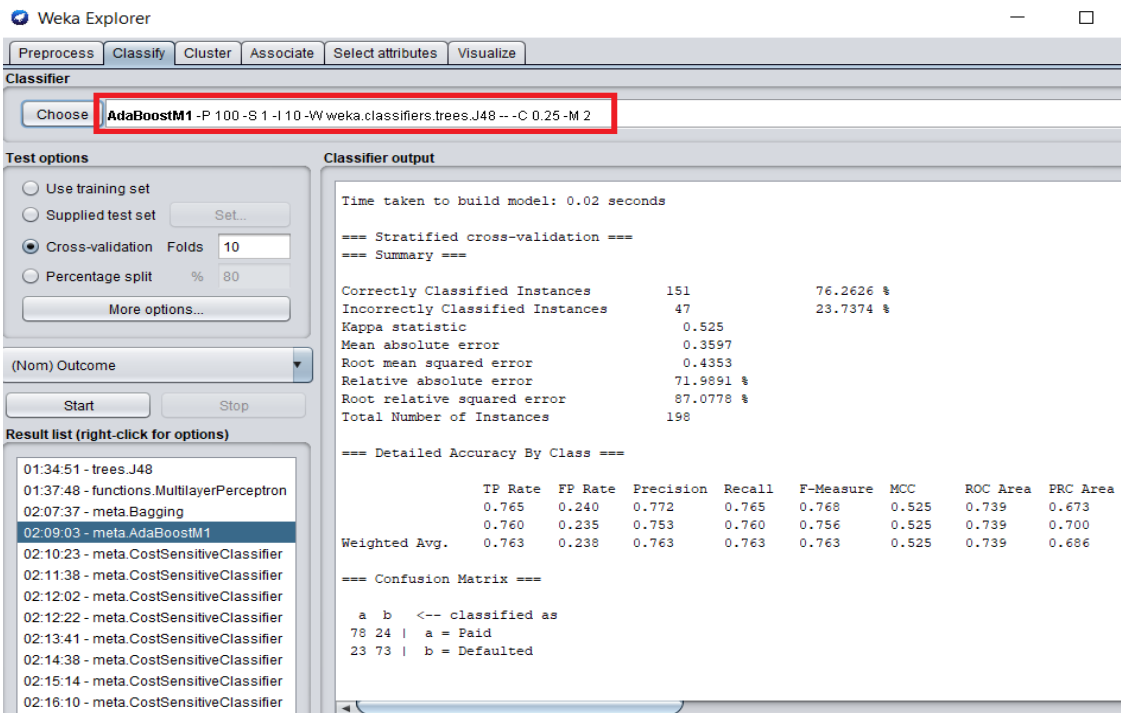
J48 is the best model to select, also we need to consider evaluating the model with ensemble methods and for cost sensitive methods for increasing the efficiency of J48 classifier in predicting the defaulters.

Ensemble Methods: It is also very important to use the ensemble methods to build models and compare their efficiencies with J48 . In the section below Bagging and AdaBoost ensemble methods are used in Weka to build models and compare. J48 classifier is evaluated with both the models and their results are captured as shown below,

Bagging ensemble method with J48 Classifier: (10-fold cross validation)



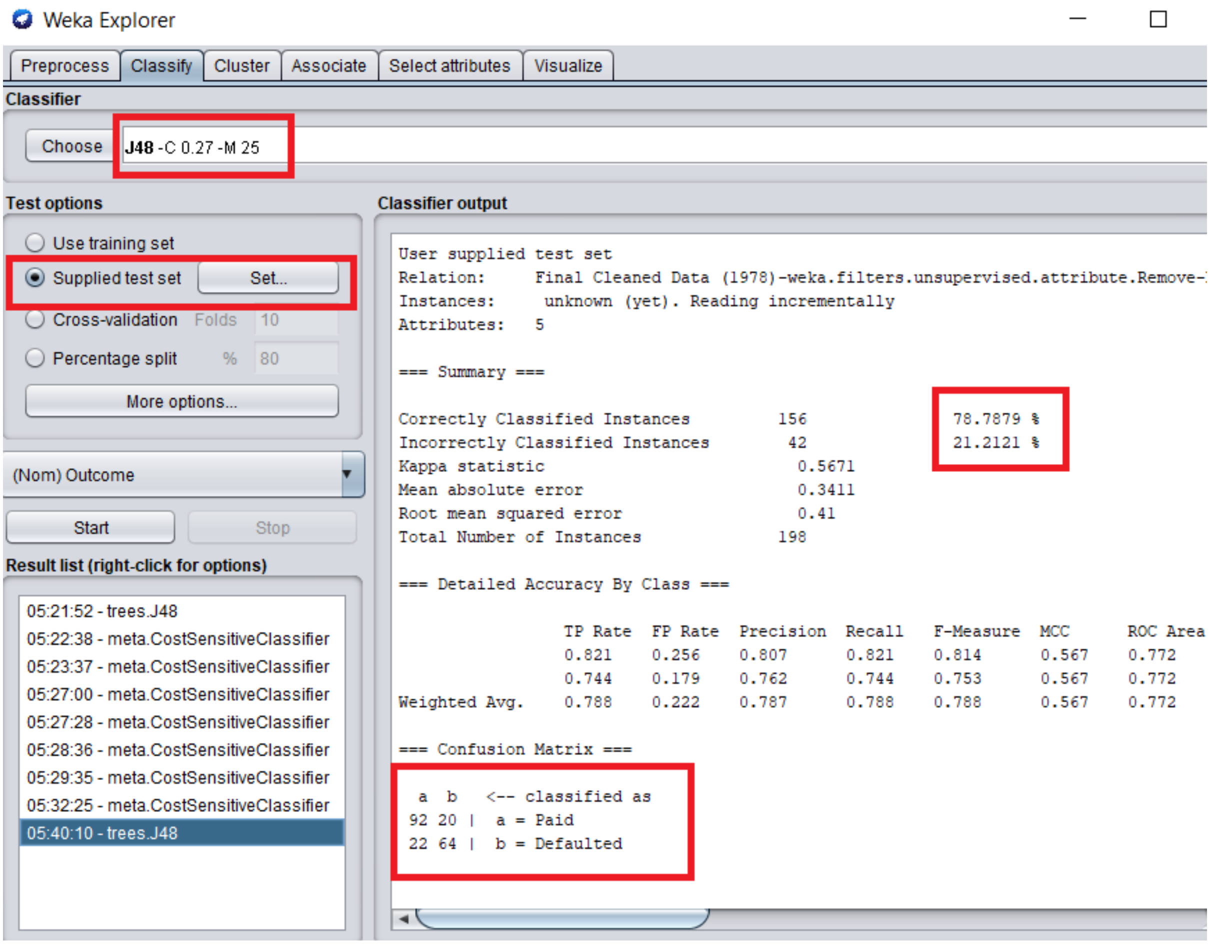
AdaboostM1 ensemble method with J48 Classifier: (10- fold cross validation)



From the above ensemble methods with J48 classifier model we can see that there is no significant changes in the accuracy, precision or Defaulter prediction after adding to ensemble method. So over all J48 classifier model has performed to the same level of accuracy in all the normal and ensemble methods.

Model Testing:

After all the evaluation and comparison between J48 and MLP models , we have chosen J48 classifier as the best fit model to predict the defaulters with most significant features. Now this model need to be tested with the created test set (198 records) to see how this model performs on the test set. On re-evaluating the J48 classifier model with 10-fold cross validated with the test set the below results are achieved,



We Could see that the model accuracy is 78.78%, Kappa statistic is 0.567, TN accuracy is 74%, with 20 wrongly classified Defaulters as Paid . As we know the model we expect should have maximum efficiency in predicting the defaulter correctly and to lower the wrong prediction of defaulters as paid.

We could see a little bias with the model in predicting the number of defaulters, this is due to the dataset instances we have. And the objective is to build a model that can predict the defaulters with maximum precision and accuracy. In order to help the model, predict the defaulters better the cost sensitive learning method is used with the J48 being the base classifier. This method helps the model to reduce the false prediction rate and improve its efficiency.

Cost sensitive Learning:

In this method the cost sensitive classifier has a default cost matrix which can be resized according to the classes of the target variable (in our case we have two outcomes so we need a 2X2 cost matrix. It has its own default weights which can be changed to get the better prediction rate of the Defaulters

Default Cost matrix (2x2)

0.0 1.0

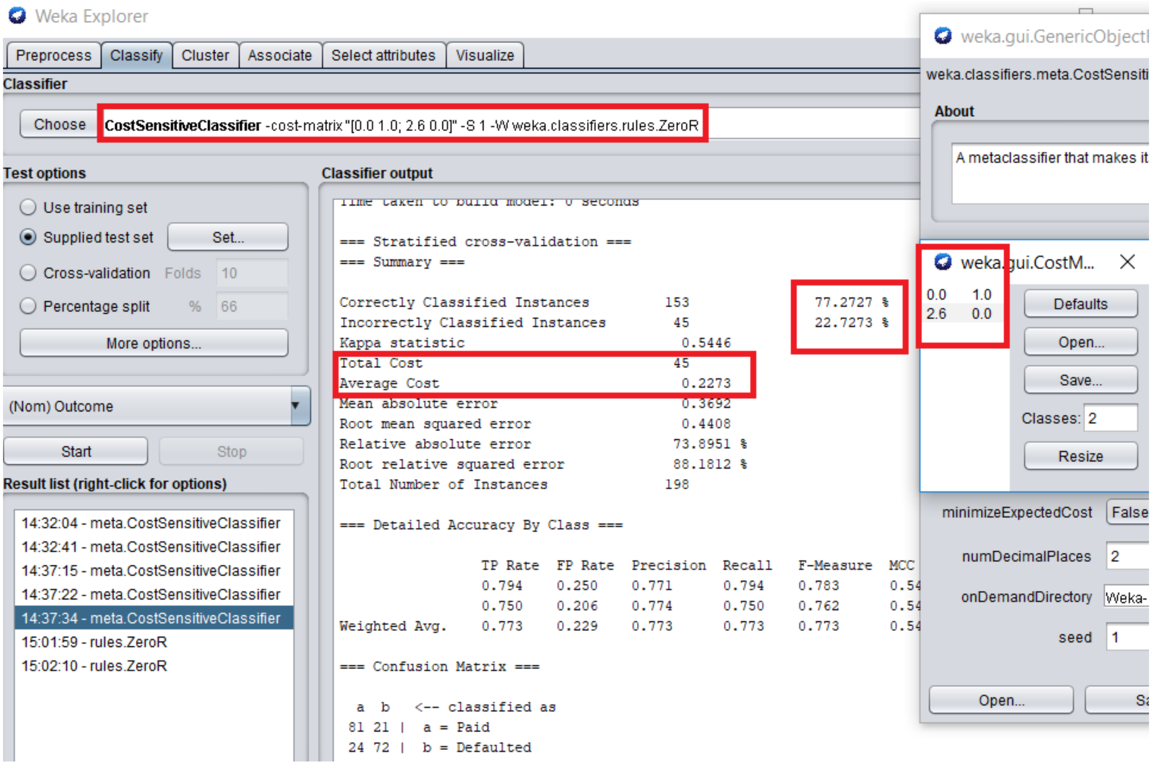
1.0 0.0

New Cost matrix (2x2)

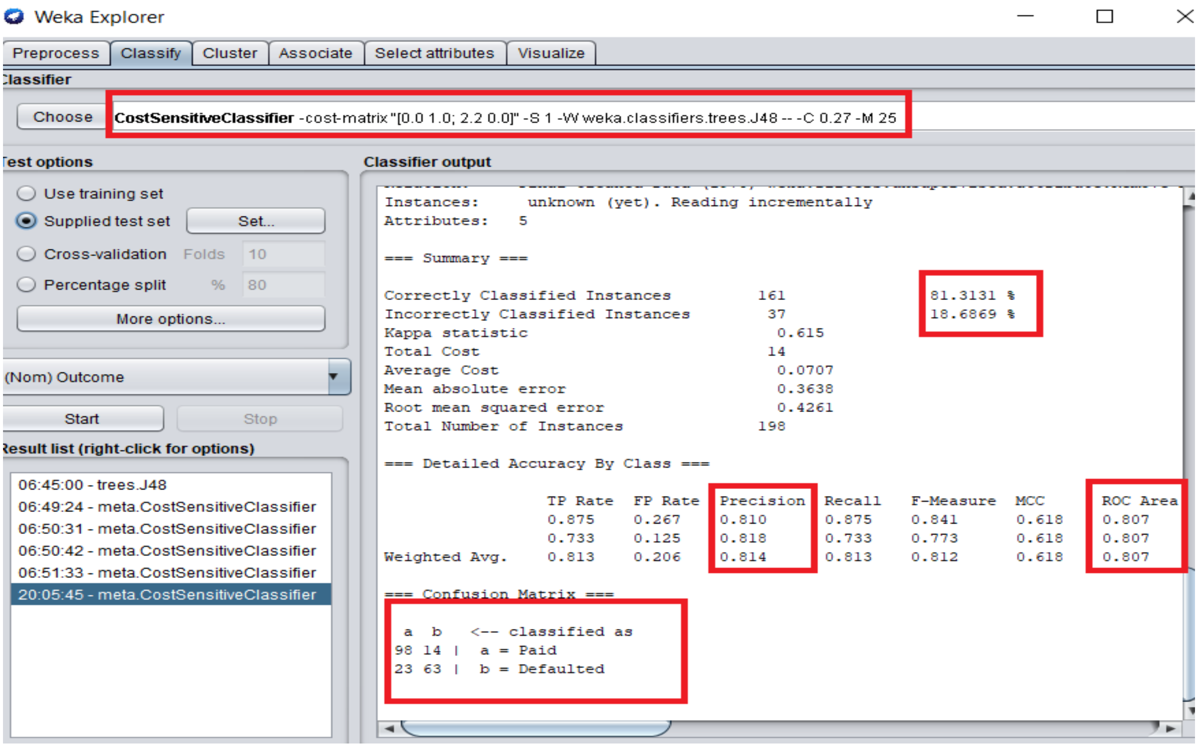
0.0 1.0

2.6 0.0

With the above weights the cost is recalculated and the results are shown below,



Above model has 77.27% accuracy with 77 % precision, cost = 45 and the average cost is 0.22. On Revaluating the above cost sensitive J48 Classifier model with the set we get the below results as shown in the screen shot below,



From the above we can see that the precision of correct predictions has increased with over all model efficiency. With calculated cost is 14 and average cost is 0.0707 which is calculated based on the weights added in the cost matrix of this model. Also the model accuracy is increased compared to the J48 classifier without the cost sensitive method evidently. The precision of the paid and defaulters prediction is high in the model with 81%, also the number of false predictions is reduced which is a good sign. Cost of the method is calculated in Weka and it is shown in the cost/benefit curve. Minimised cost is calculated based on the wrong predictions in the confusion matrix and the weights supplied to increase the model efficiency. From the Cost /Benefit curve of this model included in the below section shows that the wrong prediction rates are reduced evidently with higher prediction rate of the defaulted and paid customers. Cost of the model has reduced as well to 37, this cost calculation is shown below,

Cost Calculation:

Changed weight for the False positive in the cost matrix = 2.6

False positive value from the confusion matrix of the final model = 23

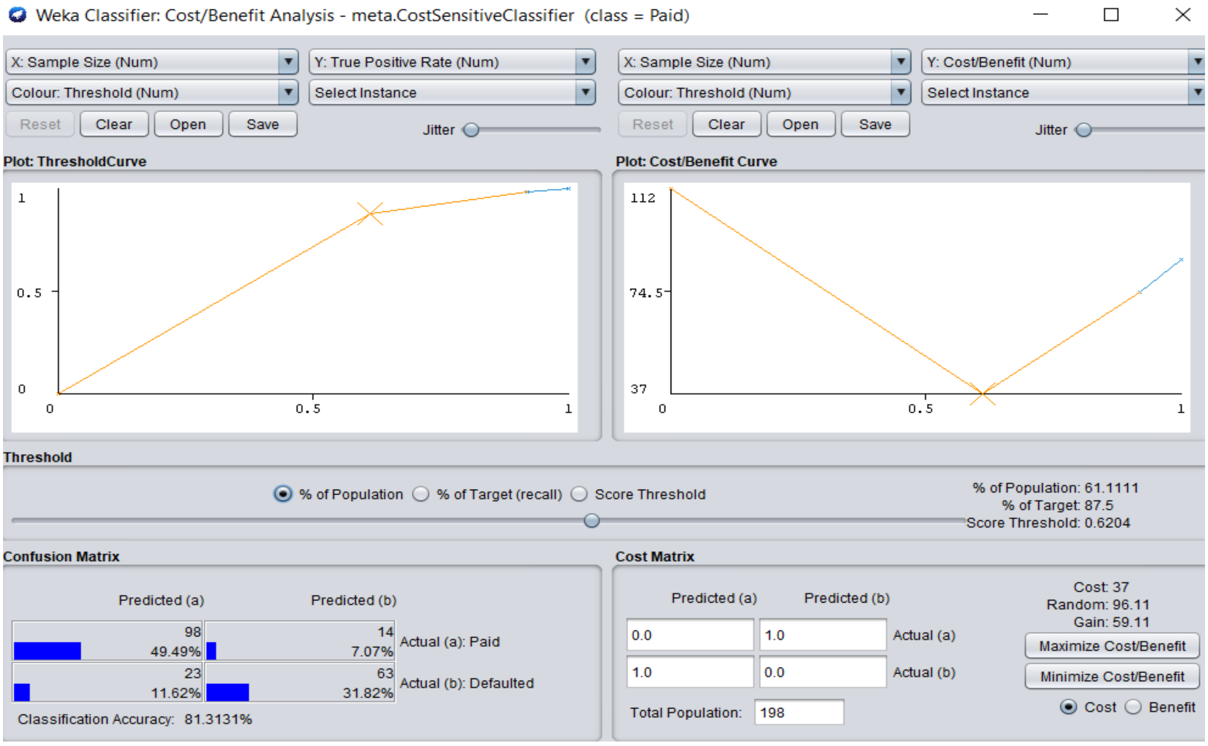
False negative value from the confusion matrix of the final model = 14

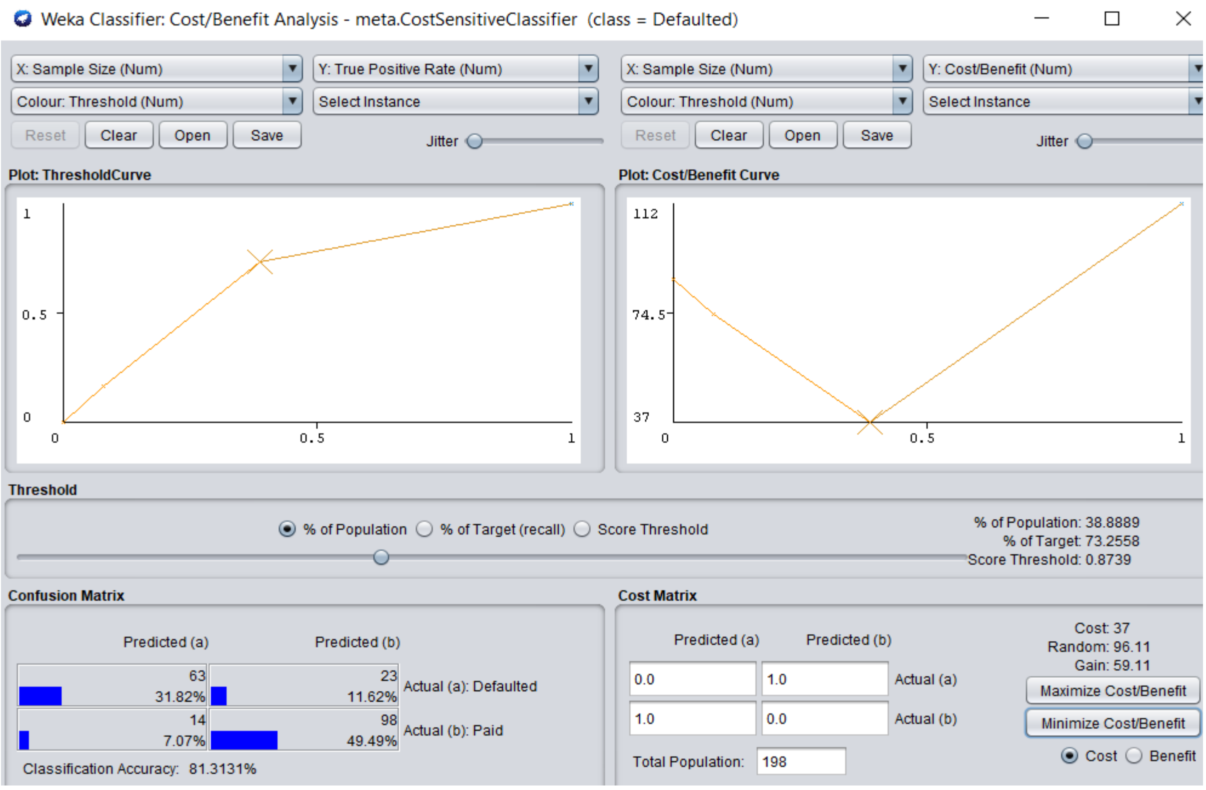
Population (instances) = 198

Cost of the model = ((2.6 * 23) + 14)/198 = 37

all the above values described, prediction rates of paid and defaulters with the cost are shown below,

Cost/Benefit curves:





Results and Errors:

Summary table to compare the J48 classifier model with and without the cost sensitive learning is shown below,

Models	Accuracy	Kappa Statistic	Precision	Paid (TP)	Defaulters (TN)	MSE
J48	78.78%	0.56	0.78	92	64	0.41
J48 (With Cost Sensitive Learning)	81.31%	0.615	0.81	98	63	0.41

From the above table we can see that J48 classifier model with the Cost sensitive method has improved the prediction accuracy, with a precision of 81% compared to 78% without the Cost sensitive learning method. Mainly J48 classifier with the cost sensitive learning has reduced the number wrongly classified instances which can be evidently visualised through the confusion matrix shown in below section. A good model should classify the customers correctly, i.e the Defaulter and Paid should be predicted correctly maximum extent by reducing the number wrong prediction. Cost sensitive classifier with J48 classifier as the base classifier and with the new cost matrix have increased the efficiency of the model with improved results with very less cost = 14 . Eat us have a look into the confusion matrix below for both the model and analyse,

Confusion matrix :

J48 Classifier model testing without Cost sensitive learning:

J48		**Actual**			
		P	**N**		
Predicted	**P**	**92**	**20**	**82%**	**Not Defaulter**
	N	**22**	**64**	**74%**	**Defaulter**
		19%	**24%**	**78%**	**Accuracy**
		Sensitivity	**Specificity**		

J48 Classifier model testing with Cost Sensitive learning:

J48		Actual			
		P	N		
Predicted	P	98	14	88%	Not Defaulter
	N	23	63	73%	Defaulter
		19%	18%	81%	Accuracy
		Sensitivity	Specificity		

From the above two confusion matrix we can compare the efficiency of J48 model with and without the cost sensitive learning. We can see that cost sensitive learning has increased the model accuracy in predicting the true values by reducing the false predictions over all. Significant difference in the False positive and False negative values in the confusion matrix, Cost method on J48 has reduced the FP and FN percentages and the expected true predictions have increased. Almost 88% of the Paid customers and 73% of the Defaulters are predicted correctly by the model with precision up to 80%. Also on the other hand the same model has reduced the wrong prediction rates to 19% and 18%. At this point we know the model has very good efficiency in predicting 80% of the customers correctly , but we should also have a look into those data points when the model failed to predict correctly .

Classifier errors of J48 (with cost sensitive method): Weka shows the classifier errors for each selected attribute contribution in the target prediction, for every single classifier model built. Classifier errors are captured and shown in the below screenshots, in which the predicted outcome values are plotted against the selected attributes and the errors are spotted. (Paid is denoted by **Blue Colour** and the Defaulted is denoted by **Red Colour**).

Classifier error and correct values representation in the plotted graph:

x **– Correctly predicted outcome (Paid)**

x **– Correctly predicted outcome (Defaulted)**

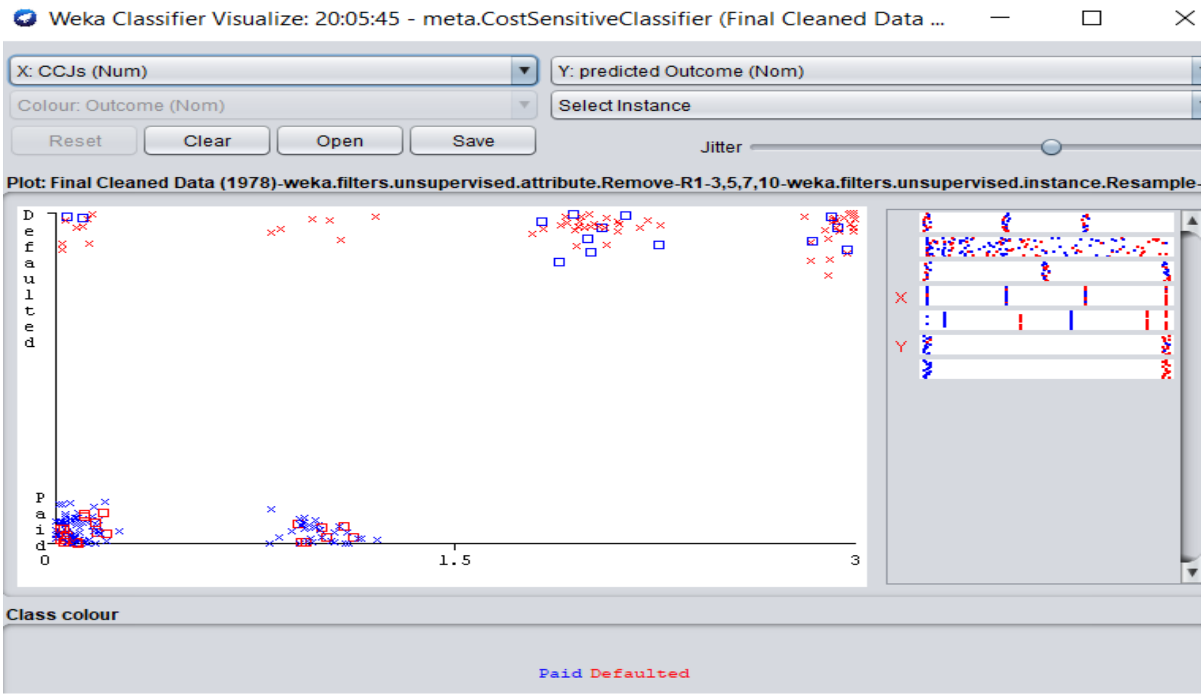
□ **- Wrongly predicted outcome(Paid)**

□ **- Wrongly predicted outcome(Defaulted)**

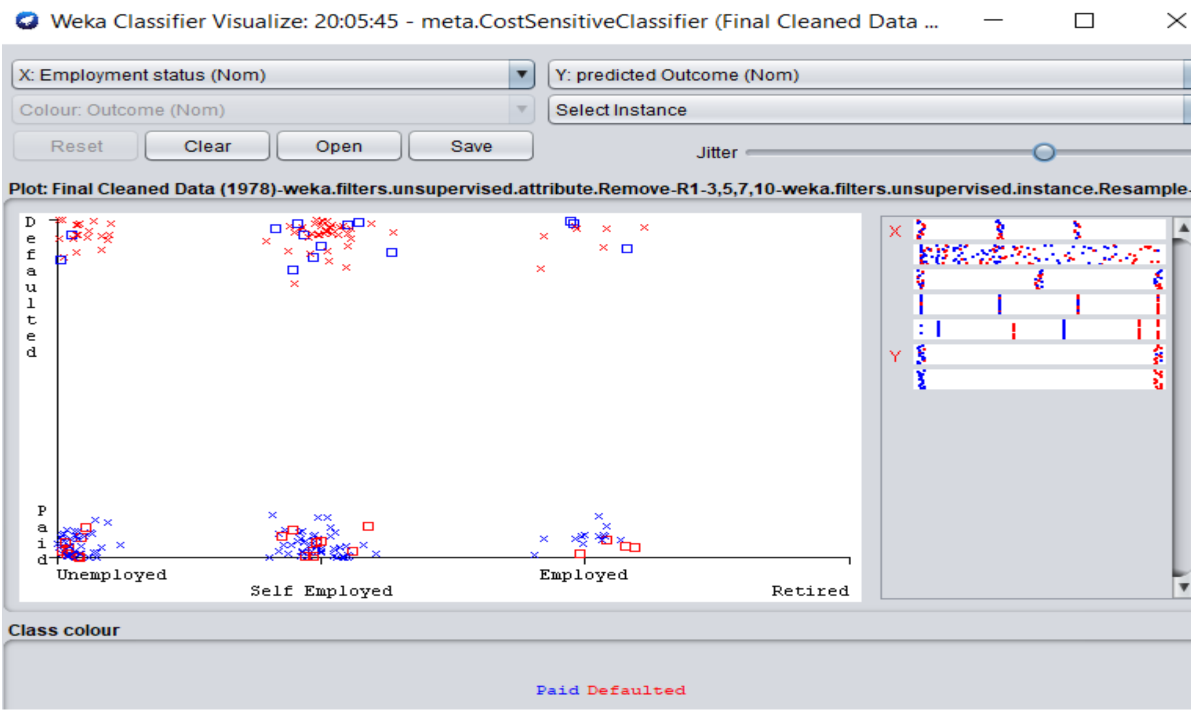
As we know from the confusion matrix of J48 model with Cost sensitive classifier (test), **14 customers are wrongly predicted as Defaulted when they are actually not , in the**

same way 23 customers are wrongly predicted as Paid when they actually defaulted , But the model has managed to correctly predict up to 81% .These are clearly shown in the plotted classifier error graphs in Weka as shown in the screen shots below,

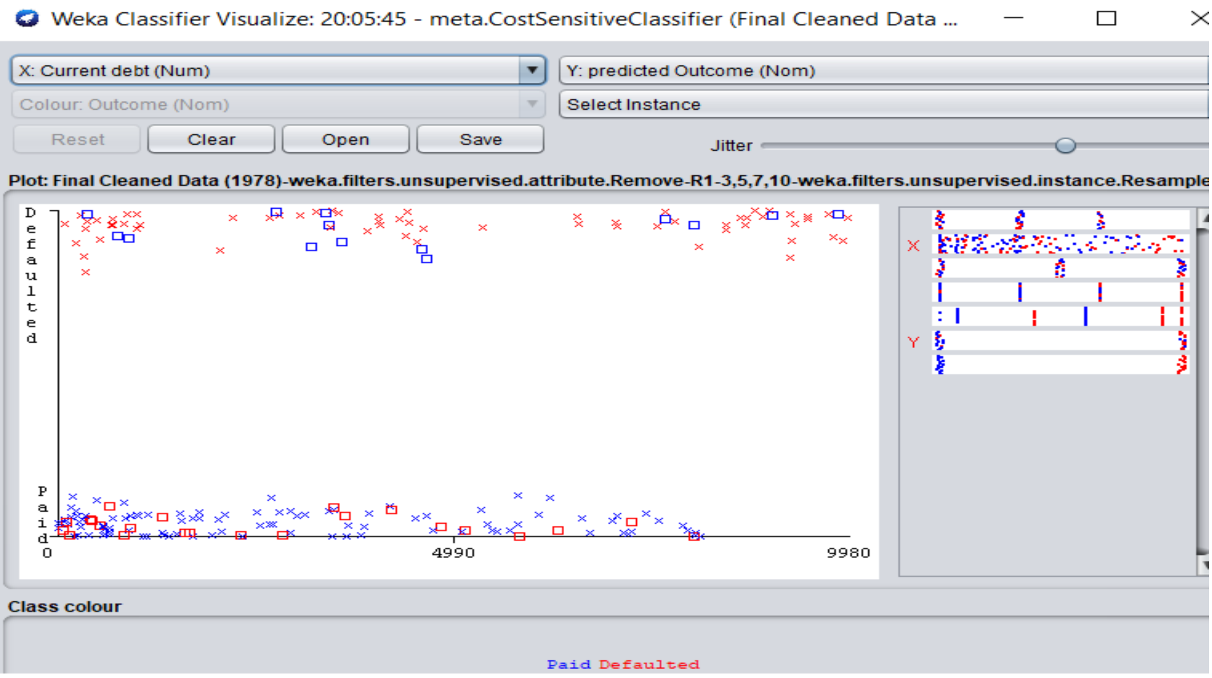
Predicted Outcome against CCJ,(With spotted 14 wrongly classified data points as Defaulted and 23 data points as Paid)



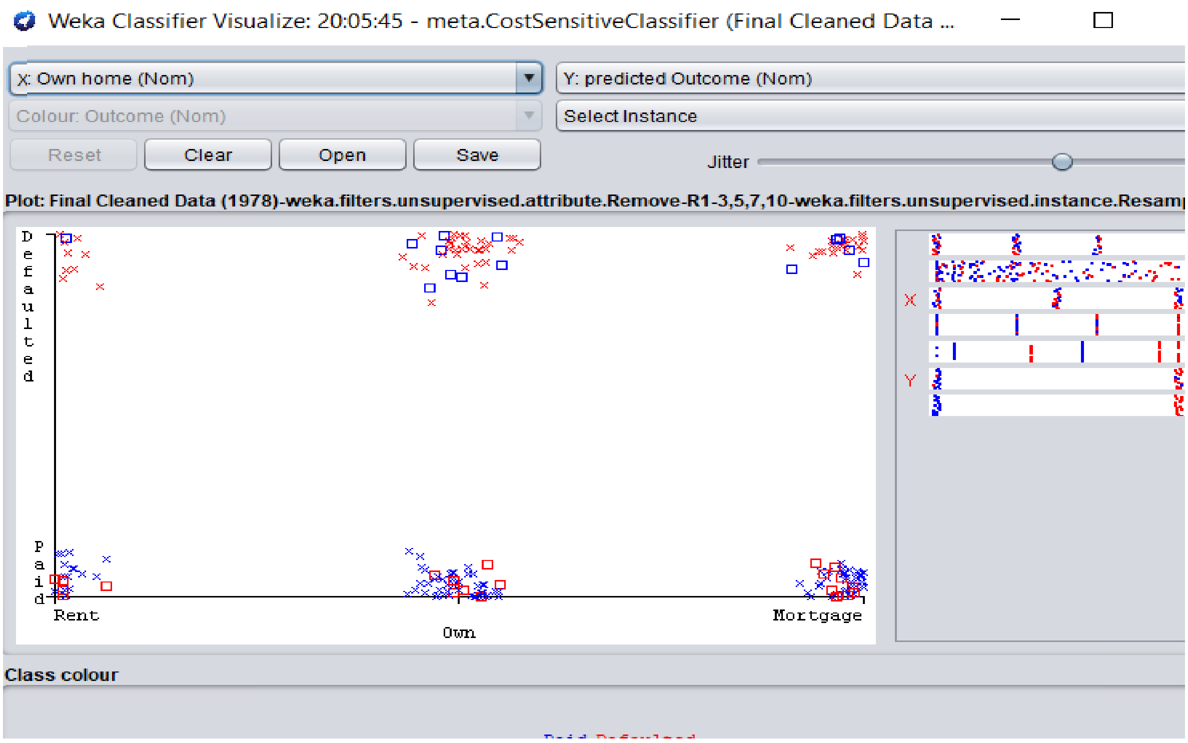
Predicted Outcome against Employment Status,(With spotted 14 wrongly classified data points as Defaulted and 23 data points as Paid)



Predicted Outcome against Current Debt,(With spotted 14 wrongly classified data points as Defaulted and 23 data points as Paid)



Predicted Outcome against Own Home,(With spotted 14 wrongly classified data points as Defaulted and 23 data points as Paid)

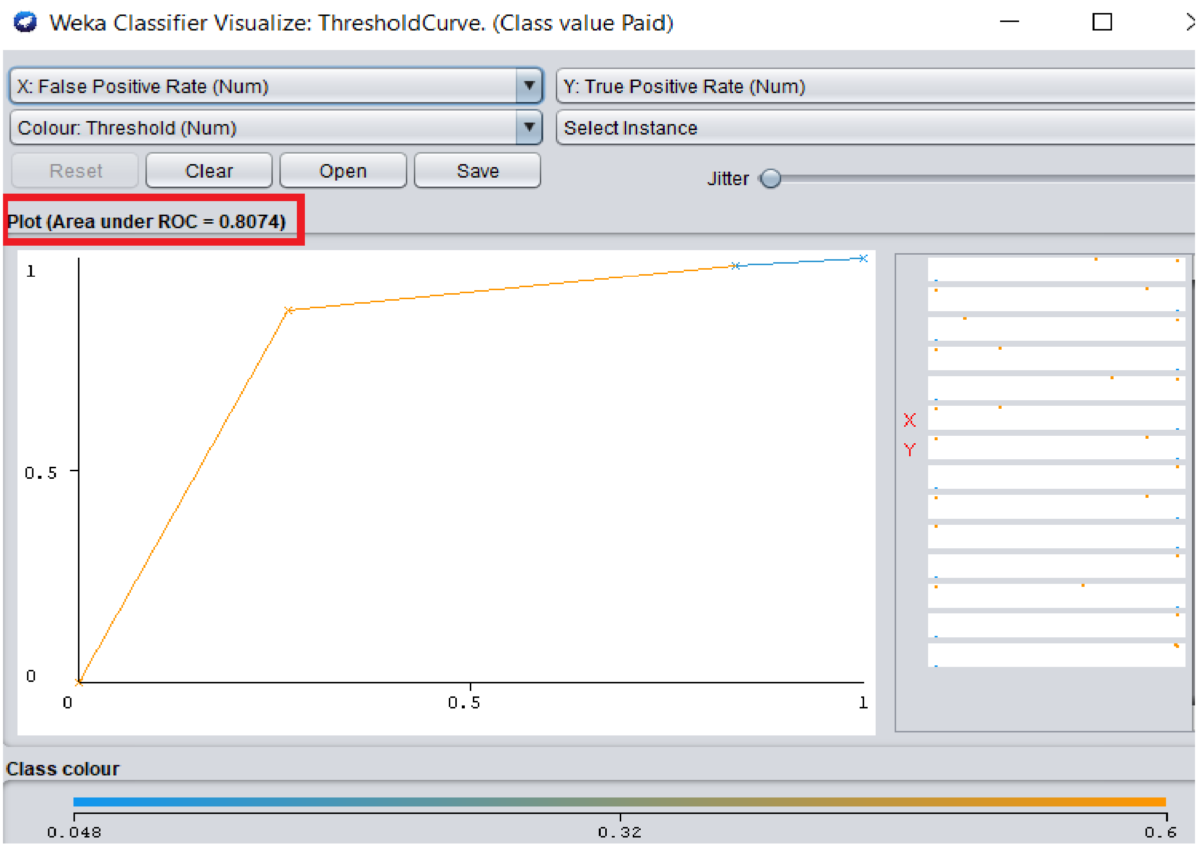


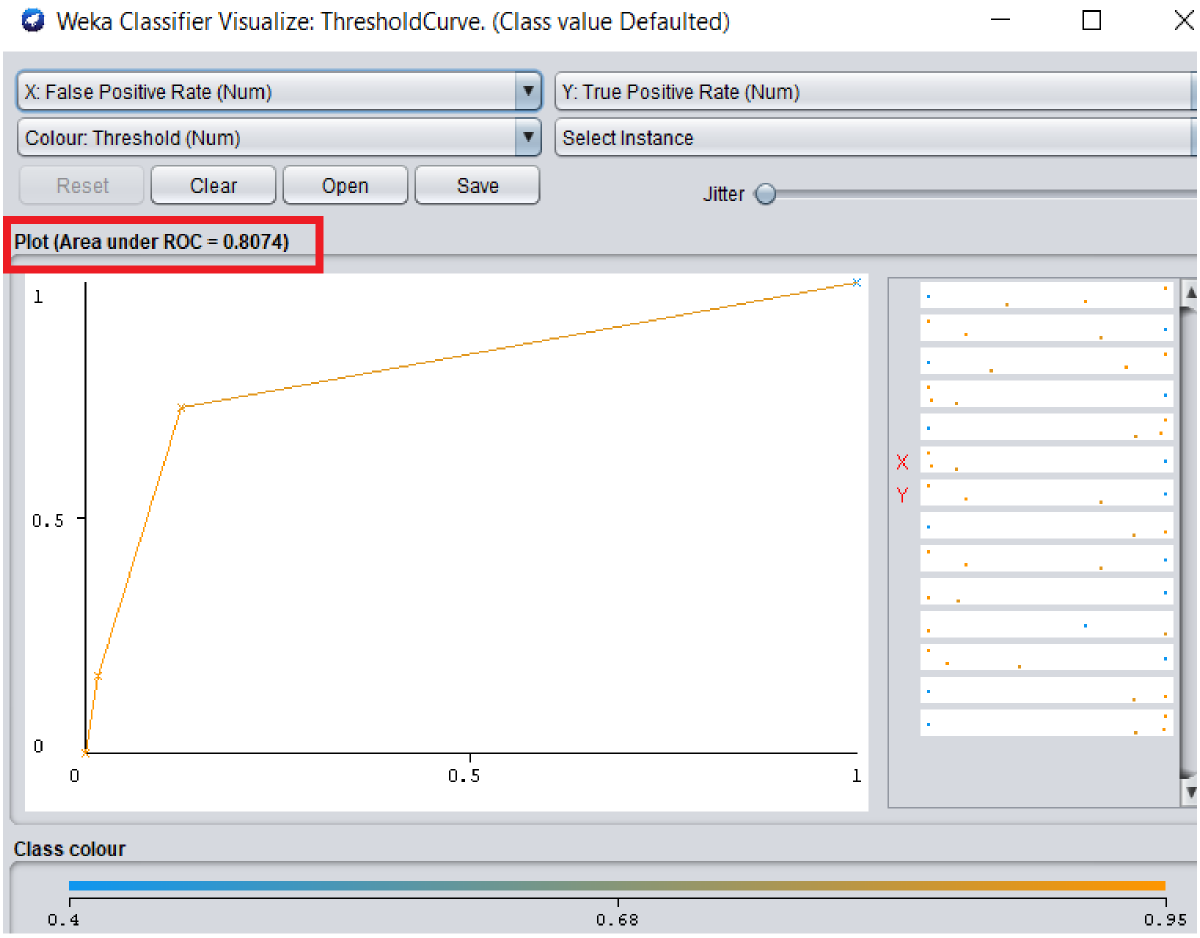
ROC analysis:

An ROC is used to analyse the TP (true positive) rate against the FP (false positive) rate for the possible number of cut points. ROC for J48 model is obtained from Weka, Exactness is estimated by the territory under the ROC. Area of 1 confirms a perfect test and less than 5.5 confirms a very bad test. In our case both obtained ROC curves are more than 5.5 (around 0.79 area). ROC for both Paid and Defaulted are shown below, in which **the area under ROC = 0.8074 ~0.80 (Approx.) for Paid prediction by the model and for Defaulted prediction by the model the Area Under ROC =0.8074~ 0.80(Approx.).**

In other simple words almost 80% of the prediction for both paid and Default is estimated exactly by the model which is visualised through the ROC curve.

When the ROC curve was compared between the models, J48 and J48 with cost sensitive learning, almost 80% area under the ROC is achieved by the model with J48 classifier (Cost sensitive Learning) with good accuracy and precision rate . J48 (Decision tree) stands out to predict better compared to other models. Screenshots of ROC for J48 classifier model with cost sensitive learning is shown below,

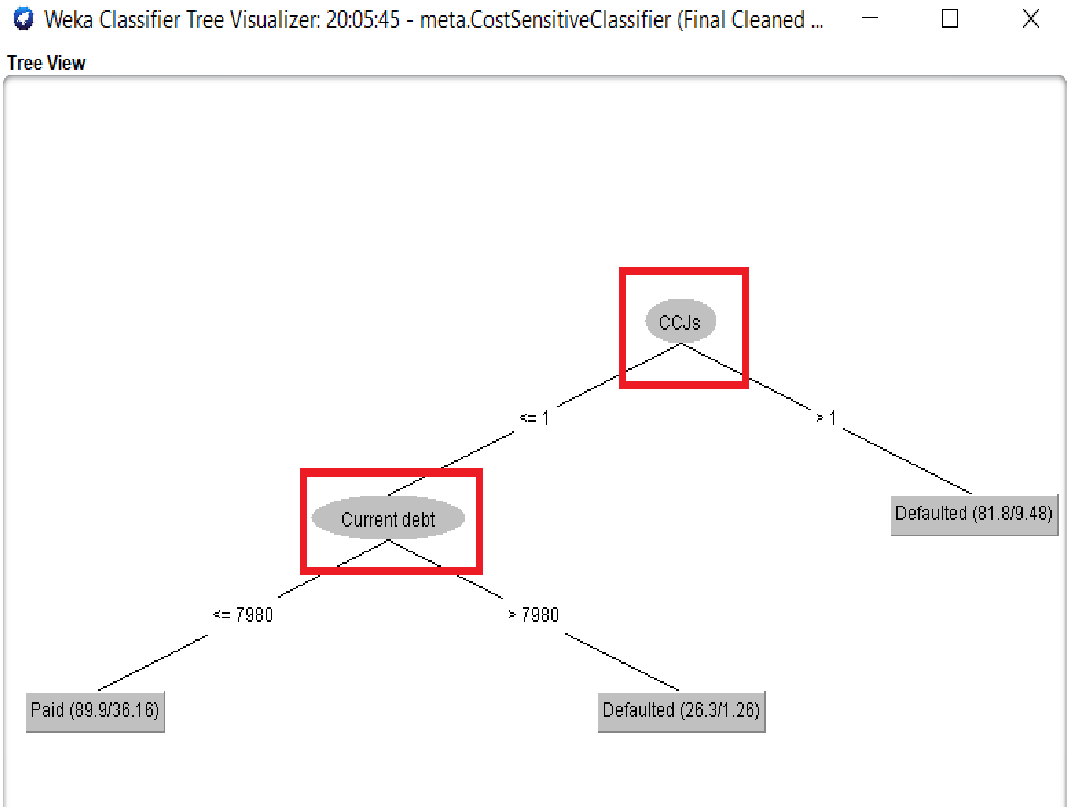




From the above ROC curve of the model we can see that almost 80% of the area is under the curve which proves that the model is smart in predicting the customers correctly.

Conclusion

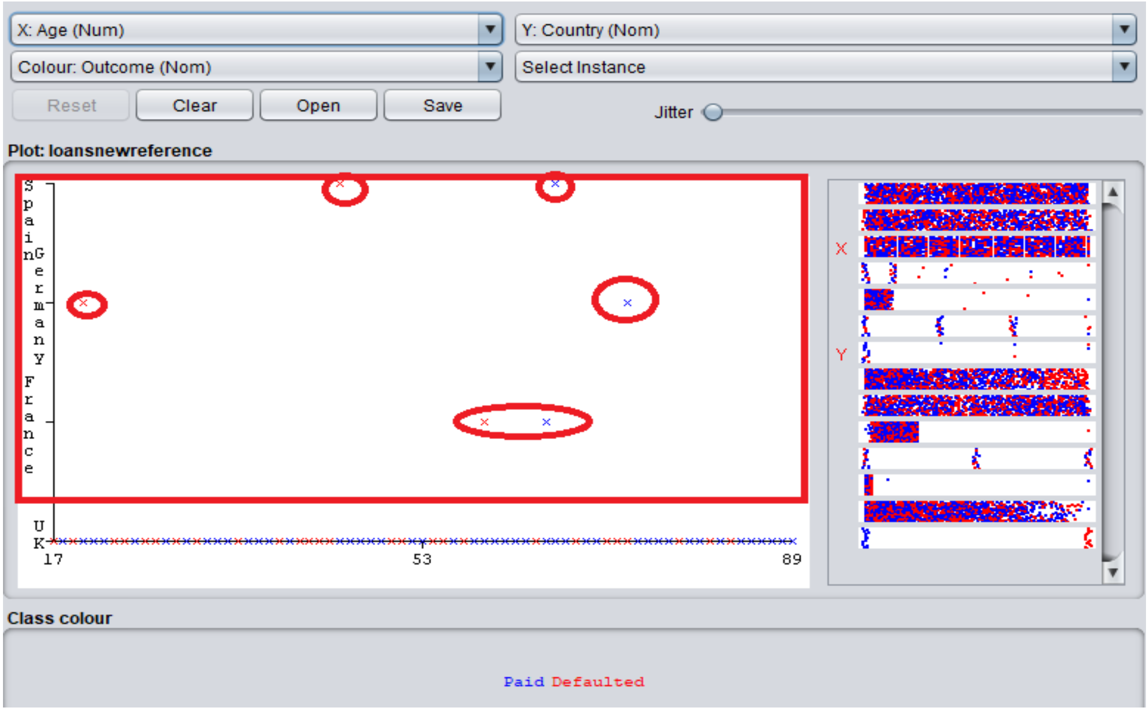
- **J48 classifier model (Cost Sensitive Learning)** with 10- fold cross validation method has better estimate of model accuracy, precision and ROC area.
- **Mainly J48 classifier model with cost sensitive learning approach has predicted maximum number of the Defaulters correctly**, which is the main objective of the model we are looking for, So J48 classifier model with cost sensitive learning stands out in better prediction of the Defaulters.
- Models performed better with the selected features – **CCJ, Current Debt, Employment status and Own home.**
- J48 Classifier tree structure throws light on the most important two variables **(CCJ, Current debt)** to classify the customers.



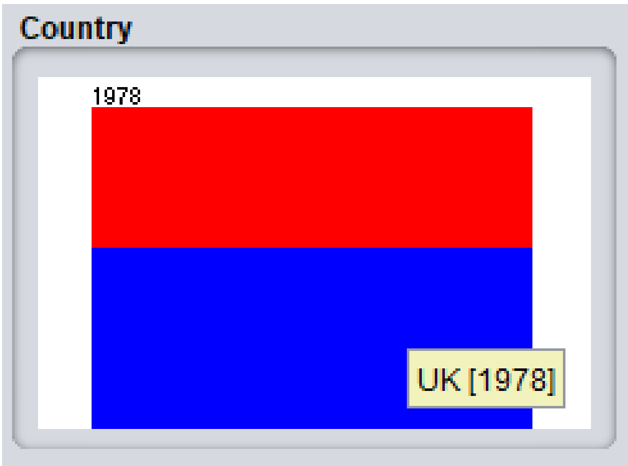
So based on the CCJ value and the Current Debt amount of a particular customer, the probability of that customer defaulting in future can be predicted and based on this factors a decision can be taken in the loan sanction.

Appendix – A

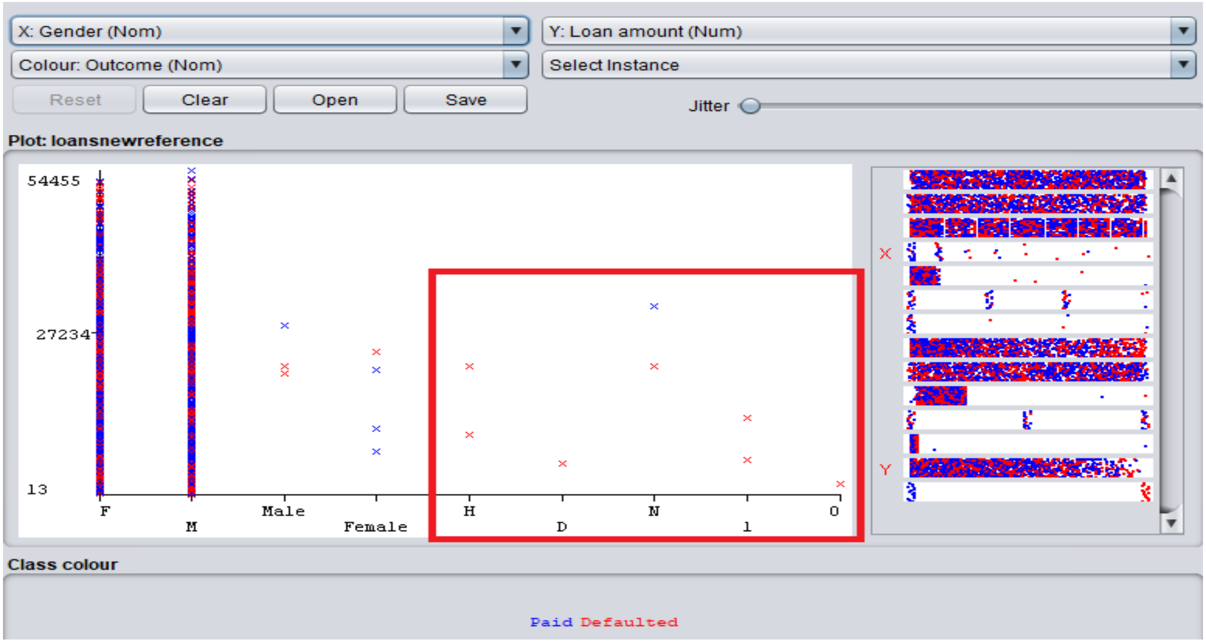
Country with outliers



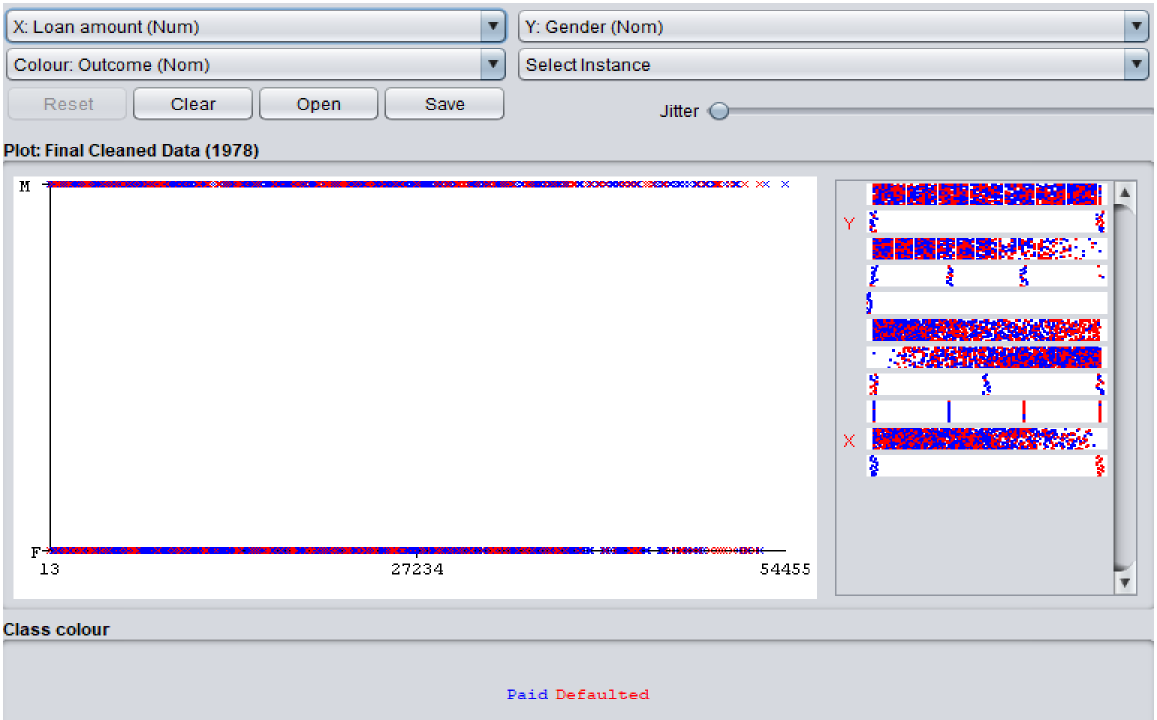
Country without outliers – removed spain, france,Germany



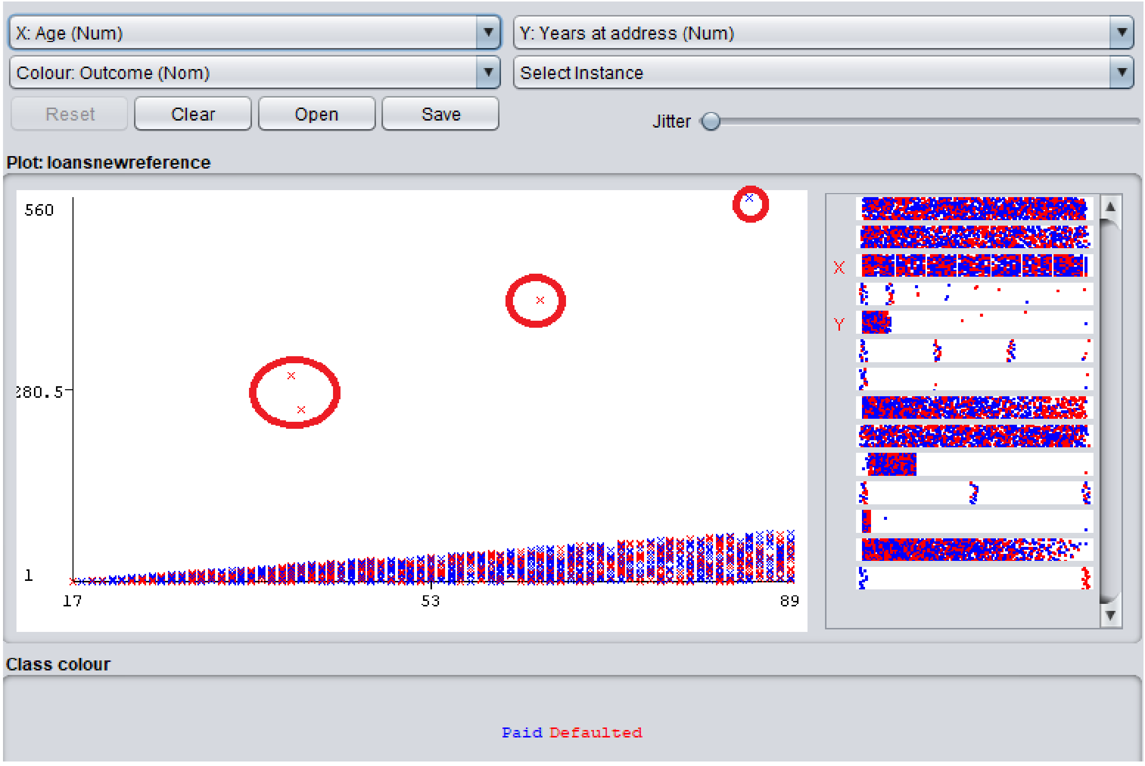
Gender with Outliers

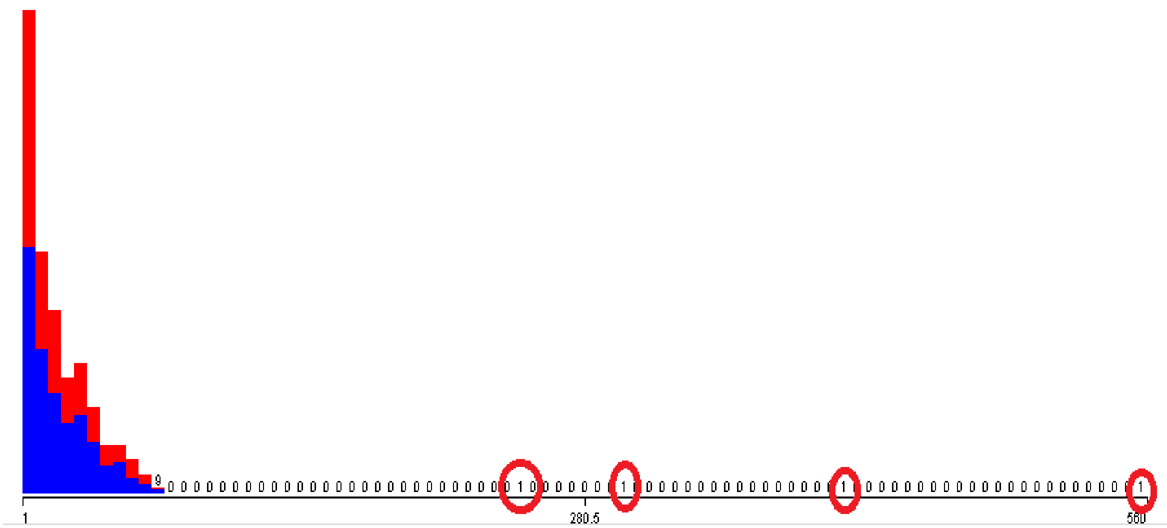


Gender with no outliers

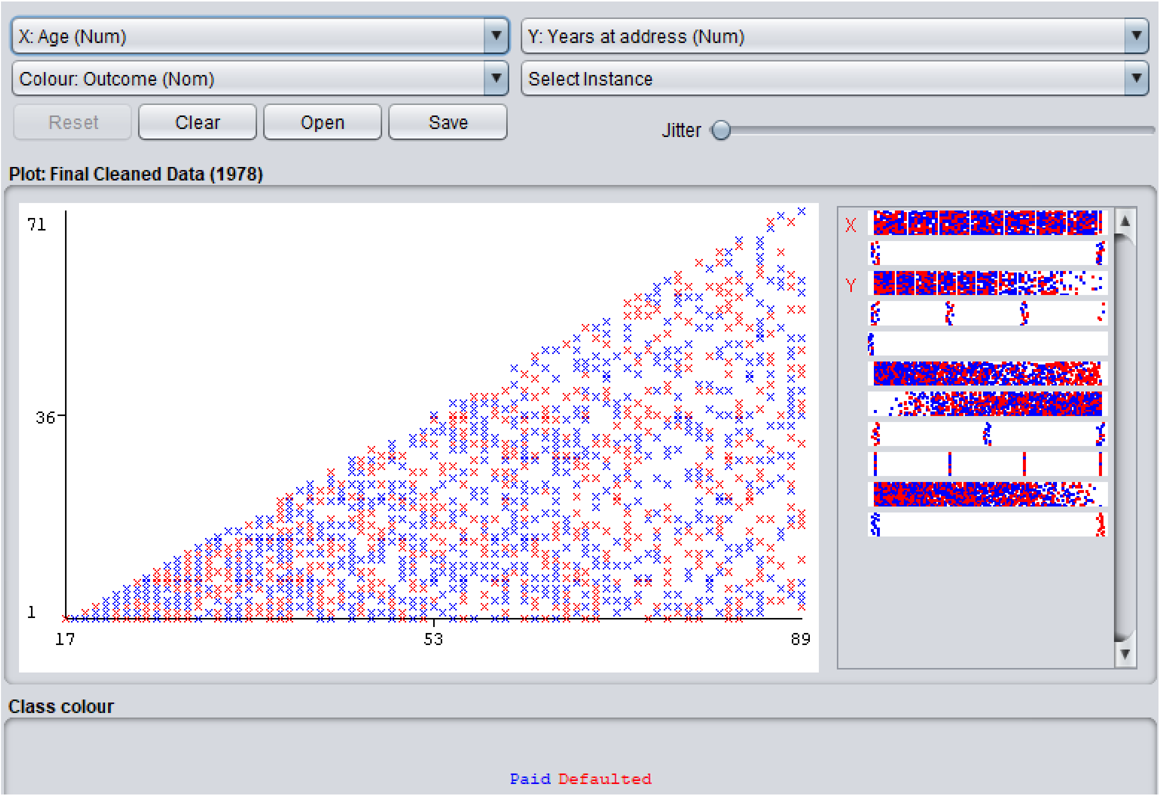


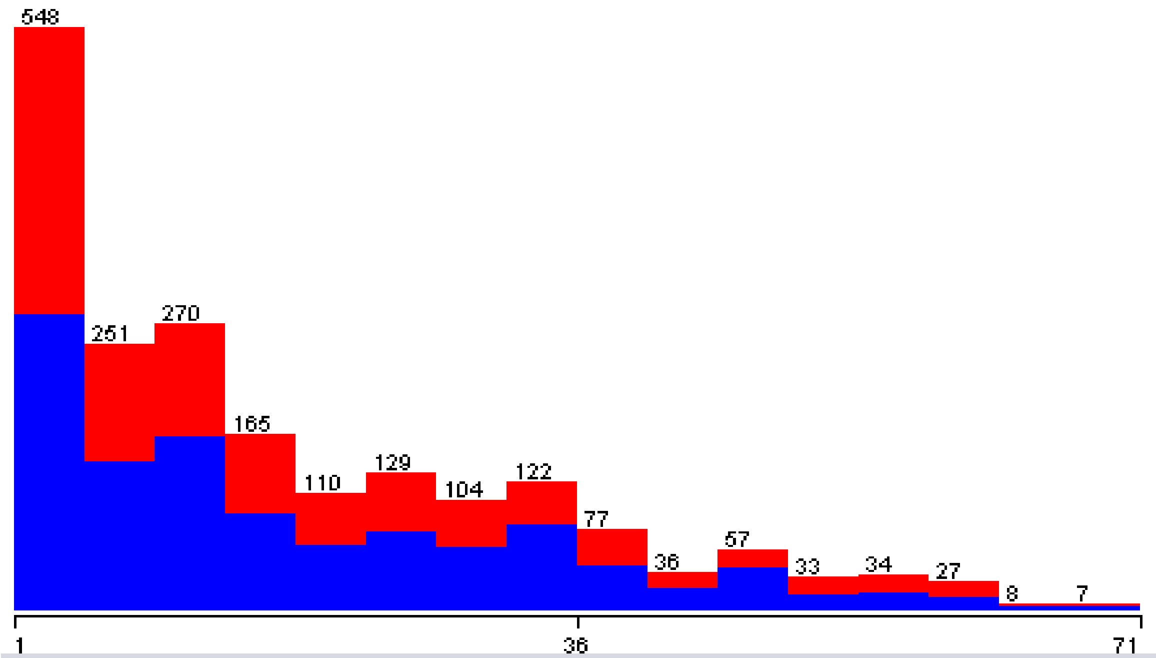
Years at address – with four outliers spotted





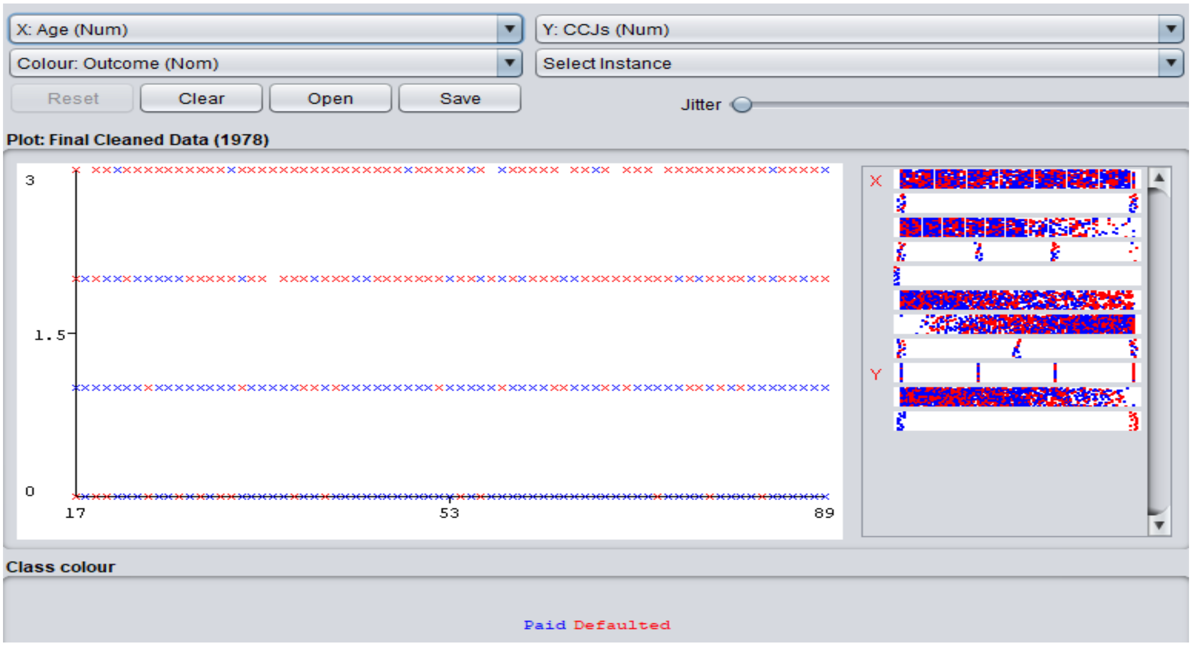
Years at Address – After removing outliers

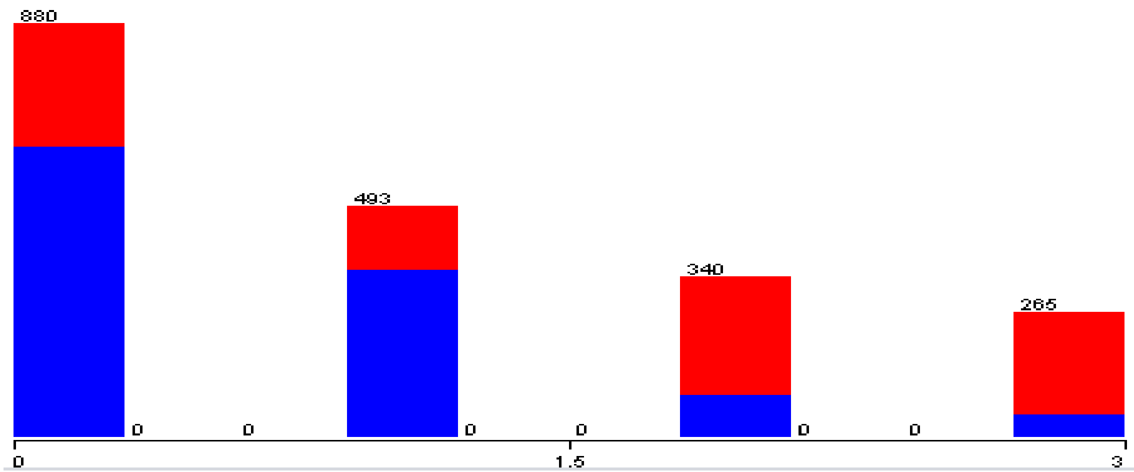




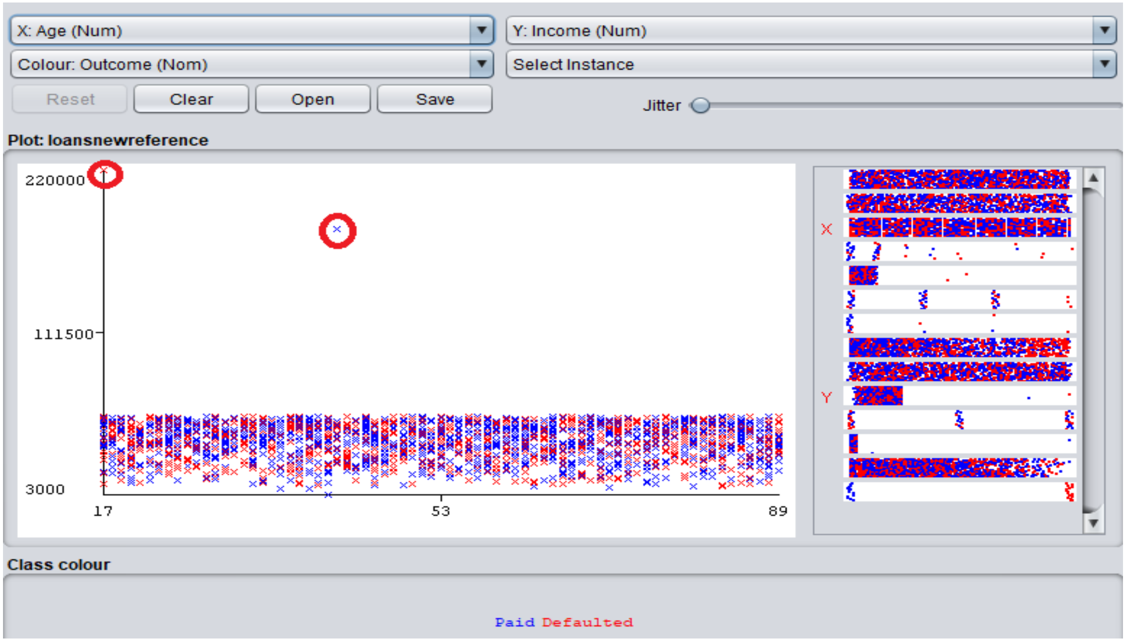
CCJ with two outliers spotted

CCJ's – after removing outliers

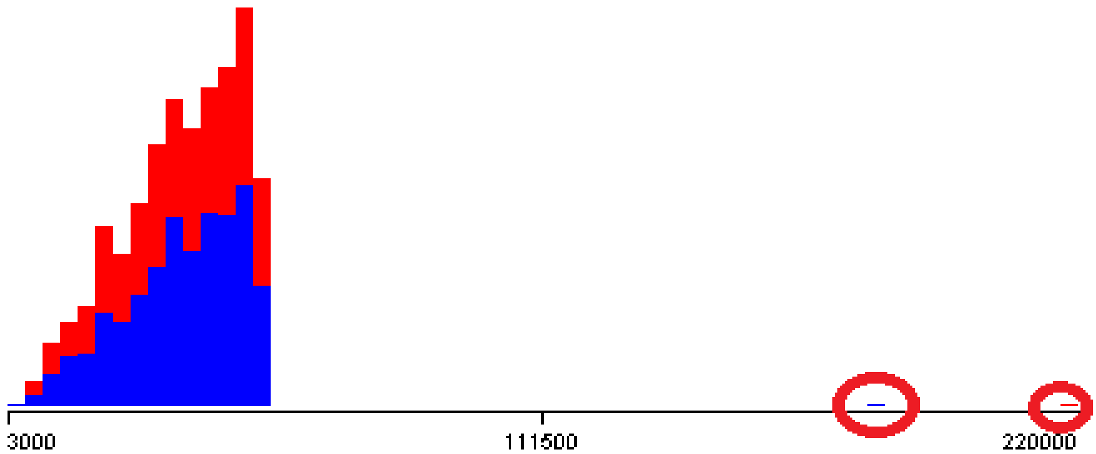




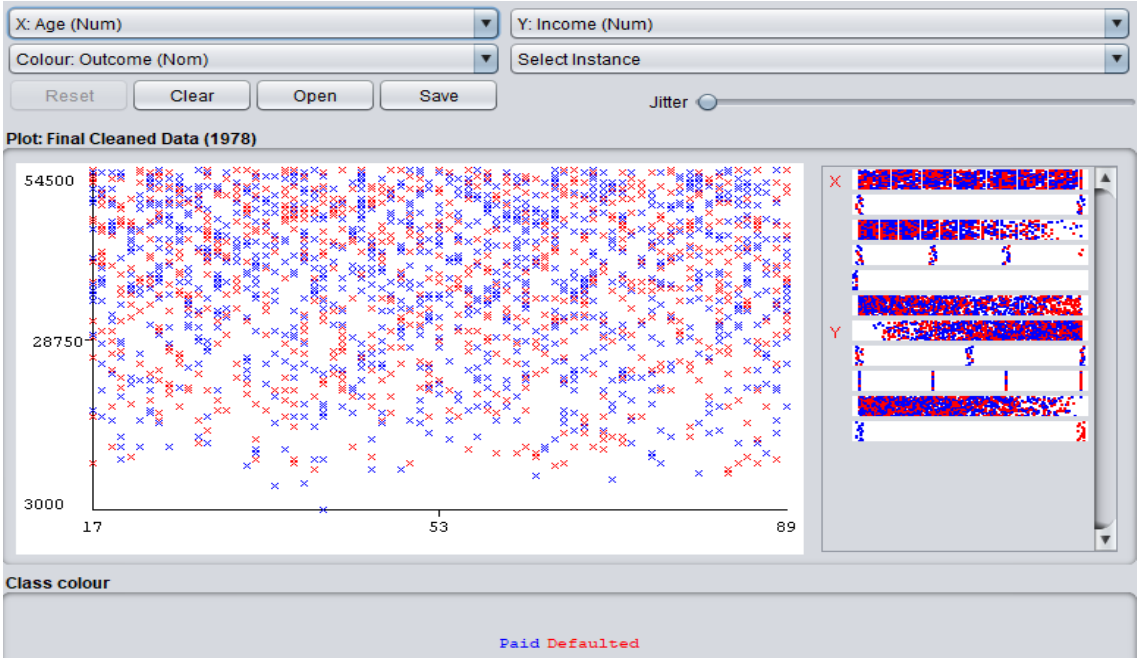
Income with two outliers

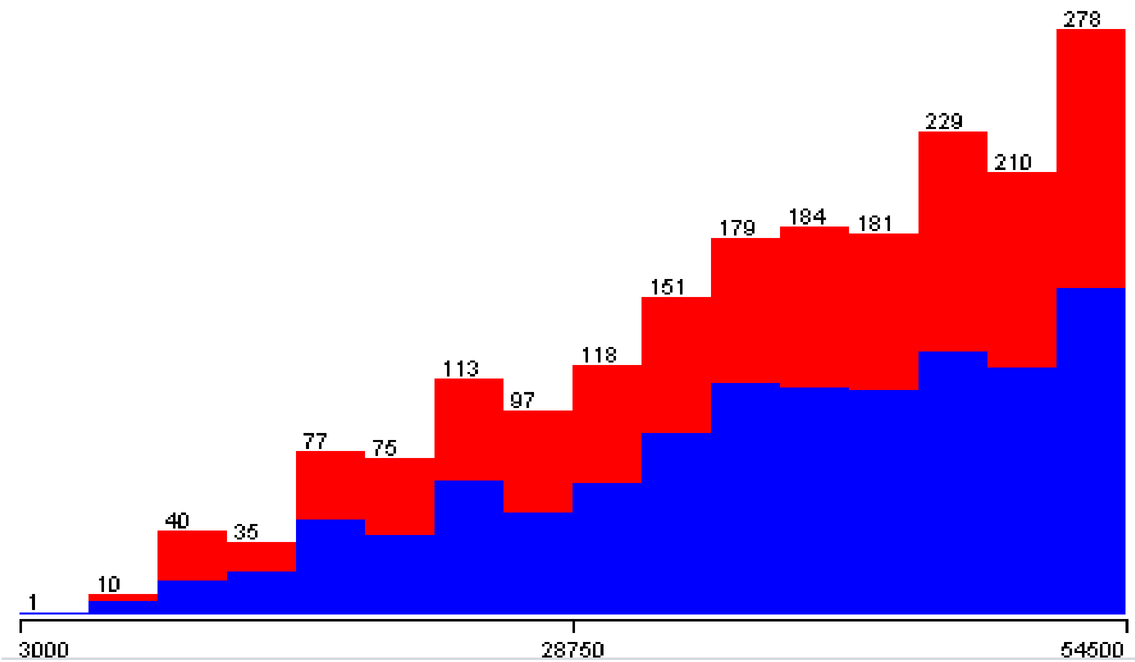


Income – with outliers



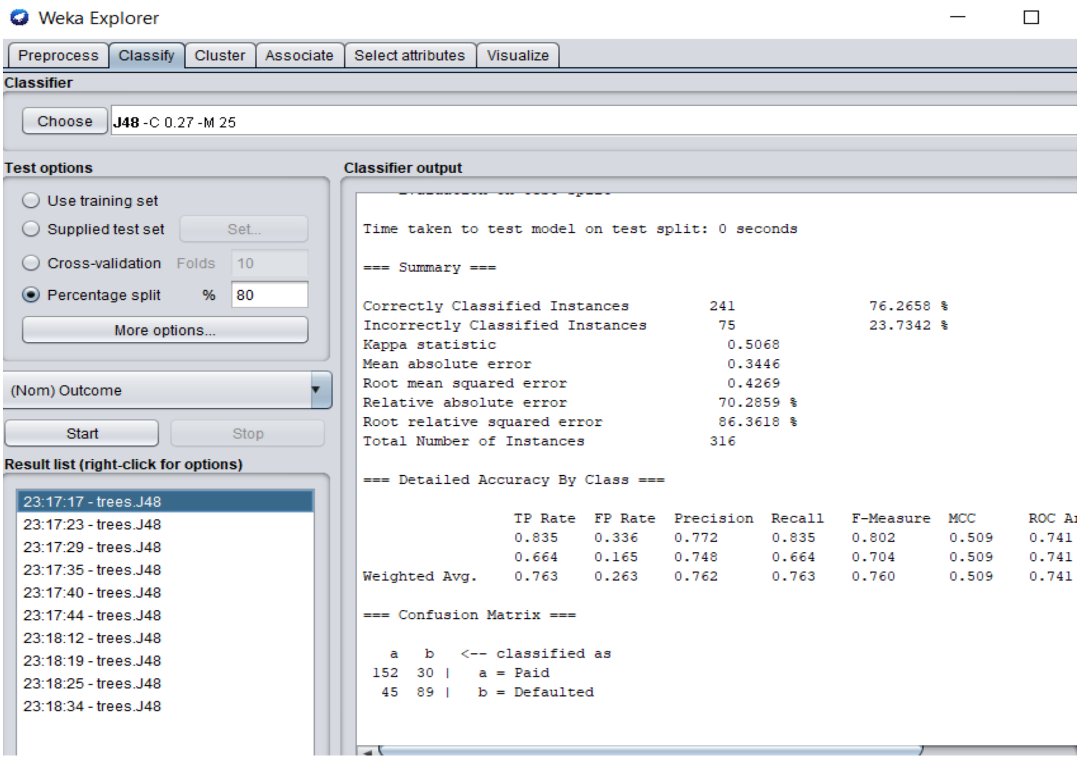
Income – after removing outlier

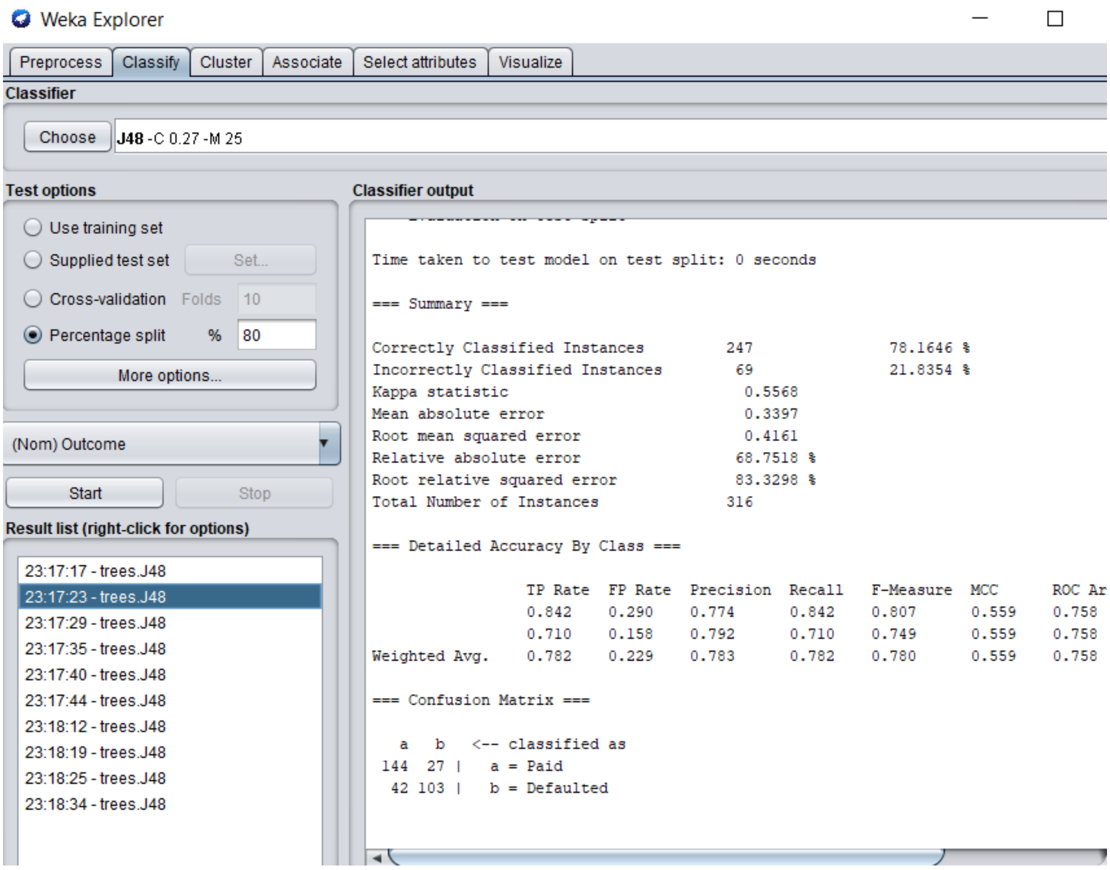


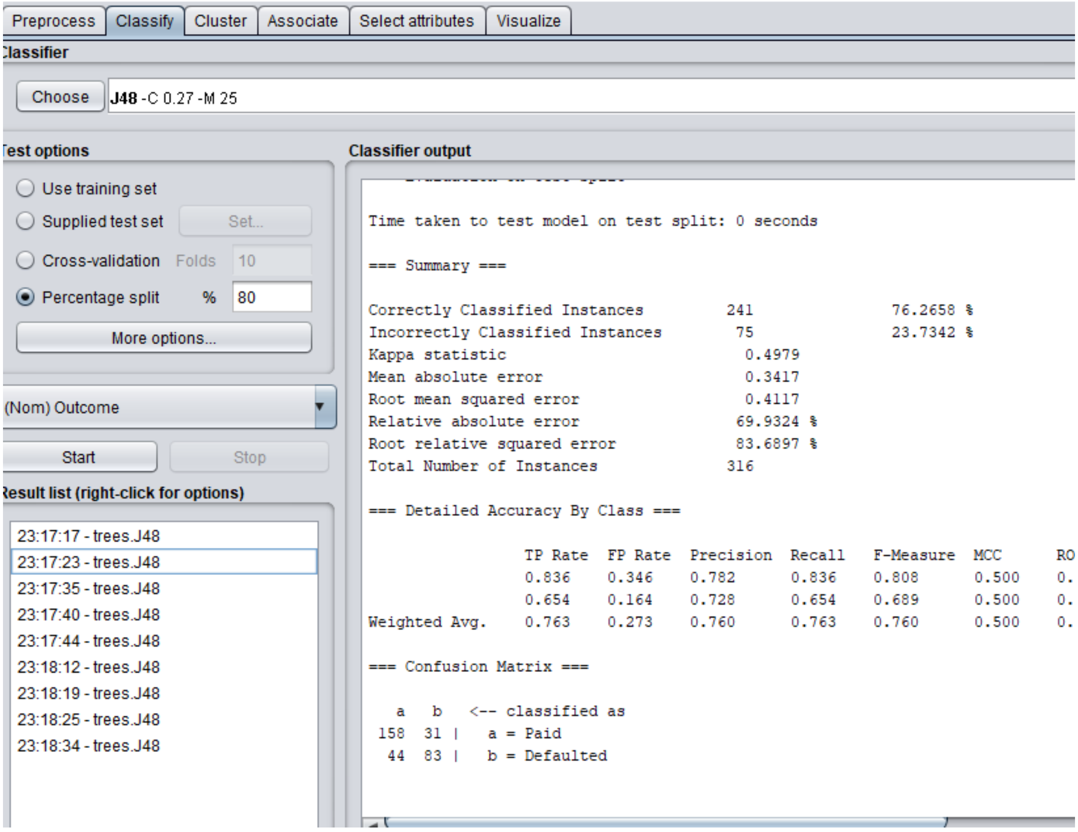


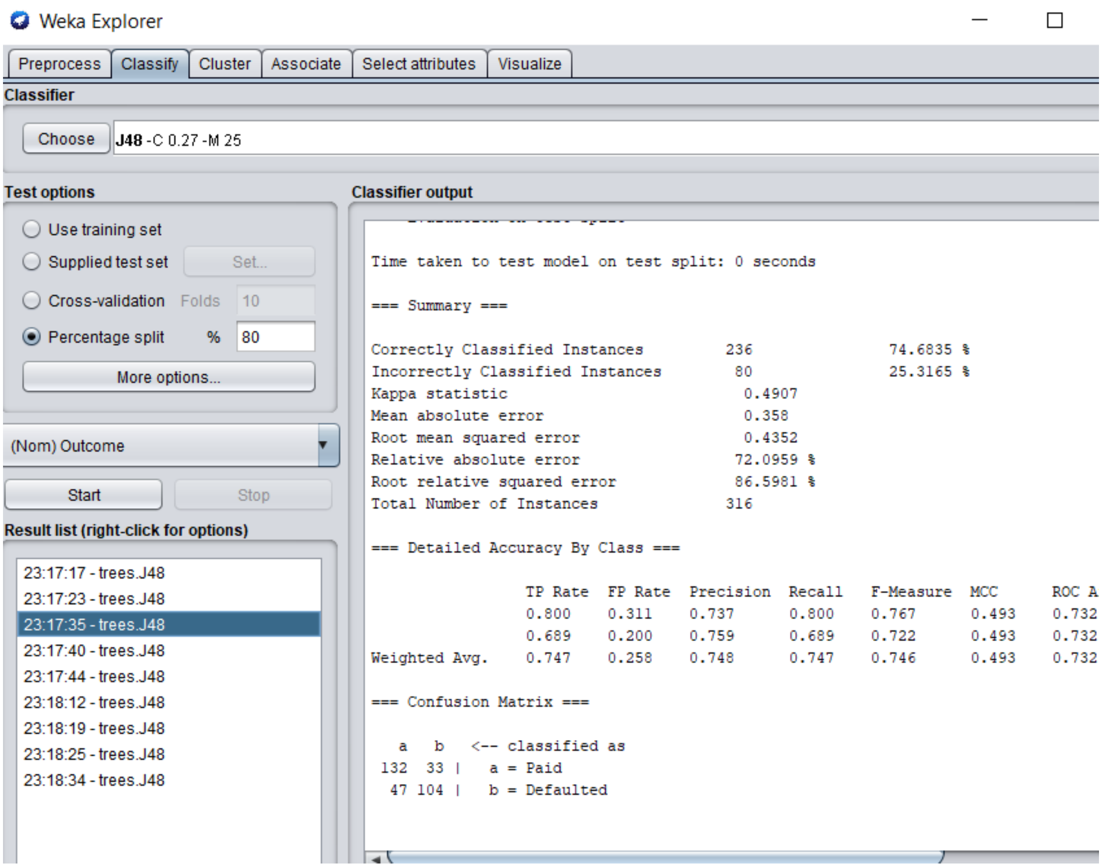
Appendix B

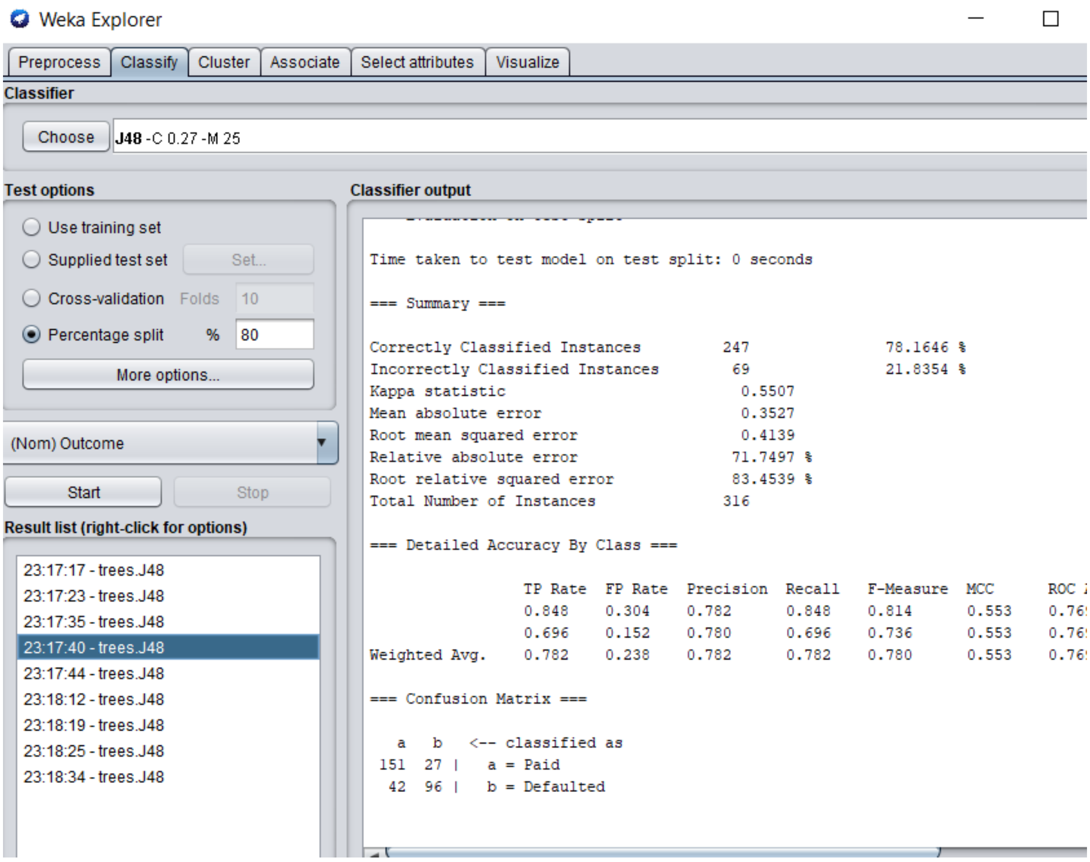
J48 classifier - %split before feature selection (hold out method)

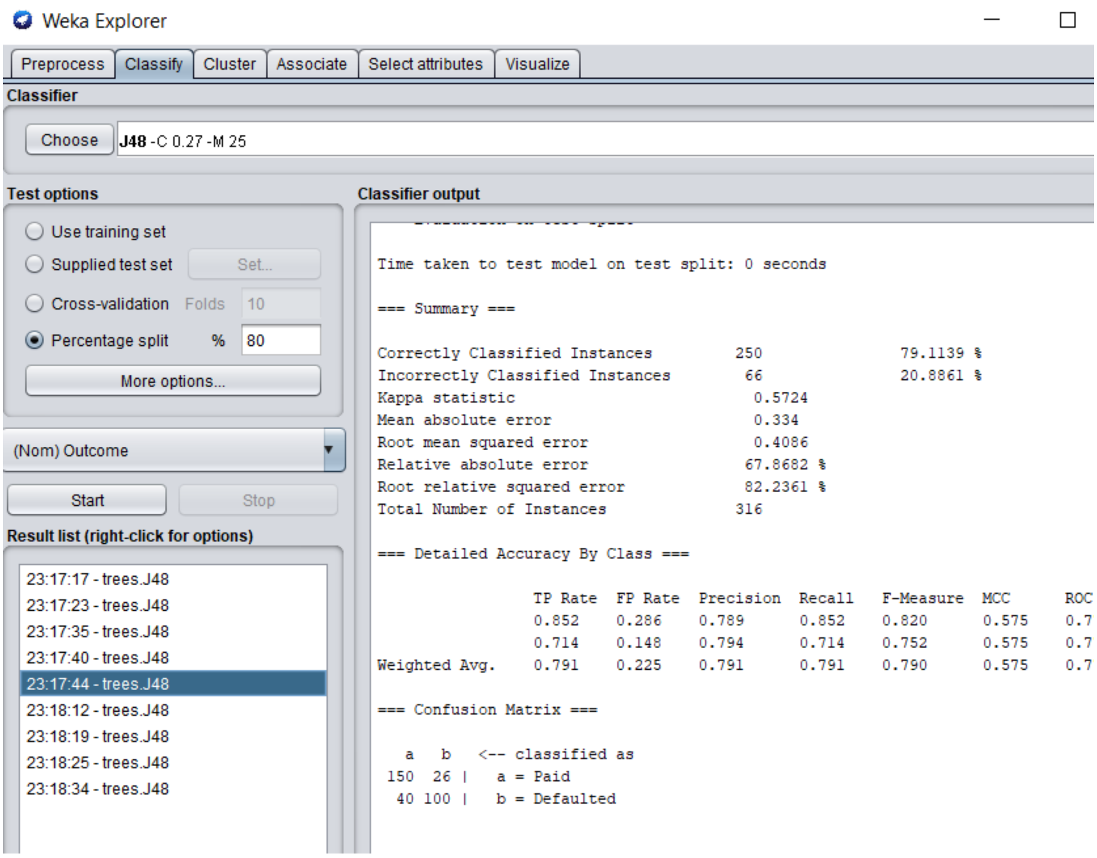


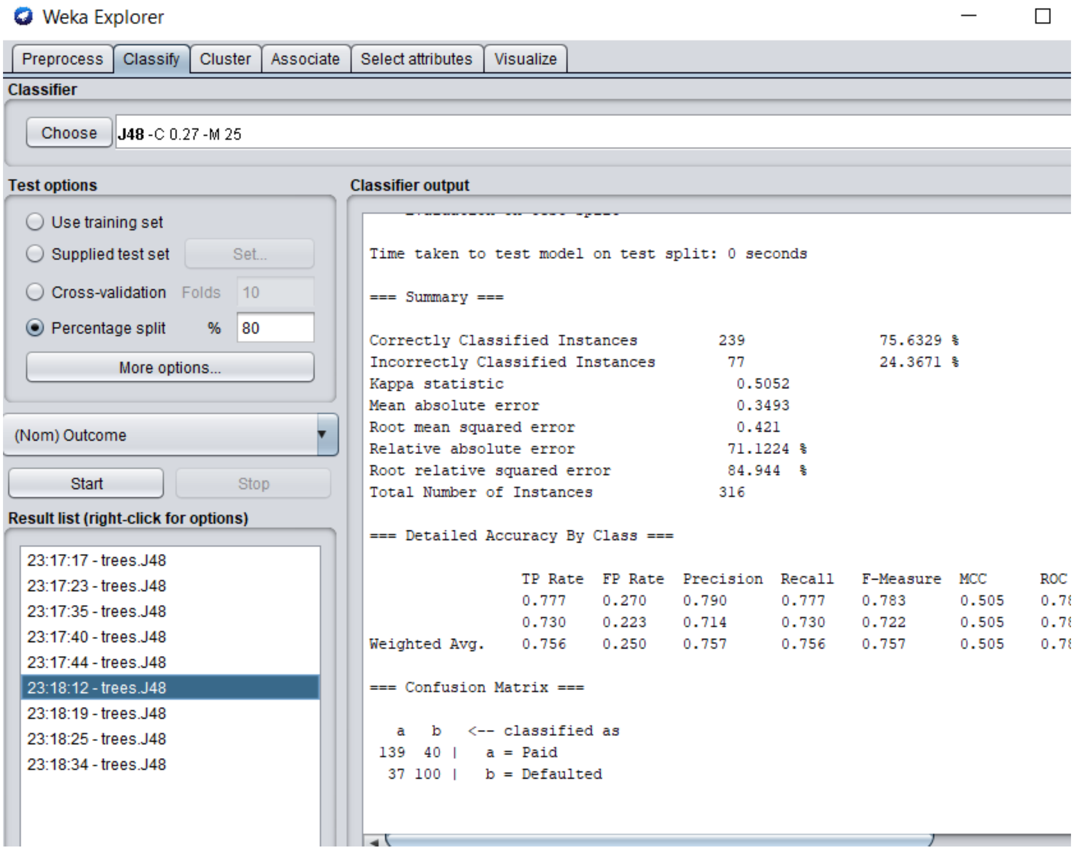


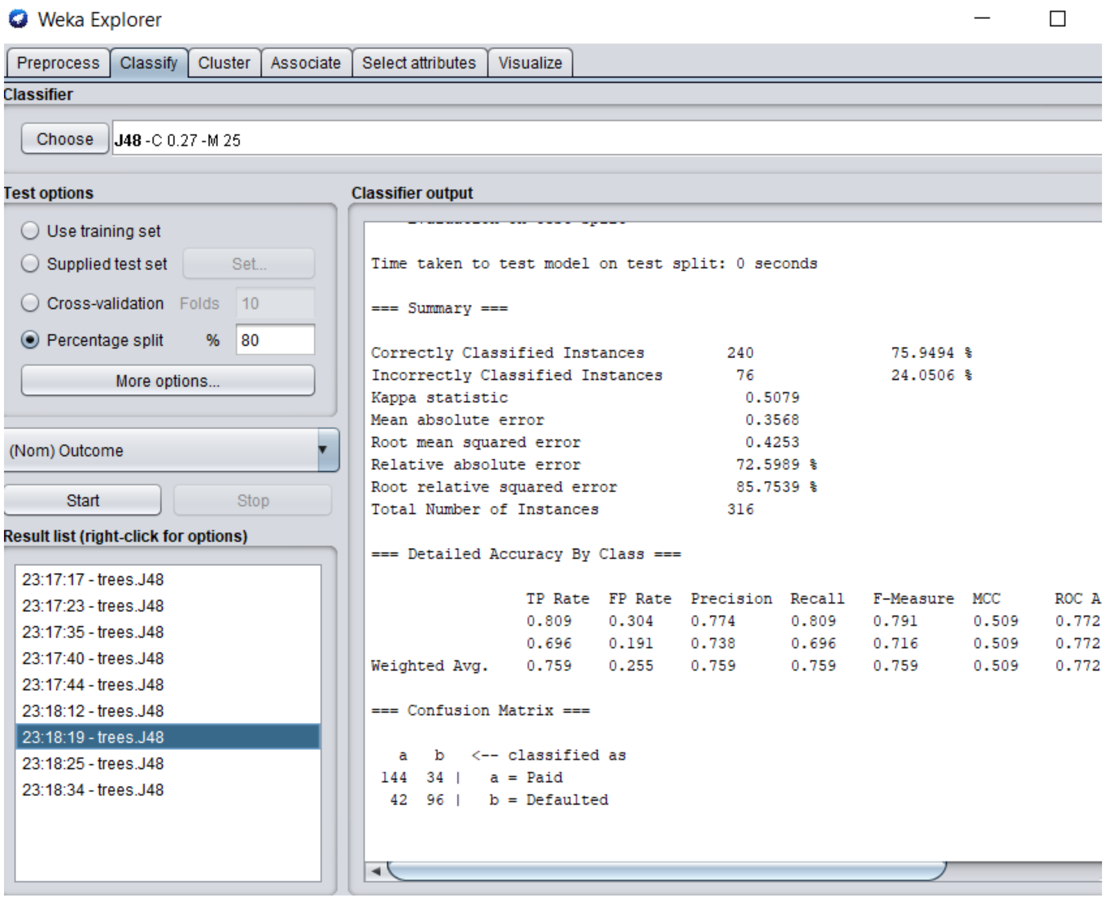

Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 241 76.2658 %
Incorrectly Classified Instances 75 23.7342 %
Kappa statistic 0.4979
Mean absolute error 0.3417
Root mean squared error 0.4117
Relative absolute error 69.9324 %
Root relative squared error 83.6897 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC RO
0.836 0.346 0.782 0.836 0.808 0.500 0.
0.654 0.164 0.728 0.654 0.689 0.500 0.
Weighted Avg. 0.763 0.273 0.760 0.763 0.760 0.500 0.
=== Confusion Matrix ===
a b <-- classified as
158 31 | a = Paid
44 83 | b = Defaulted

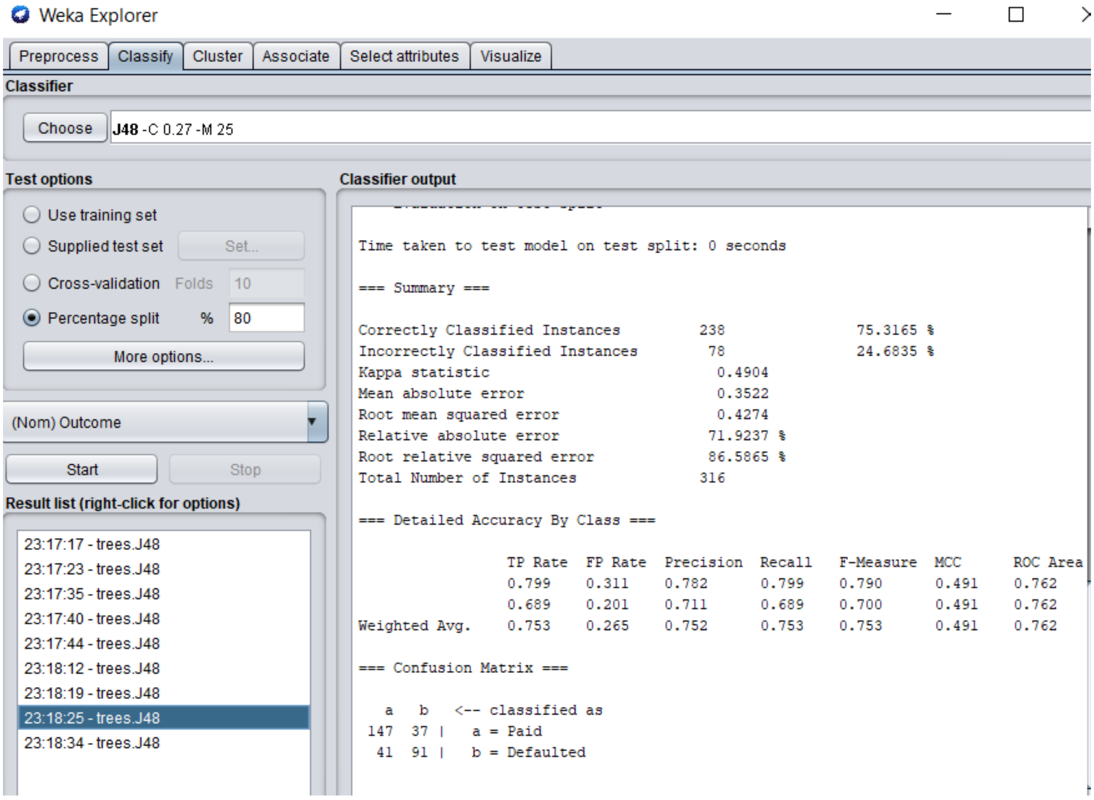

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 236 74.6835 %
Incorrectly Classified Instances 80 25.3165 %
Kappa statistic 0.4907
Mean absolute error 0.358
Root mean squared error 0.4352
Relative absolute error 72.0959 %
Root relative squared error 86.5981 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC A
0.800 0.311 0.737 0.800 0.767 0.493 0.732
0.689 0.200 0.759 0.689 0.722 0.493 0.732
Weighted Avg. 0.747 0.258 0.748 0.747 0.746 0.493 0.732
=== Confusion Matrix ===
a b <-- classified as
132 33 | a = Paid
47 104 | b = Defaulted

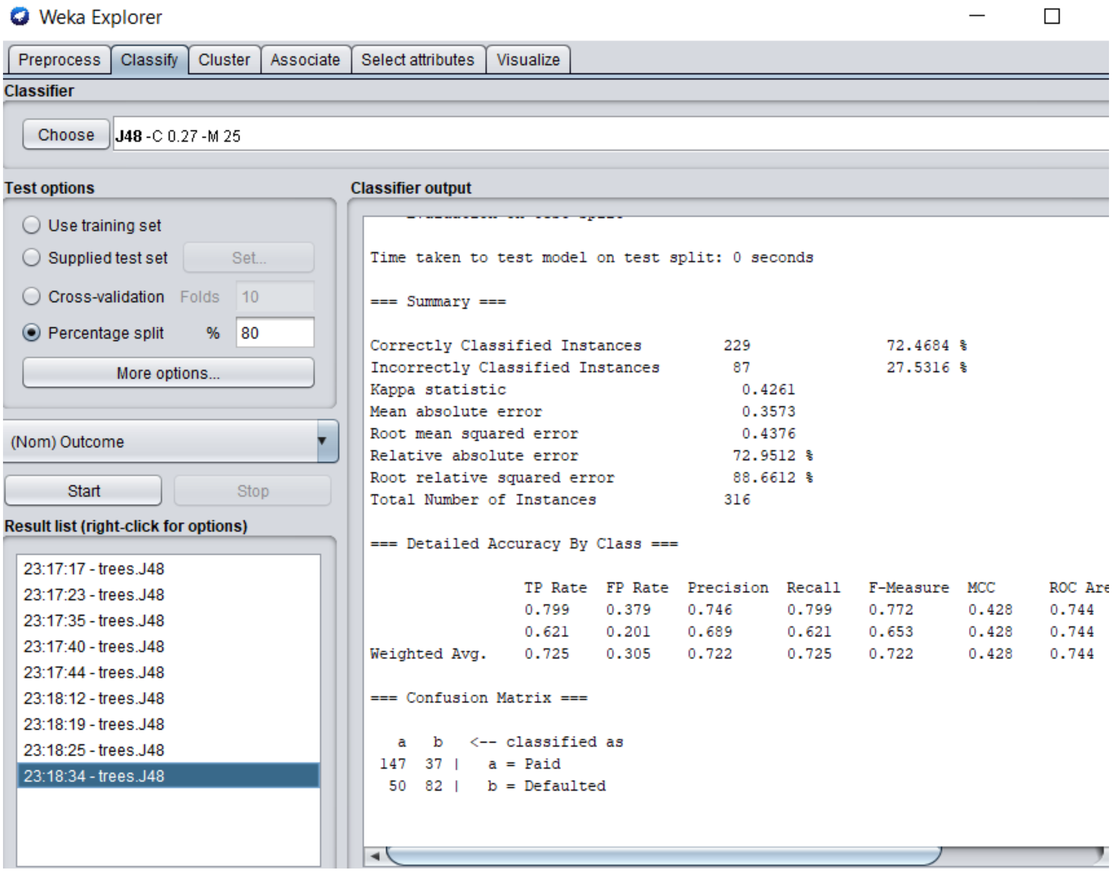

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 247 78.1646 %
Incorrectly Classified Instances 69 21.8354 %
Kappa statistic 0.5507
Mean absolute error 0.3527
Root mean squared error 0.4139
Relative absolute error 71.7497 %
Root relative squared error 83.4539 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.848 0.304 0.782 0.848 0.814 0.553
0.696 0.152 0.780 0.696 0.736 0.553
Weighted Avg. 0.782 0.238 0.782 0.782 0.780 0.553
=== Confusion Matrix ===
a b <-- classified as
151 27 | a = Paid
42 96 | b = Defaulted

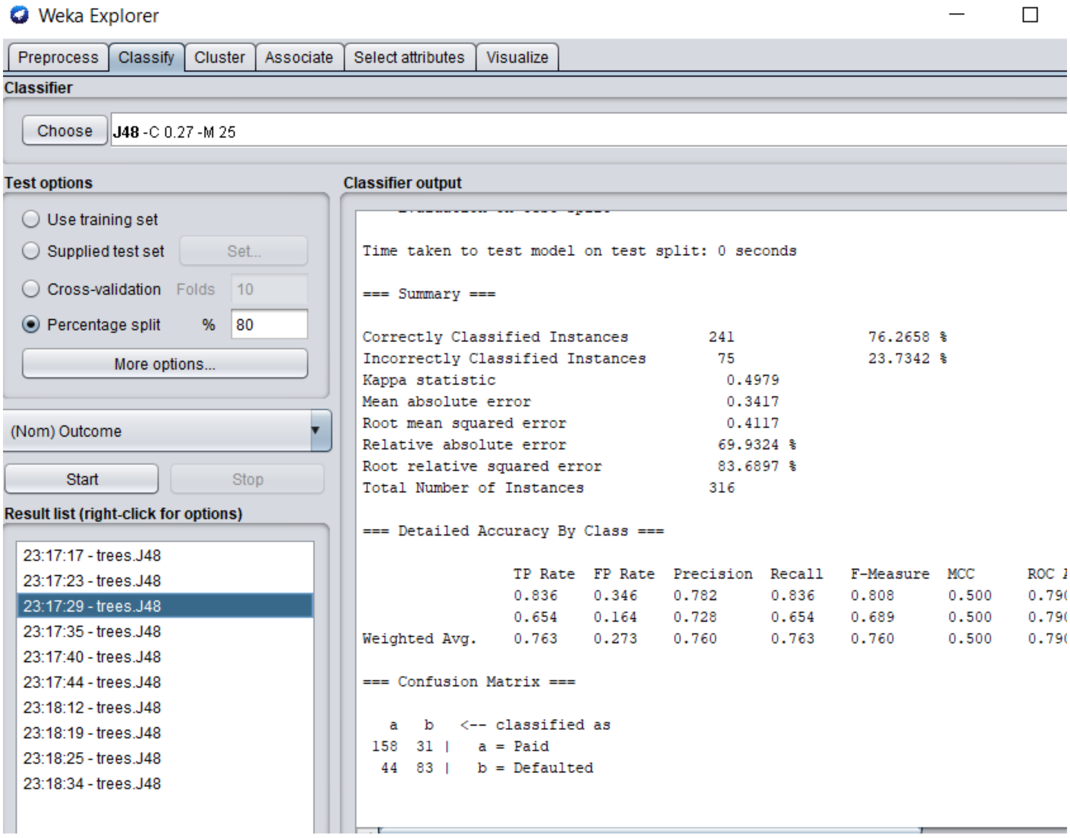

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 250 79.1139 %
Incorrectly Classified Instances 66 20.8861 %
Kappa statistic 0.5724
Mean absolute error 0.334
Root mean squared error 0.4086
Relative absolute error 67.8682 %
Root relative squared error 82.2361 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.852 0.286 0.789 0.852 0.820 0.575 0.7
0.714 0.148 0.794 0.714 0.752 0.575 0.7
Weighted Avg. 0.791 0.225 0.791 0.791 0.790 0.575 0.7
=== Confusion Matrix ===
a b <-- classified as
150 26 | a = Paid
40 100 | b = Defaulted

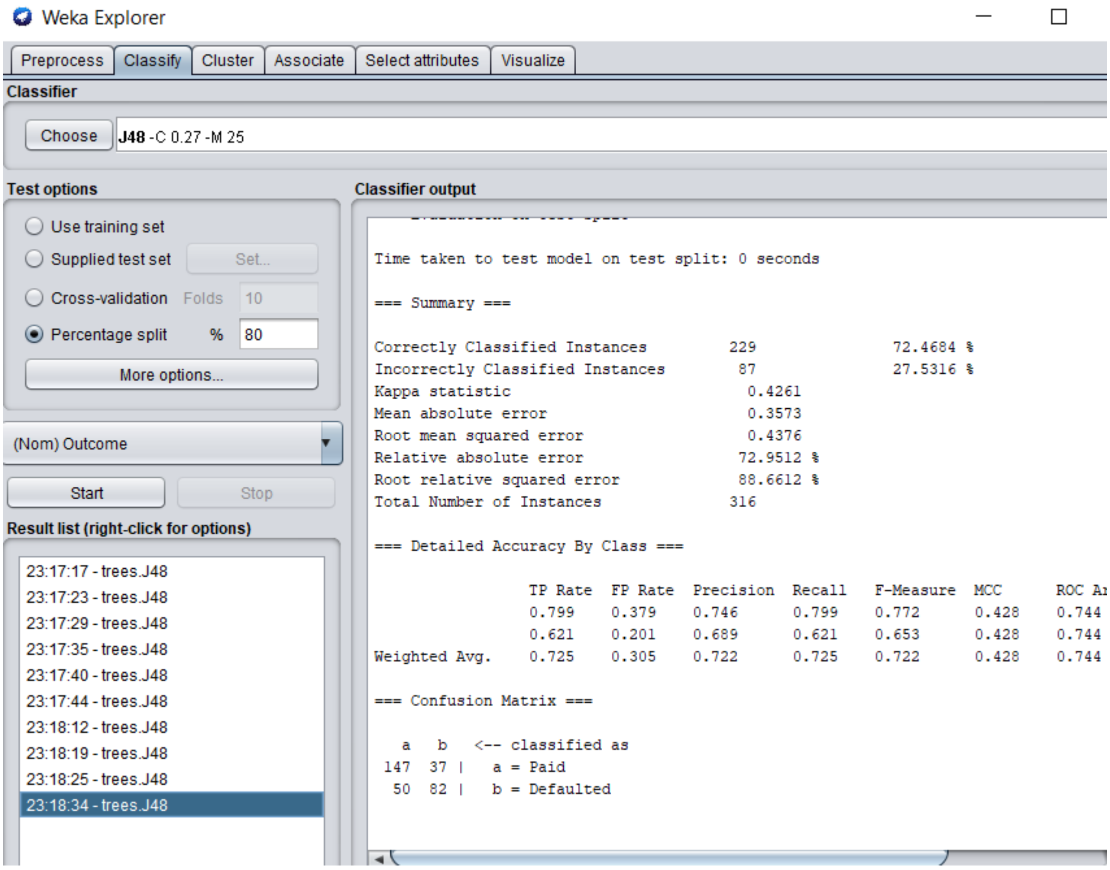

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 239 75.6329 %
Incorrectly Classified Instances 77 24.3671 %
Kappa statistic 0.5052
Mean absolute error 0.3493
Root mean squared error 0.421
Relative absolute error 71.1224 %
Root relative squared error 84.944 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.777 0.270 0.790 0.777 0.783 0.505
0.730 0.223 0.714 0.730 0.722 0.505
Weighted Avg. 0.756 0.250 0.757 0.756 0.757 0.505
=== Confusion Matrix ===
a b <-- classified as
139 40 | a = Paid
37 100 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 240 75.9494 %
Incorrectly Classified Instances 76 24.0506 %
Kappa statistic 0.5079
Mean absolute error 0.3568
Root mean squared error 0.4253
Relative absolute error 72.5989 %
Root relative squared error 85.7539 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC A
0.809 0.304 0.774 0.809 0.791 0.509 0.772
0.696 0.191 0.738 0.696 0.716 0.509 0.772
Weighted Avg. 0.759 0.255 0.759 0.759 0.759 0.509 0.772
=== Confusion Matrix ===
a b <-- classified as
144 34 | a = Paid
42 96 | b = Defaulted

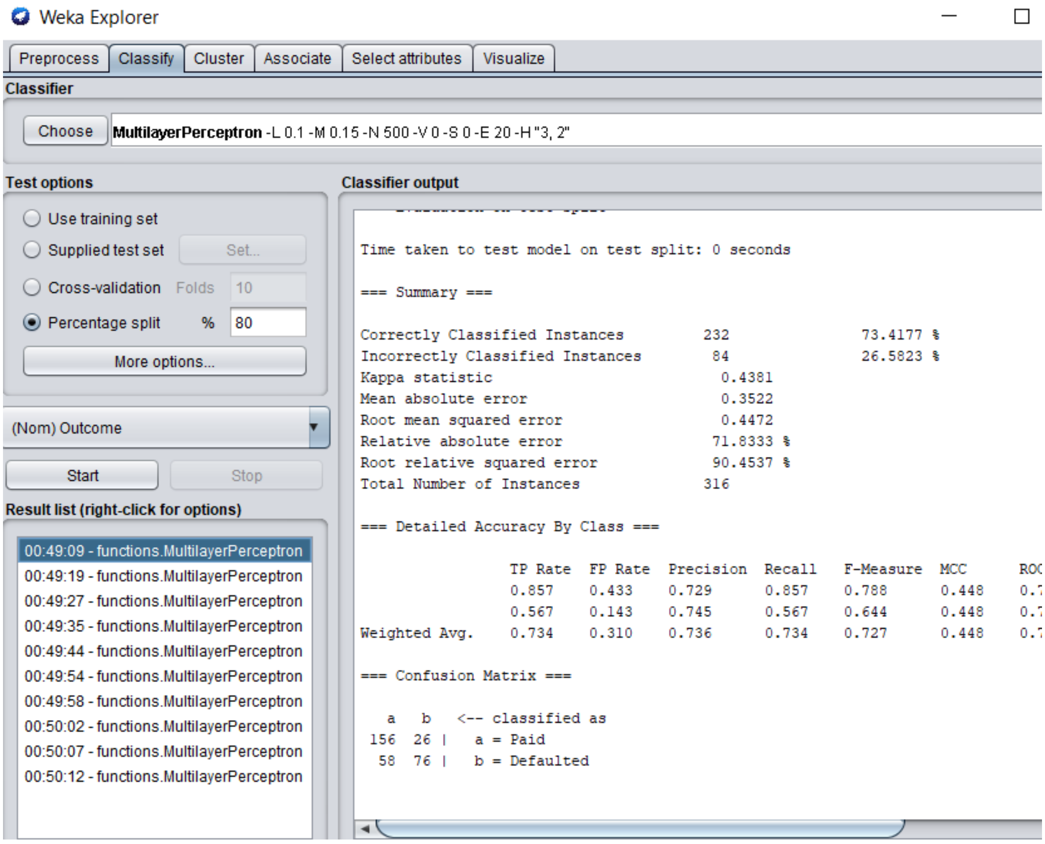

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 238 75.3165 %
Incorrectly Classified Instances 78 24.6835 %
Kappa statistic 0.4904
Mean absolute error 0.3522
Root mean squared error 0.4274
Relative absolute error 71.9237 %
Root relative squared error 86.5865 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area
0.799 0.311 0.782 0.799 0.790 0.491 0.762
0.689 0.201 0.711 0.689 0.700 0.491 0.762
Weighted Avg. 0.753 0.265 0.752 0.753 0.753 0.491 0.762
=== Confusion Matrix ===
a b <-- classified as
147 37 | a = Paid
41 91 | b = Defaulted

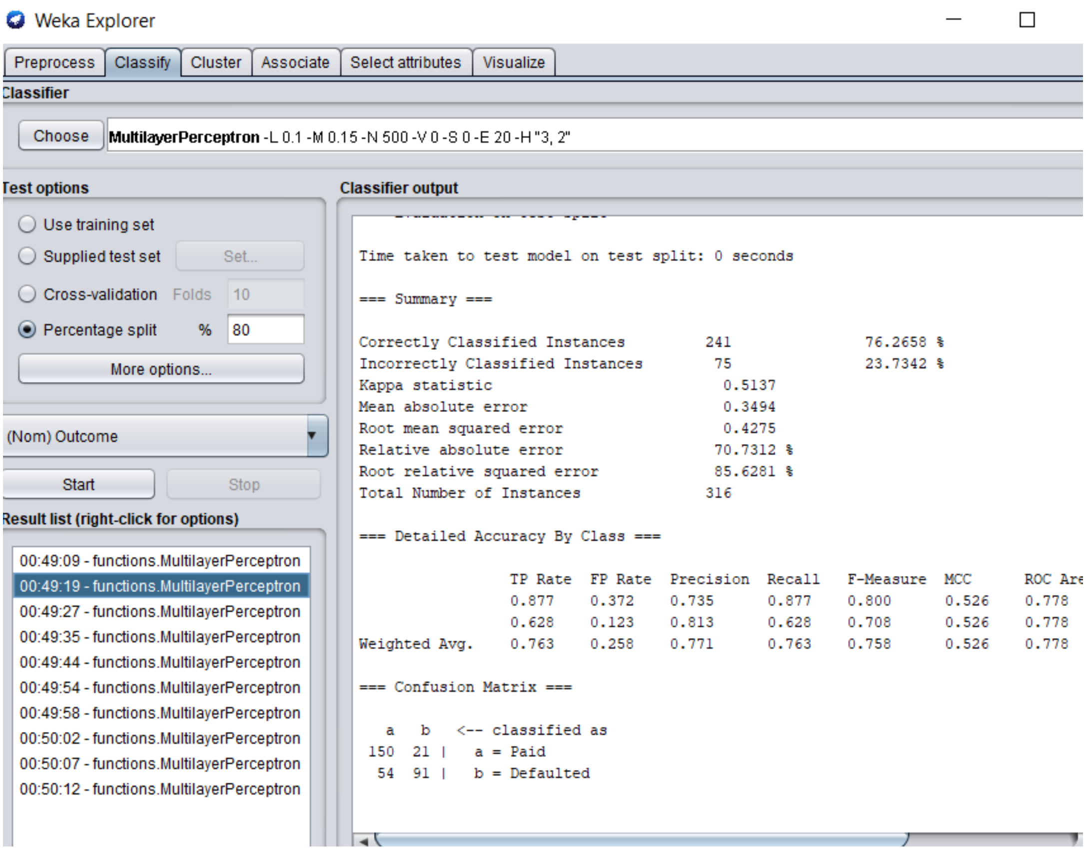

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 229 72.4684 %
Incorrectly Classified Instances 87 27.5316 %
Kappa statistic 0.4261
Mean absolute error 0.3573
Root mean squared error 0.4376
Relative absolute error 72.9512 %
Root relative squared error 88.6612 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.799 0.379 0.746 0.799 0.772 0.428 0.744
0.621 0.201 0.689 0.621 0.653 0.428 0.744
Weighted Avg. 0.725 0.305 0.722 0.725 0.722 0.428 0.744
=== Confusion Matrix ===
a b <-- classified as
147 37 | a = Paid
50 82 | b = Defaulted

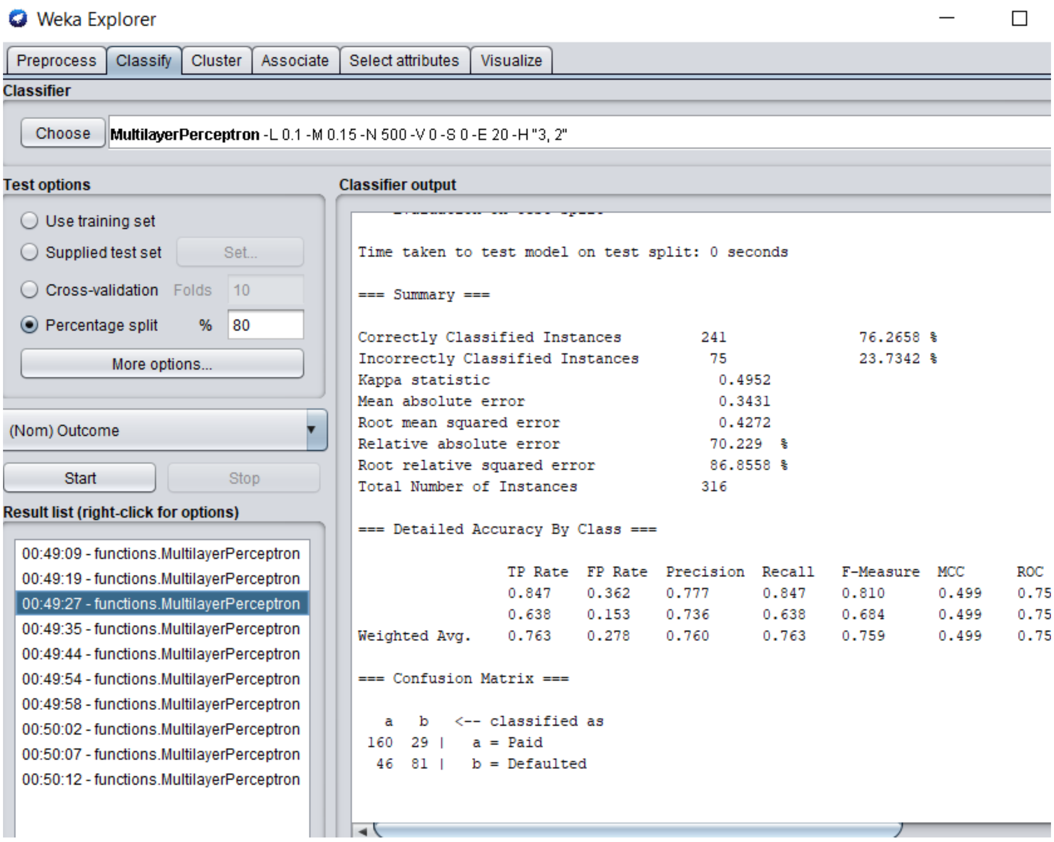

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:29 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 241 76.2658 %
Incorrectly Classified Instances 75 23.7342 %
Kappa statistic 0.4979
Mean absolute error 0.3417
Root mean squared error 0.4117
Relative absolute error 69.9324 %
Root relative squared error 83.6897 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.836 0.346 0.782 0.836 0.808 0.500
0.654 0.164 0.728 0.654 0.689 0.500
Weighted Avg. 0.763 0.273 0.760 0.763 0.760 0.500
=== Confusion Matrix ===
a b <-- classified as
158 31 | a = Paid
44 83 | b = Defaulted

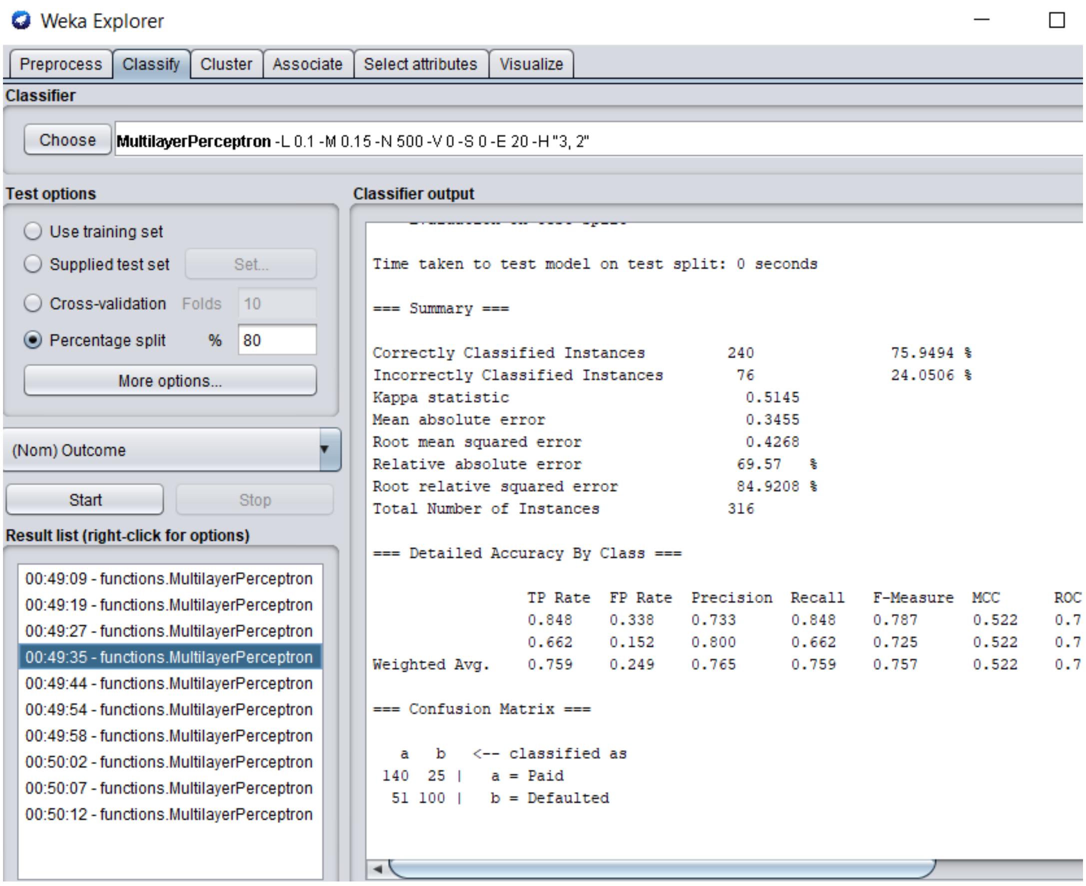

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
23:17:17 - trees.J48
23:17:23 - trees.J48
23:17:29 - trees.J48
23:17:35 - trees.J48
23:17:40 - trees.J48
23:17:44 - trees.J48
23:18:12 - trees.J48
23:18:19 - trees.J48
23:18:25 - trees.J48
23:18:34 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 229 72.4684 %
Incorrectly Classified Instances 87 27.5316 %
Kappa statistic 0.4261
Mean absolute error 0.3573
Root mean squared error 0.4376
Relative absolute error 72.9512 %
Root relative squared error 88.6612 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.799 0.379 0.746 0.799 0.772 0.428 0.744
0.621 0.201 0.689 0.621 0.653 0.428 0.744
Weighted Avg. 0.725 0.305 0.722 0.725 0.722 0.428 0.744
=== Confusion Matrix ===
a b <-- classified as
147 37 | a = Paid
50 82 | b = Defaulted

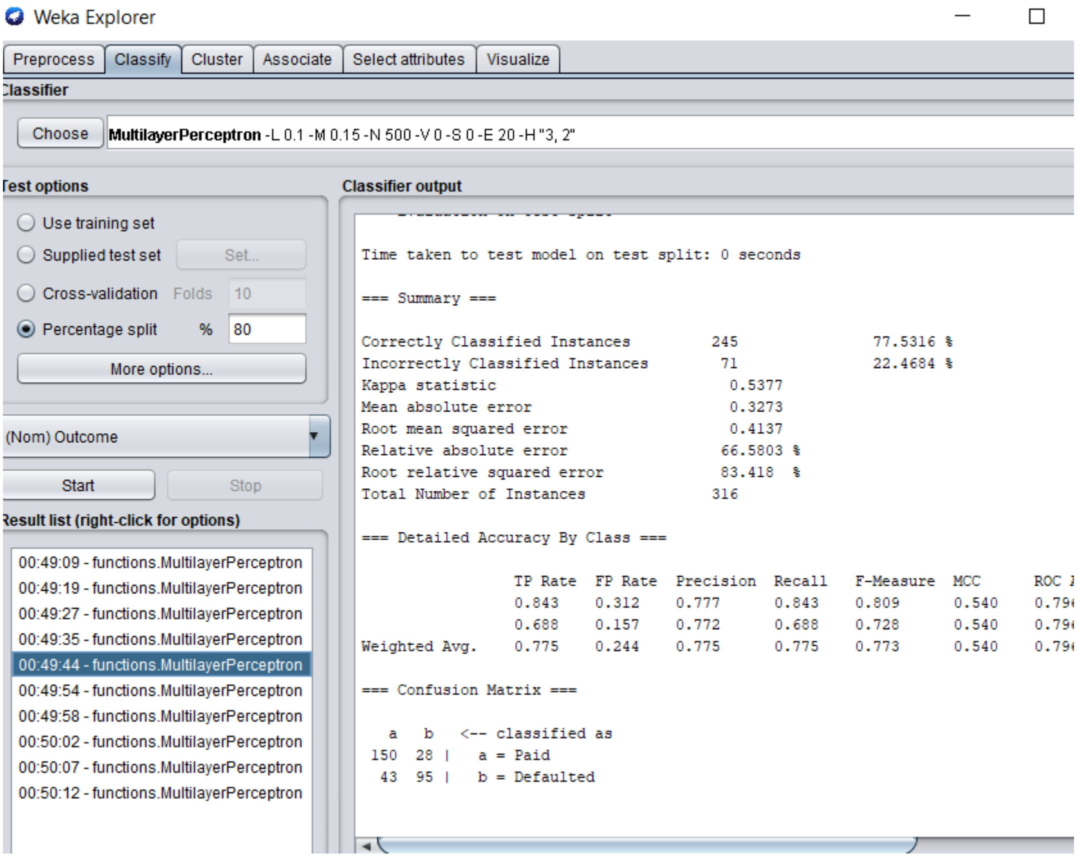
MLP models - %split before feature selection (hold out method)

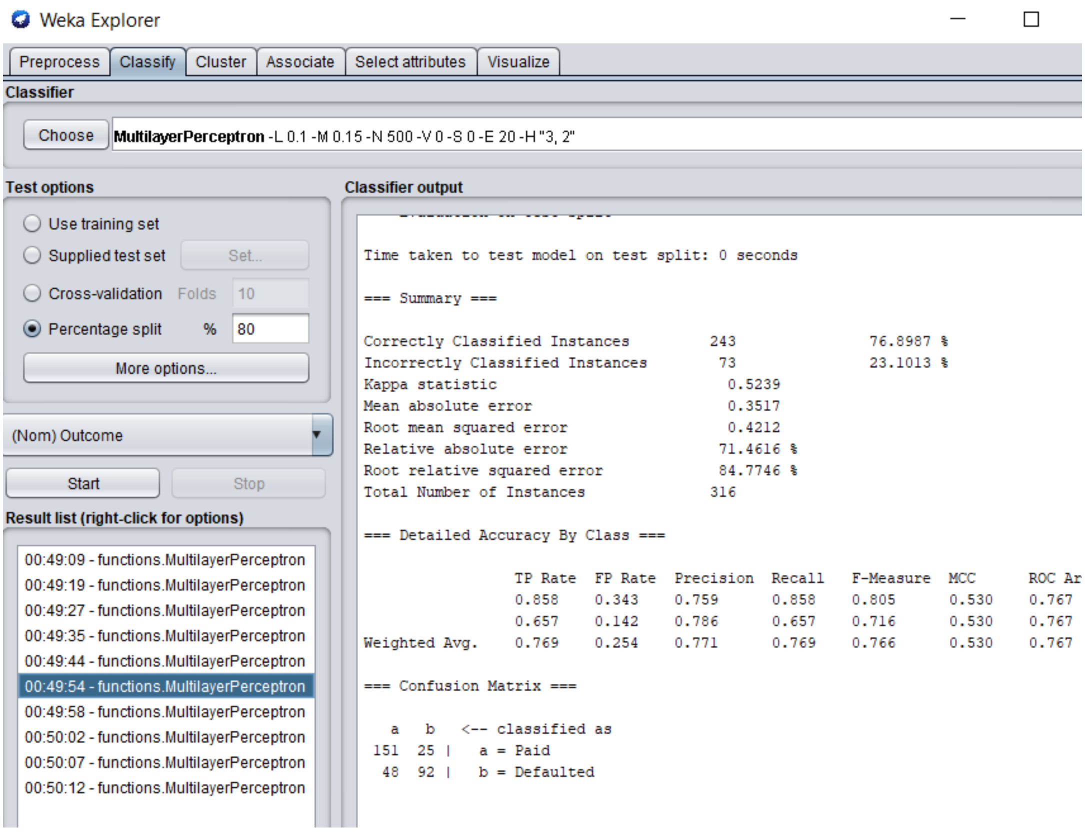


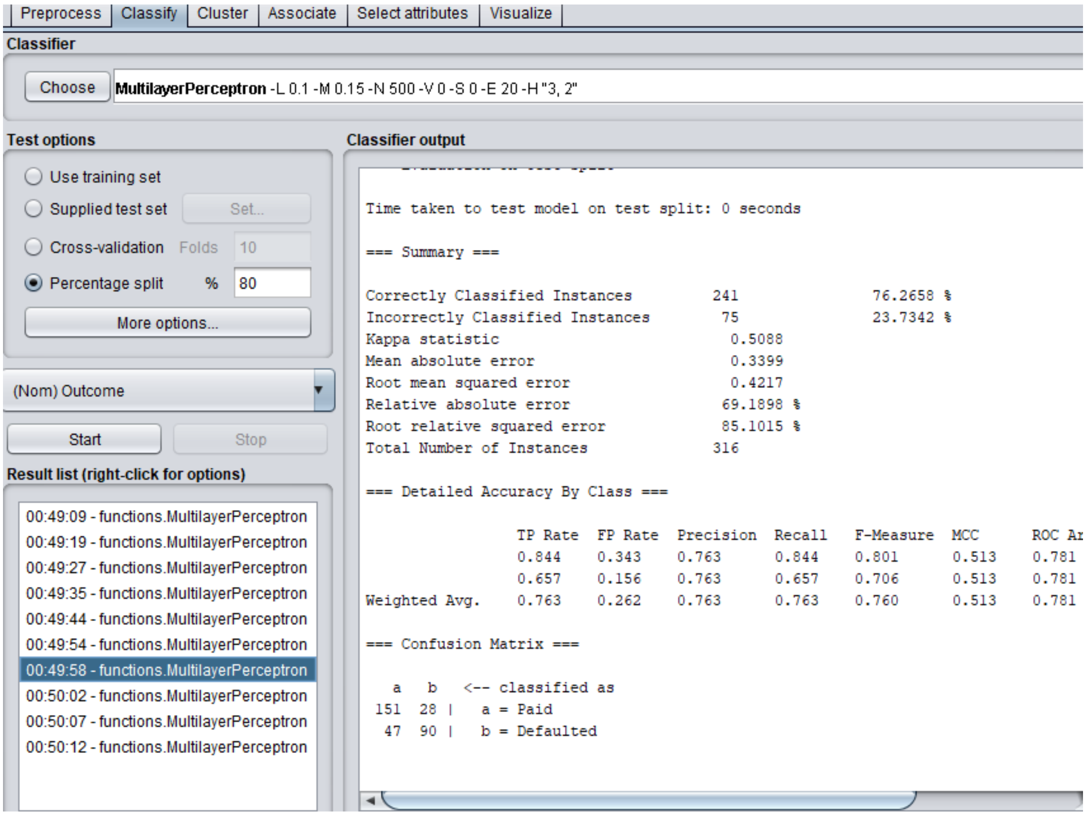


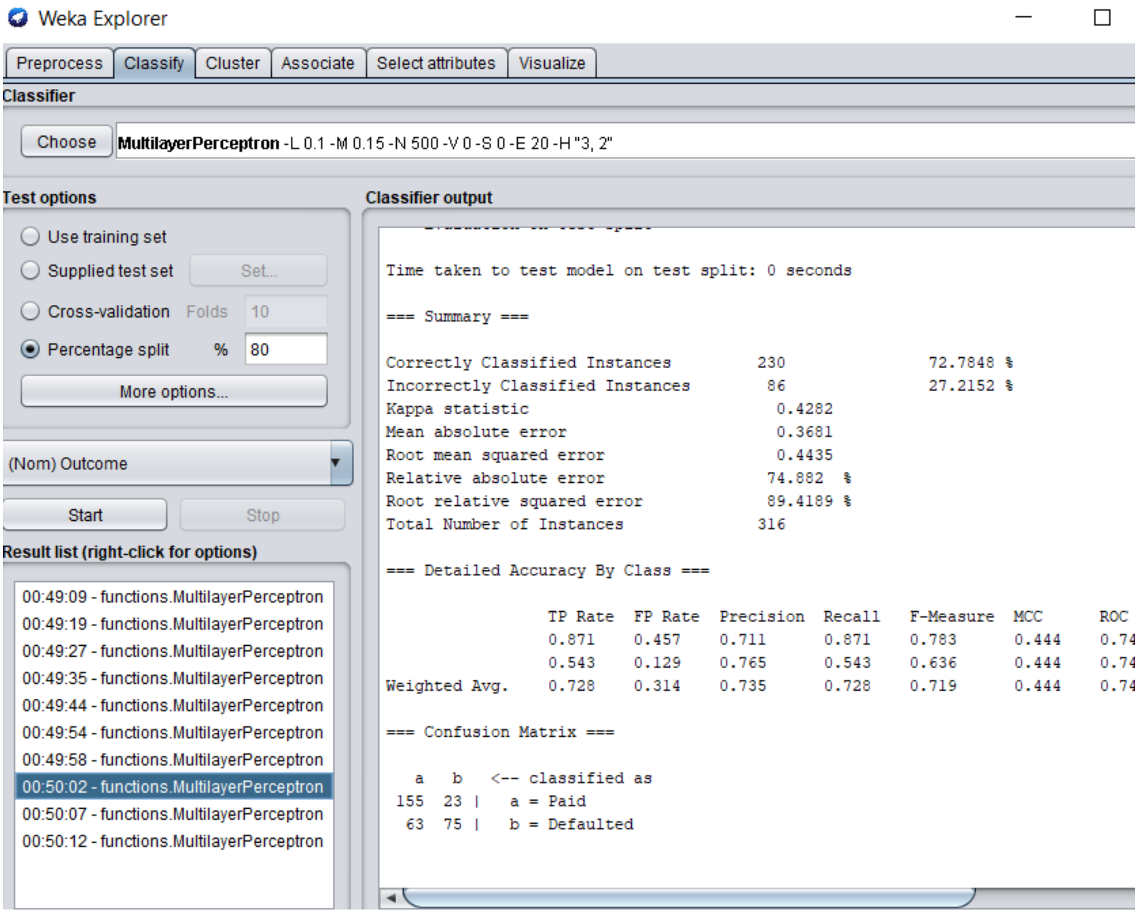

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 241 76.2658 %
Incorrectly Classified Instances 75 23.7342 %
Kappa statistic 0.4952
Mean absolute error 0.3431
Root mean squared error 0.4272
Relative absolute error 70.229 %
Root relative squared error 86.8558 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.847 0.362 0.777 0.847 0.810 0.499 0.75
0.638 0.153 0.736 0.638 0.684 0.499 0.75
Weighted Avg. 0.763 0.278 0.760 0.763 0.759 0.499 0.75
=== Confusion Matrix ===
a b <-- classified as
160 29 | a = Paid
46 81 | b = Defaulted

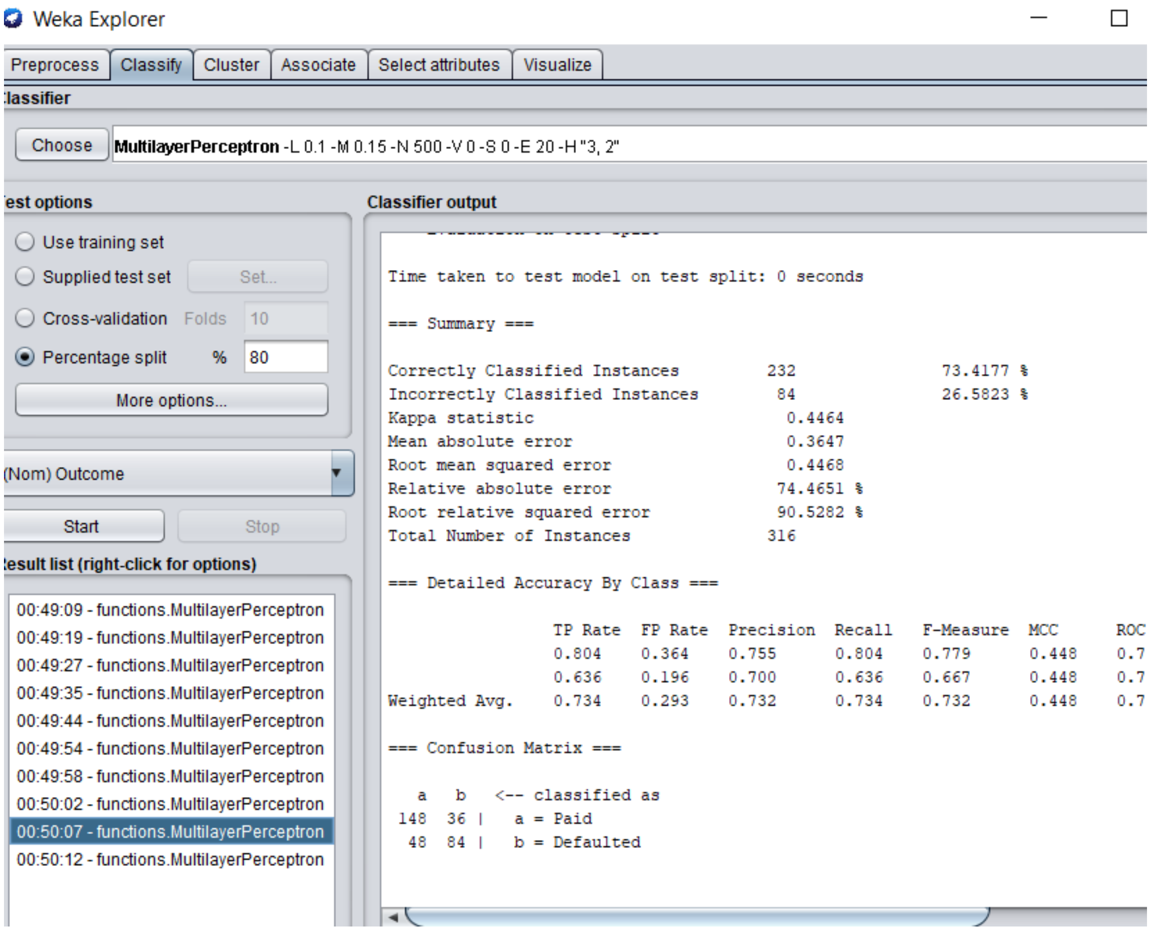

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 240 75.9494 %
Incorrectly Classified Instances 76 24.0506 %
Kappa statistic 0.5145
Mean absolute error 0.3455
Root mean squared error 0.4268
Relative absolute error 69.57 %
Root relative squared error 84.9208 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.848 0.338 0.733 0.848 0.787 0.522 0.7
0.662 0.152 0.800 0.662 0.725 0.522 0.7
Weighted Avg. 0.759 0.249 0.765 0.759 0.757 0.522 0.7
=== Confusion Matrix ===
a b <-- classified as
140 25 | a = Paid
51 100 | b = Defaulted

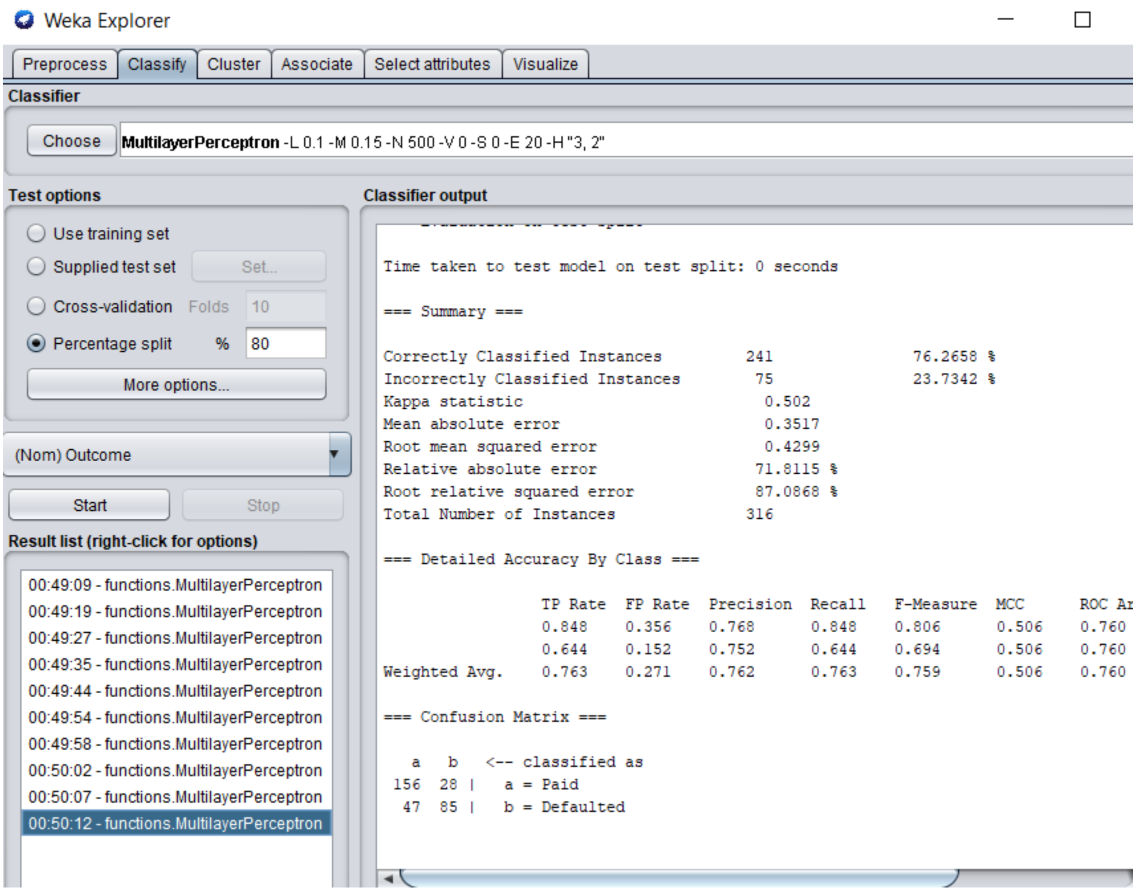

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 245 77.5316 %
Incorrectly Classified Instances 71 22.4684 %
Kappa statistic 0.5377
Mean absolute error 0.3273
Root mean squared error 0.4137
Relative absolute error 66.5803 %
Root relative squared error 83.418 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.843 0.312 0.777 0.843 0.809 0.540
0.688 0.157 0.772 0.688 0.728 0.540
Weighted Avg. 0.775 0.244 0.775 0.775 0.773 0.540
=== Confusion Matrix ===
a b <-- classified as
150 28 | a = Paid
43 95 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 243 76.8987 %
Incorrectly Classified Instances 73 23.1013 %
Kappa statistic 0.5239
Mean absolute error 0.3517
Root mean squared error 0.4212
Relative absolute error 71.4616 %
Root relative squared error 84.7746 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.858 0.343 0.759 0.858 0.805 0.530 0.767
0.657 0.142 0.786 0.657 0.716 0.530 0.767
Weighted Avg. 0.769 0.254 0.771 0.769 0.766 0.530 0.767
=== Confusion Matrix ===
a b <-- classified as
151 25 | a = Paid
48 92 | b = Defaulted


Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 241 76.2658 %
Incorrectly Classified Instances 75 23.7342 %
Kappa statistic 0.5088
Mean absolute error 0.3399
Root mean squared error 0.4217
Relative absolute error 69.1898 %
Root relative squared error 85.1015 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Ar
0.844 0.343 0.763 0.844 0.801 0.513 0.781
0.657 0.156 0.763 0.657 0.706 0.513 0.781
Weighted Avg. 0.763 0.262 0.763 0.763 0.760 0.513 0.781
=== Confusion Matrix ===
a b <-- classified as
151 28 | a = Paid
47 90 | b = Defaulted

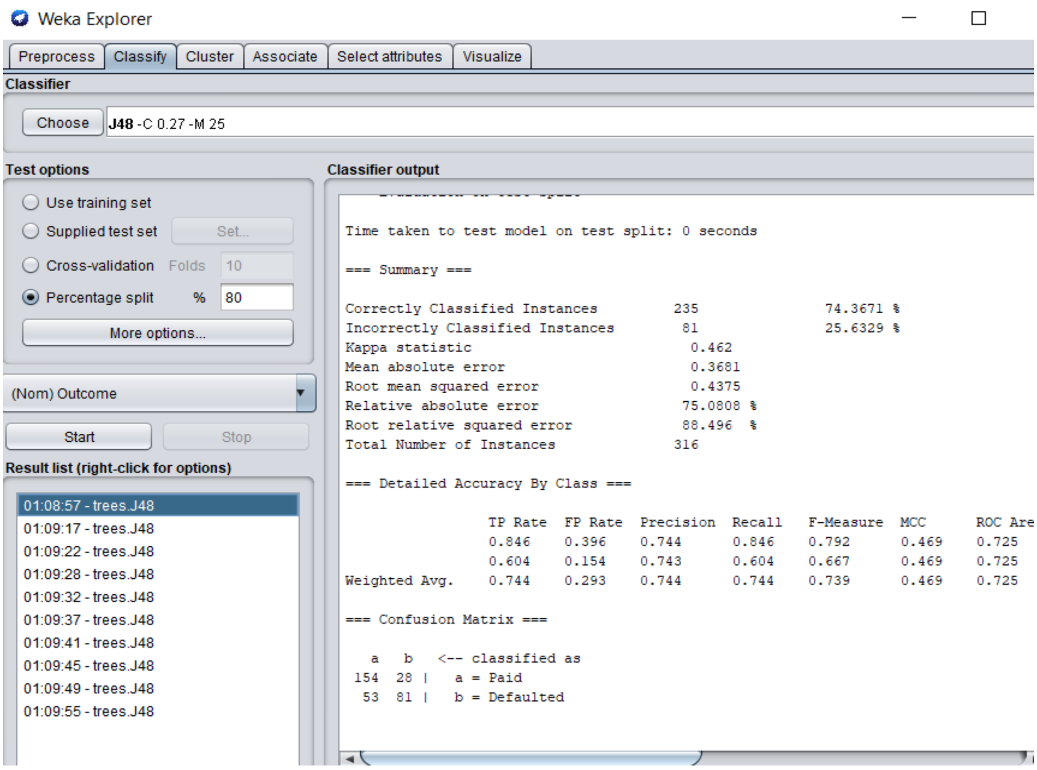

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 230 72.7848 %
Incorrectly Classified Instances 86 27.2152 %
Kappa statistic 0.4282
Mean absolute error 0.3681
Root mean squared error 0.4435
Relative absolute error 74.882 %
Root relative squared error 89.4189 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.871 0.457 0.711 0.871 0.783 0.444 0.74
0.543 0.129 0.765 0.543 0.636 0.444 0.74
Weighted Avg. 0.728 0.314 0.735 0.728 0.719 0.444 0.74
=== Confusion Matrix ===
a b <-- classified as
155 23 | a = Paid
63 75 | b = Defaulted

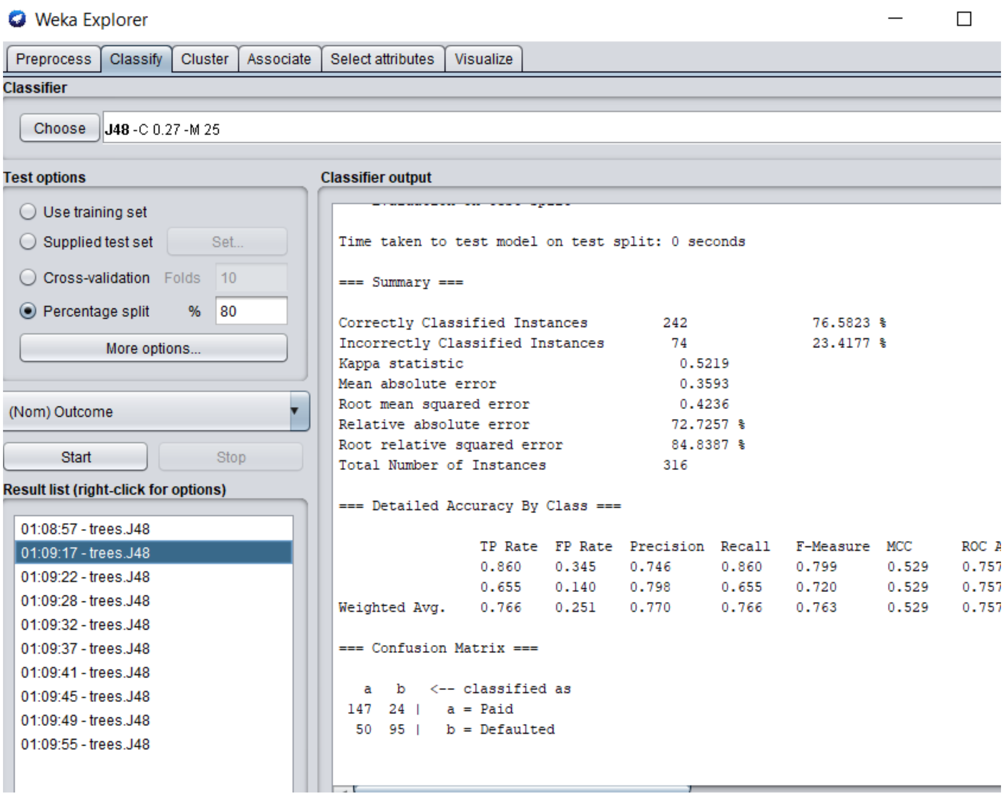

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
lassifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
est options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
esult list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 232 73.4177 %
Incorrectly Classified Instances 84 26.5823 %
Kappa statistic 0.4464
Mean absolute error 0.3647
Root mean squared error 0.4468
Relative absolute error 74.4651 %
Root relative squared error 90.5282 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.804 0.364 0.755 0.804 0.779 0.448 0.7
0.636 0.196 0.700 0.636 0.667 0.448 0.7
Weighted Avg. 0.734 0.293 0.732 0.734 0.732 0.448 0.7
=== Confusion Matrix ===
a b <-- classified as
148 36 | a = Paid
48 84 | b = Defaulted

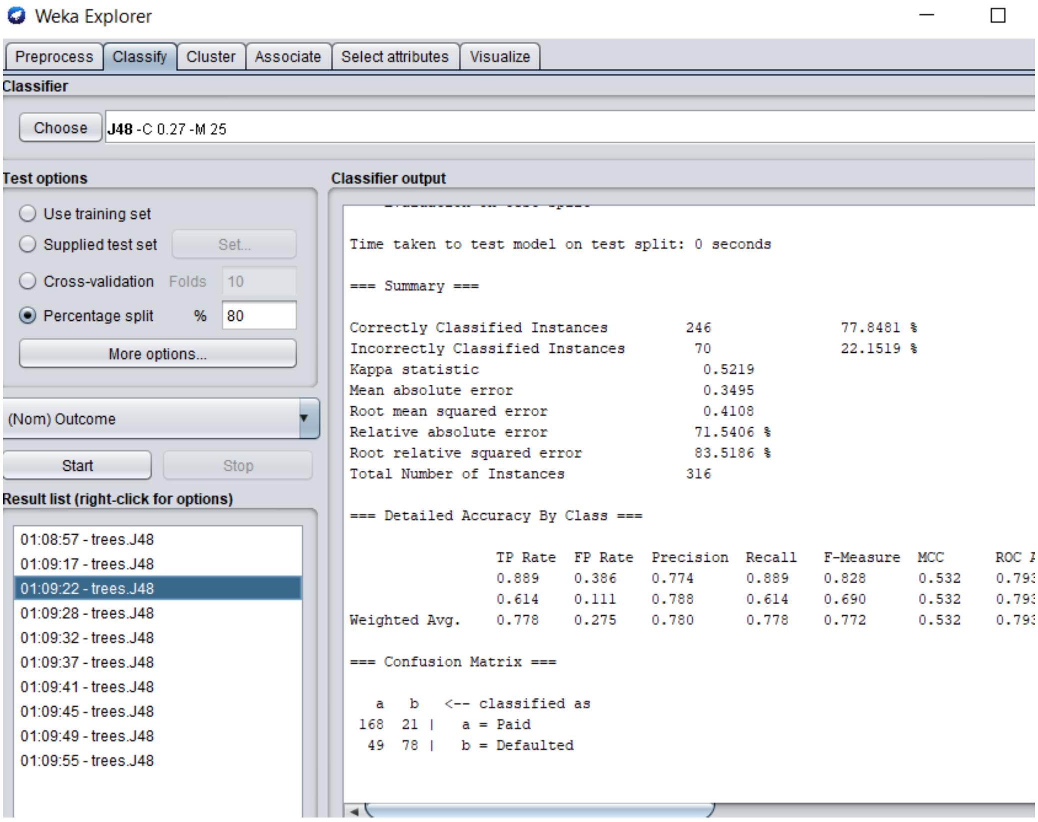

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
00:49:09 - functions.MultilayerPerceptron
00:49:19 - functions.MultilayerPerceptron
00:49:27 - functions.MultilayerPerceptron
00:49:35 - functions.MultilayerPerceptron
00:49:44 - functions.MultilayerPerceptron
00:49:54 - functions.MultilayerPerceptron
00:49:58 - functions.MultilayerPerceptron
00:50:02 - functions.MultilayerPerceptron
00:50:07 - functions.MultilayerPerceptron
00:50:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 241 76.2658 %
Incorrectly Classified Instances 75 23.7342 %
Kappa statistic 0.502
Mean absolute error 0.3517
Root mean squared error 0.4299
Relative absolute error 71.8115 %
Root relative squared error 87.0868 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Ar
0.848 0.356 0.768 0.848 0.806 0.506 0.760
0.644 0.152 0.752 0.644 0.694 0.506 0.760
Weighted Avg. 0.763 0.271 0.762 0.763 0.759 0.506 0.760
=== Confusion Matrix ===
a b <-- classified as
156 28 | a = Paid
47 85 | b = Defaulted

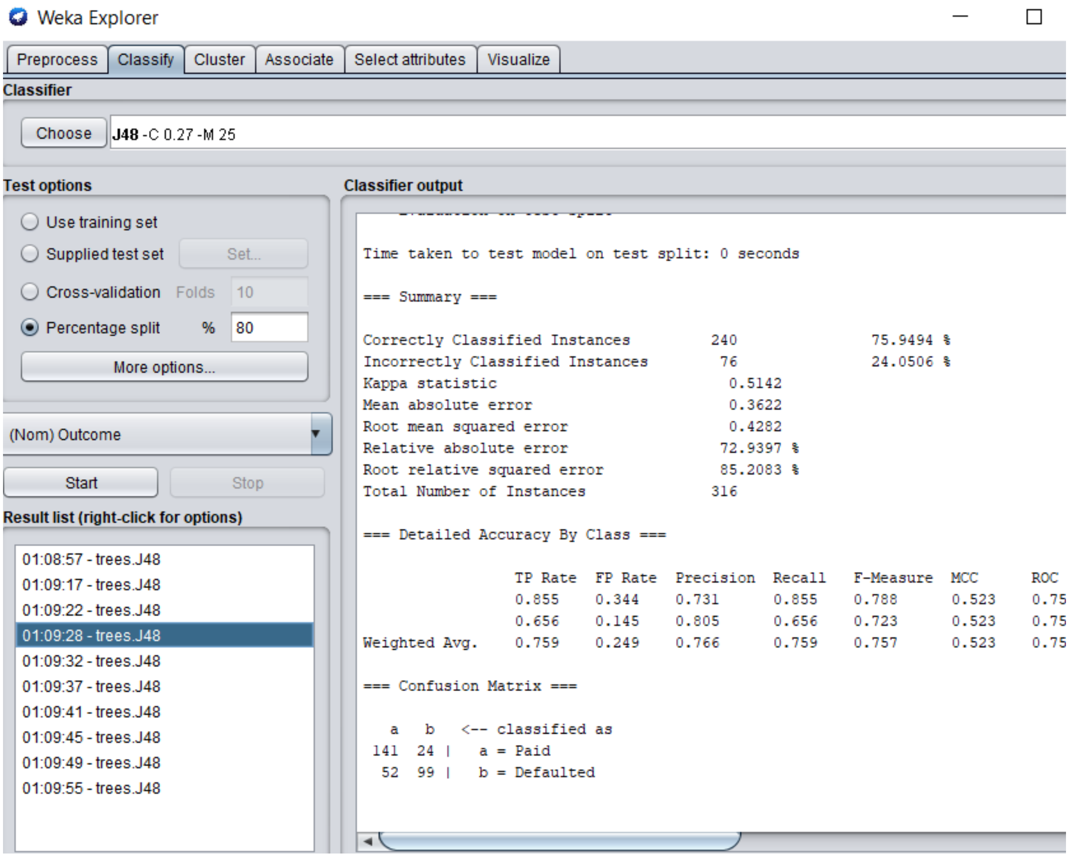
Appendix C

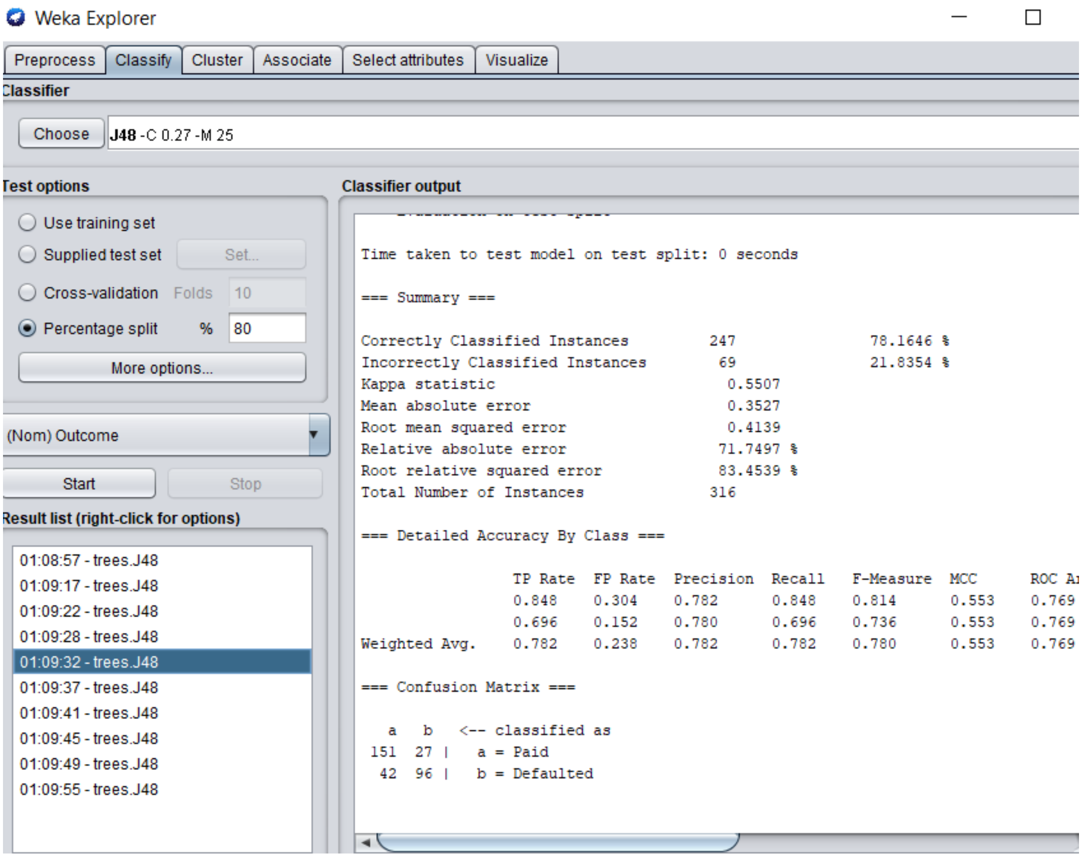
J48 - % split after Feature selection (Hold out method)

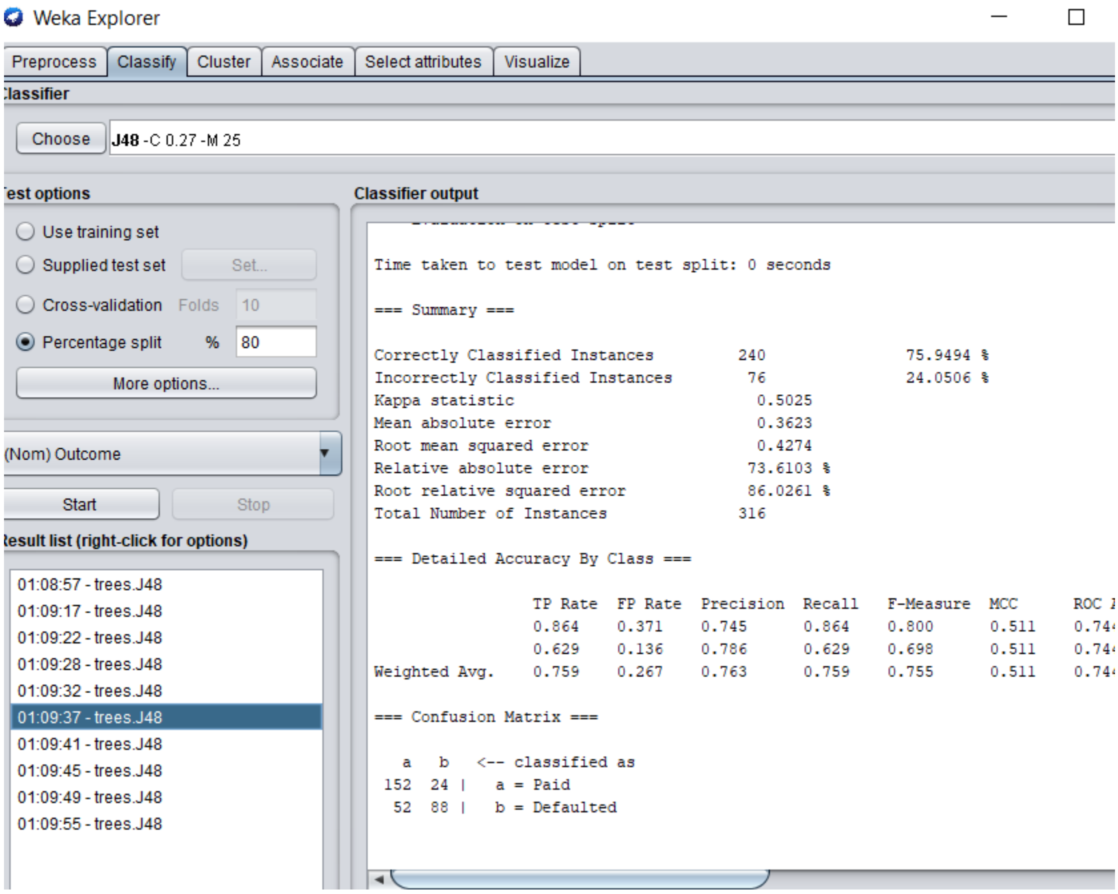


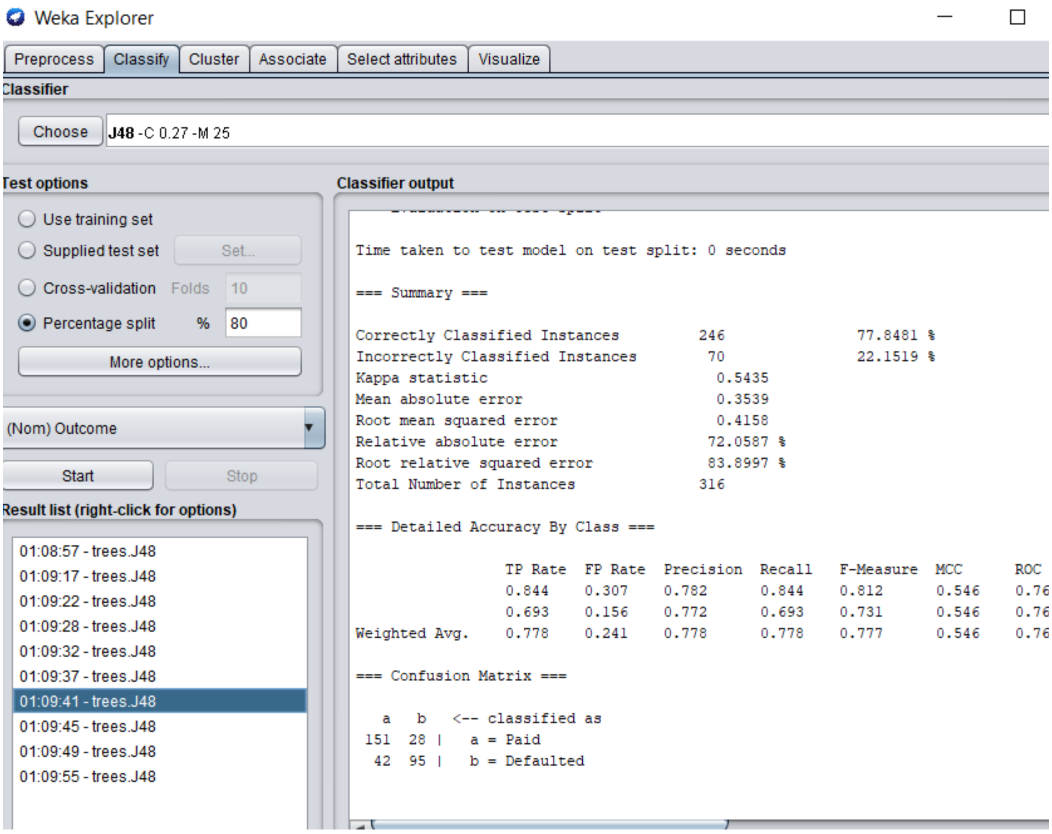


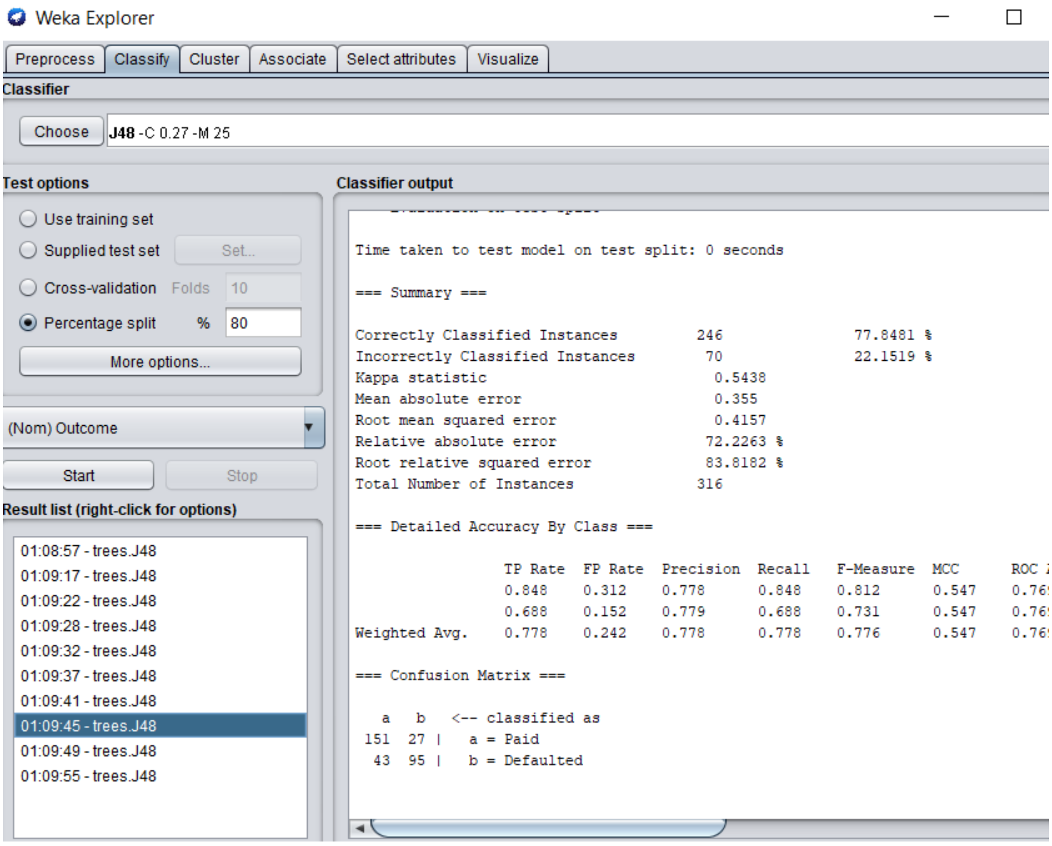

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 246 77.8481 %
Incorrectly Classified Instances 70 22.1519 %
Kappa statistic 0.5219
Mean absolute error 0.3495
Root mean squared error 0.4108
Relative absolute error 71.5406 %
Root relative squared error 83.5186 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.889 0.386 0.774 0.889 0.828 0.532
0.614 0.111 0.788 0.614 0.690 0.532
Weighted Avg. 0.778 0.275 0.780 0.778 0.772 0.532
=== Confusion Matrix ===
a b <-- classified as
168 21 | a = Paid
49 78 | b = Defaulted

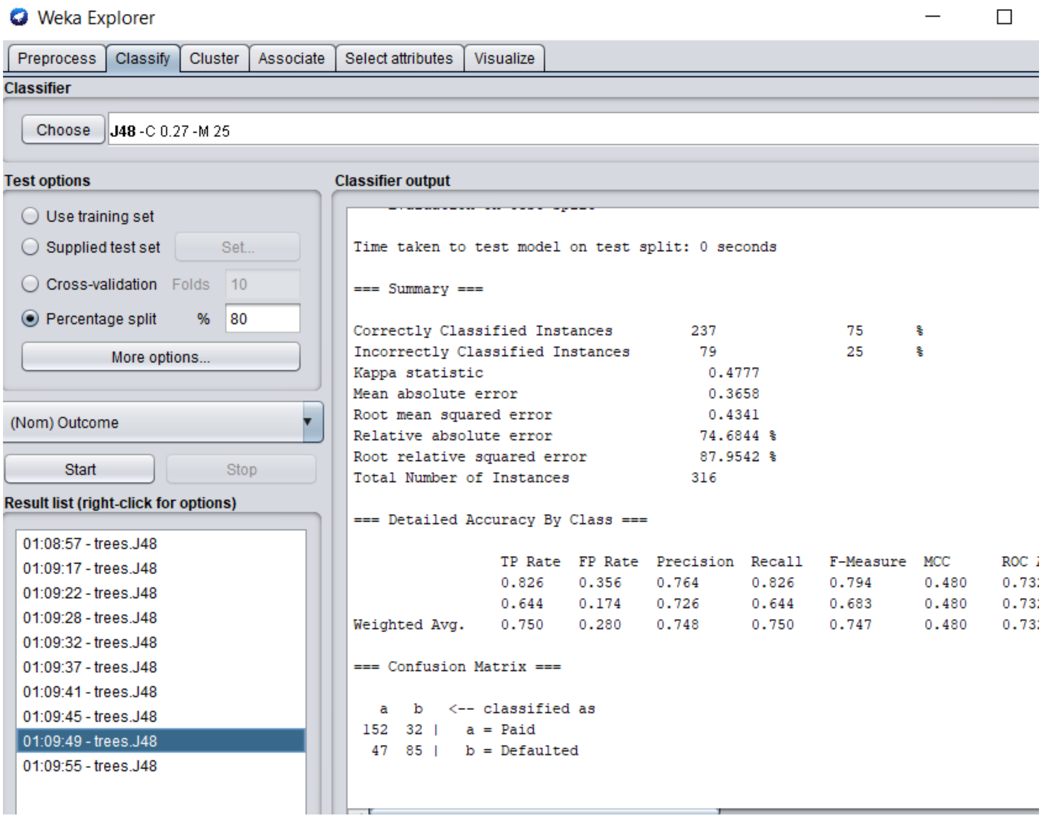

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 240 75.9494 %
Incorrectly Classified Instances 76 24.0506 %
Kappa statistic 0.5142
Mean absolute error 0.3622
Root mean squared error 0.4282
Relative absolute error 72.9397 %
Root relative squared error 85.2083 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.855 0.344 0.731 0.855 0.788 0.523 0.75
0.656 0.145 0.805 0.656 0.723 0.523 0.75
Weighted Avg. 0.759 0.249 0.766 0.759 0.757 0.523 0.75
=== Confusion Matrix ===
a b <-- classified as
141 24 | a = Paid
52 99 | b = Defaulted

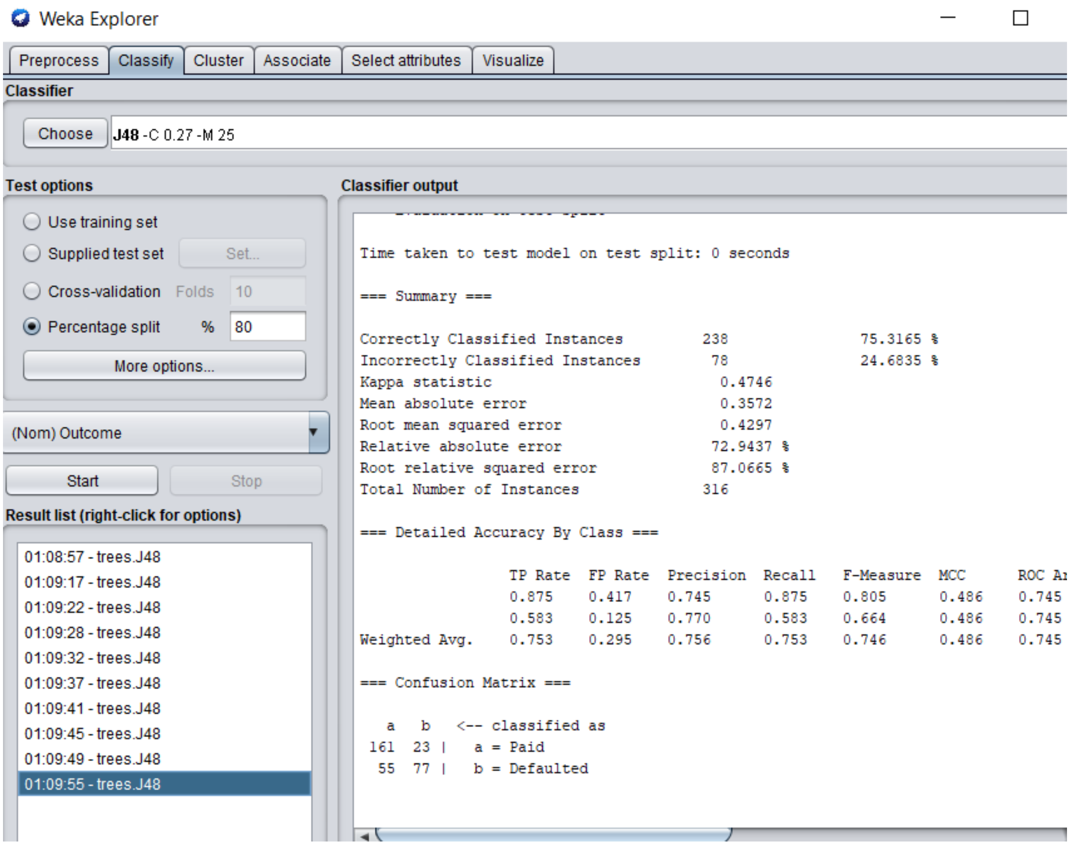

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 247 78.1646 %
Incorrectly Classified Instances 69 21.8354 %
Kappa statistic 0.5507
Mean absolute error 0.3527
Root mean squared error 0.4139
Relative absolute error 71.7497 %
Root relative squared error 83.4539 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.848 0.304 0.782 0.848 0.814 0.553 0.769
0.696 0.152 0.780 0.696 0.736 0.553 0.769
Weighted Avg. 0.782 0.238 0.782 0.782 0.780 0.553 0.769
=== Confusion Matrix ===
a b <-- classified as
151 27 | a = Paid
42 96 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Choose
J48 -C 0.27 -M 25
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 240 75.9494 %
Incorrectly Classified Instances 76 24.0506 %
Kappa statistic 0.5025
Mean absolute error 0.3623
Root mean squared error 0.4274
Relative absolute error 73.6103 %
Root relative squared error 86.0261 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.864 0.371 0.745 0.864 0.800 0.511
0.629 0.136 0.786 0.629 0.698 0.511
Weighted Avg. 0.759 0.267 0.763 0.759 0.755 0.511
=== Confusion Matrix ===
a b <-- classified as
152 24 | a = Paid
52 88 | b = Defaulted

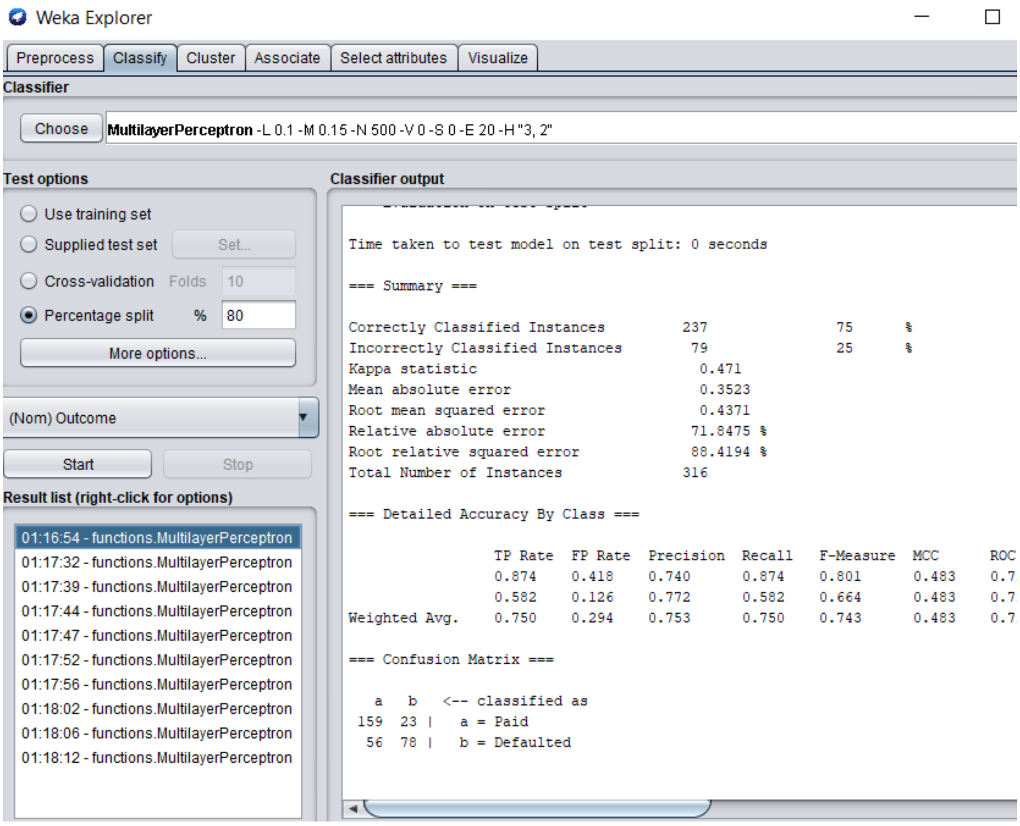

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 246 77.8481 %
Incorrectly Classified Instances 70 22.1519 %
Kappa statistic 0.5435
Mean absolute error 0.3539
Root mean squared error 0.4158
Relative absolute error 72.0587 %
Root relative squared error 83.8997 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.844 0.307 0.782 0.844 0.812 0.546 0.76
0.693 0.156 0.772 0.693 0.731 0.546 0.76
Weighted Avg. 0.778 0.241 0.778 0.778 0.777 0.546 0.76
=== Confusion Matrix ===
a b <-- classified as
151 28 | a = Paid
42 95 | b = Defaulted

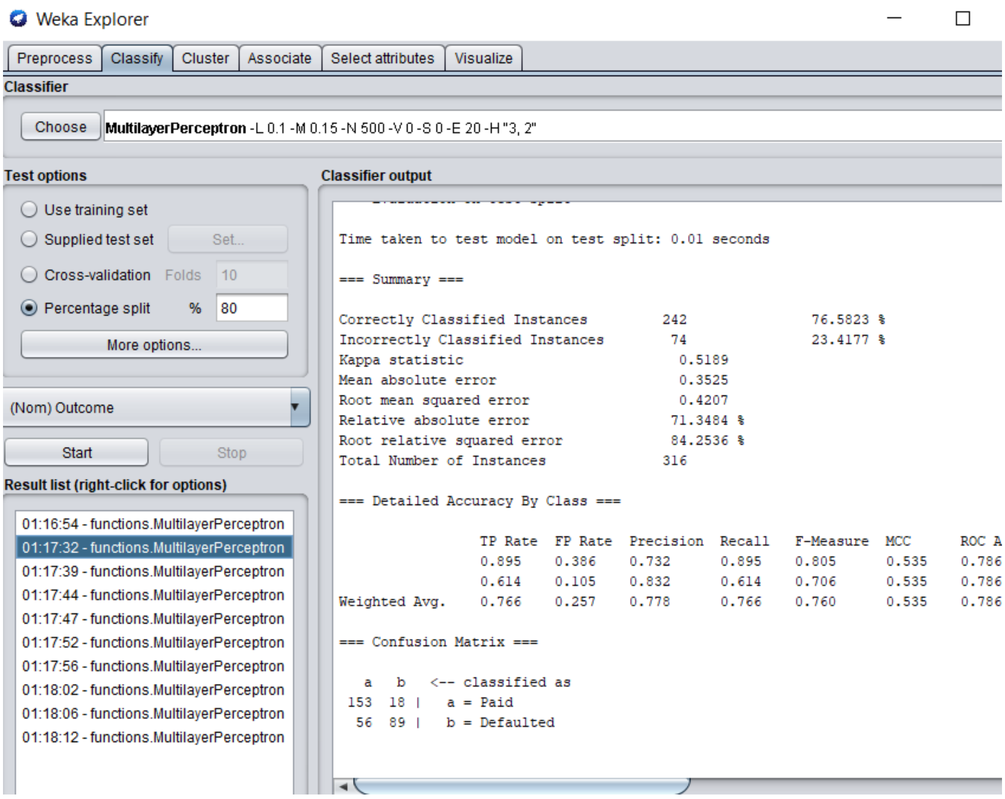

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 246 77.8481 %
Incorrectly Classified Instances 70 22.1519 %
Kappa statistic 0.5438
Mean absolute error 0.355
Root mean squared error 0.4157
Relative absolute error 72.2263 %
Root relative squared error 83.8182 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.848 0.312 0.778 0.848 0.812 0.547
0.688 0.152 0.779 0.688 0.731 0.547
Weighted Avg. 0.778 0.242 0.778 0.778 0.776 0.547
=== Confusion Matrix ===
a b <-- classified as
151 27 | a = Paid
43 95 | b = Defaulted

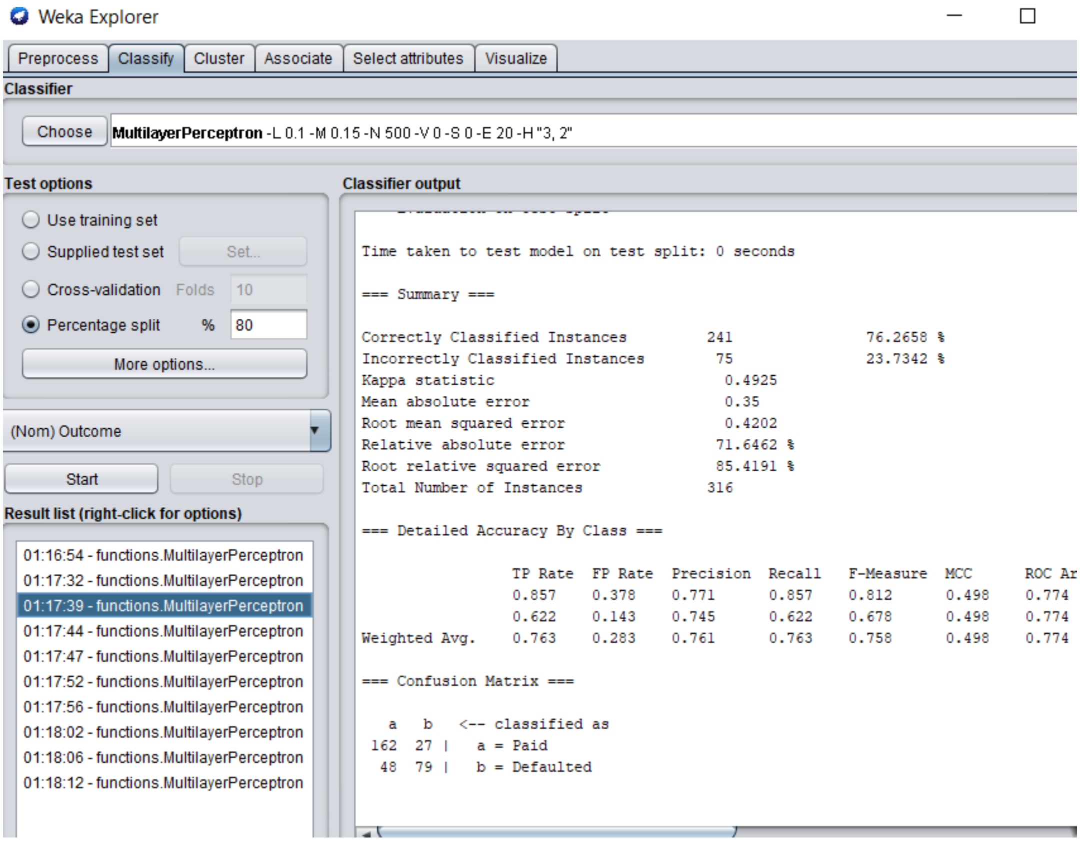

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 237 75 %
Incorrectly Classified Instances 79 25 %
Kappa statistic 0.4777
Mean absolute error 0.3658
Root mean squared error 0.4341
Relative absolute error 74.6844 %
Root relative squared error 87.9542 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.826 0.356 0.764 0.826 0.794 0.480
0.644 0.174 0.726 0.644 0.683 0.480
Weighted Avg. 0.750 0.280 0.748 0.750 0.747 0.480
=== Confusion Matrix ===
a b <-- classified as
152 32 | a = Paid
47 85 | b = Defaulted

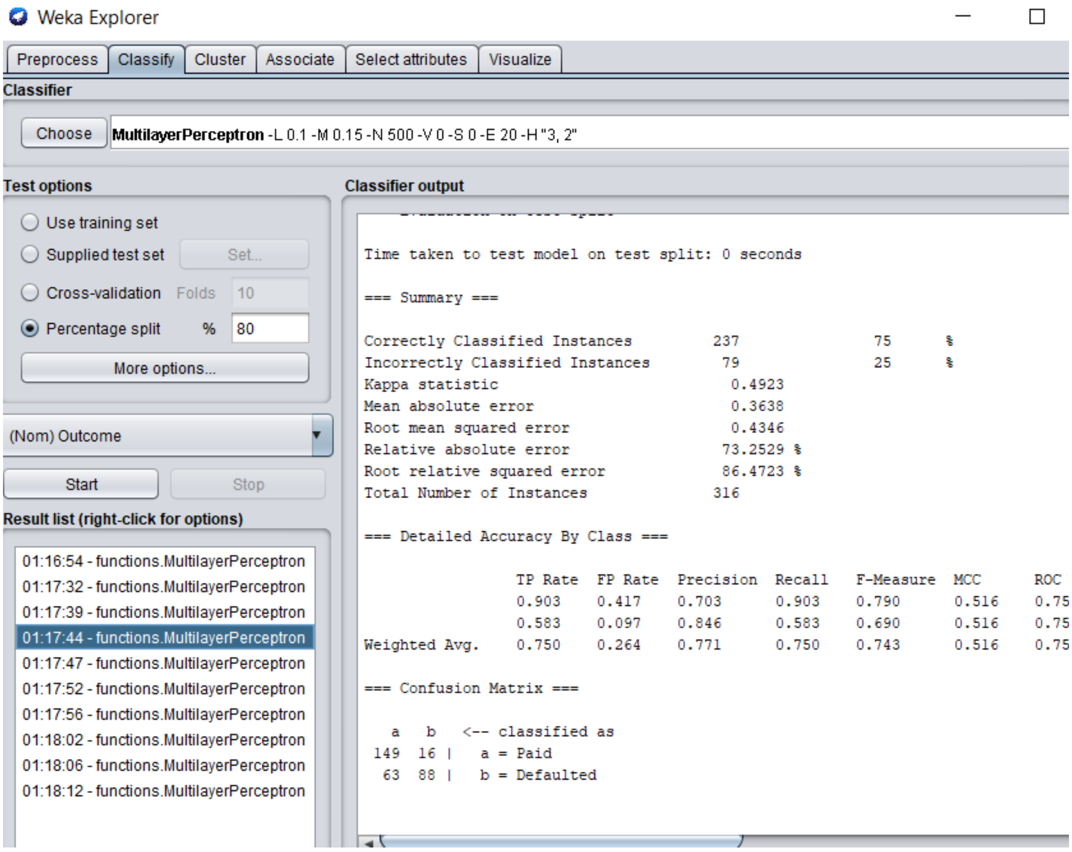

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.27 -M 25
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:08:57 - trees.J48
01:09:17 - trees.J48
01:09:22 - trees.J48
01:09:28 - trees.J48
01:09:32 - trees.J48
01:09:37 - trees.J48
01:09:41 - trees.J48
01:09:45 - trees.J48
01:09:49 - trees.J48
01:09:55 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 238 75.3165 %
Incorrectly Classified Instances 78 24.6835 %
Kappa statistic 0.4746
Mean absolute error 0.3572
Root mean squared error 0.4297
Relative absolute error 72.9437 %
Root relative squared error 87.0665 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.875 0.417 0.745 0.875 0.805 0.486 0.745
0.583 0.125 0.770 0.583 0.664 0.486 0.745
Weighted Avg. 0.753 0.295 0.756 0.753 0.746 0.486 0.745
=== Confusion Matrix ===
a b <-- classified as
161 23 | a = Paid
55 77 | b = Defaulted

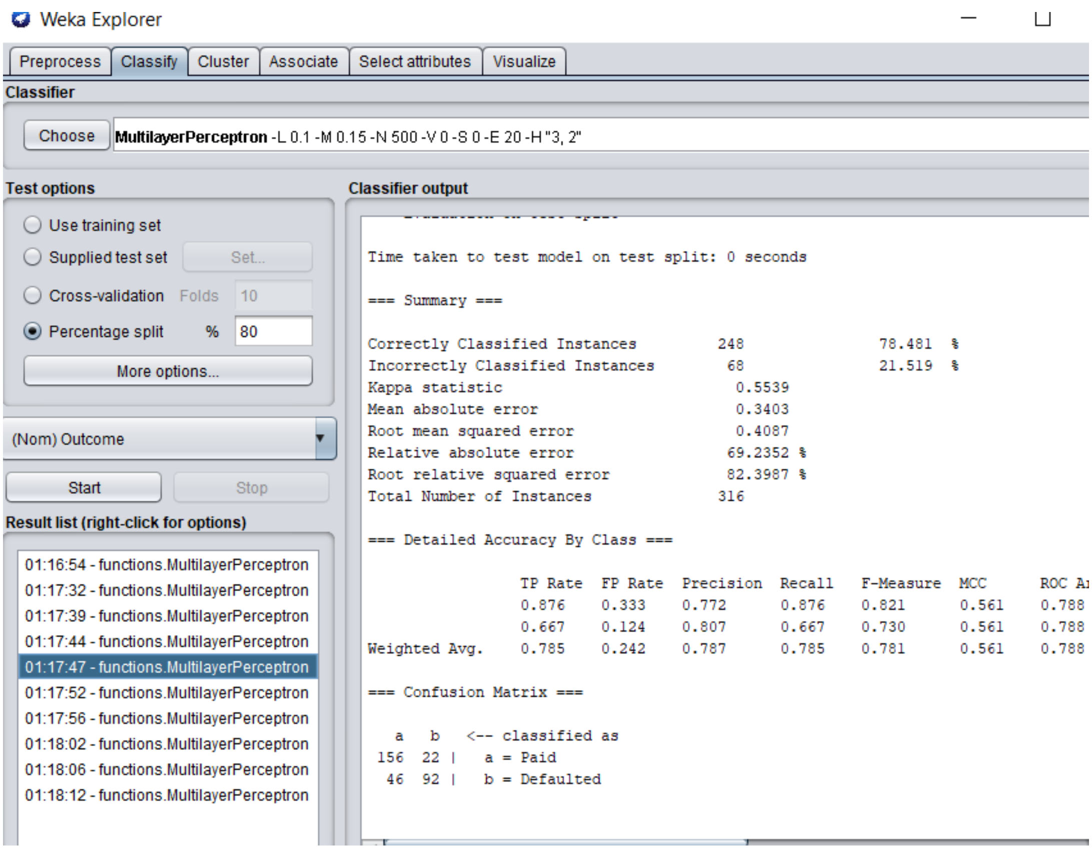
MLP - % split after Feature selection (Hold out method)

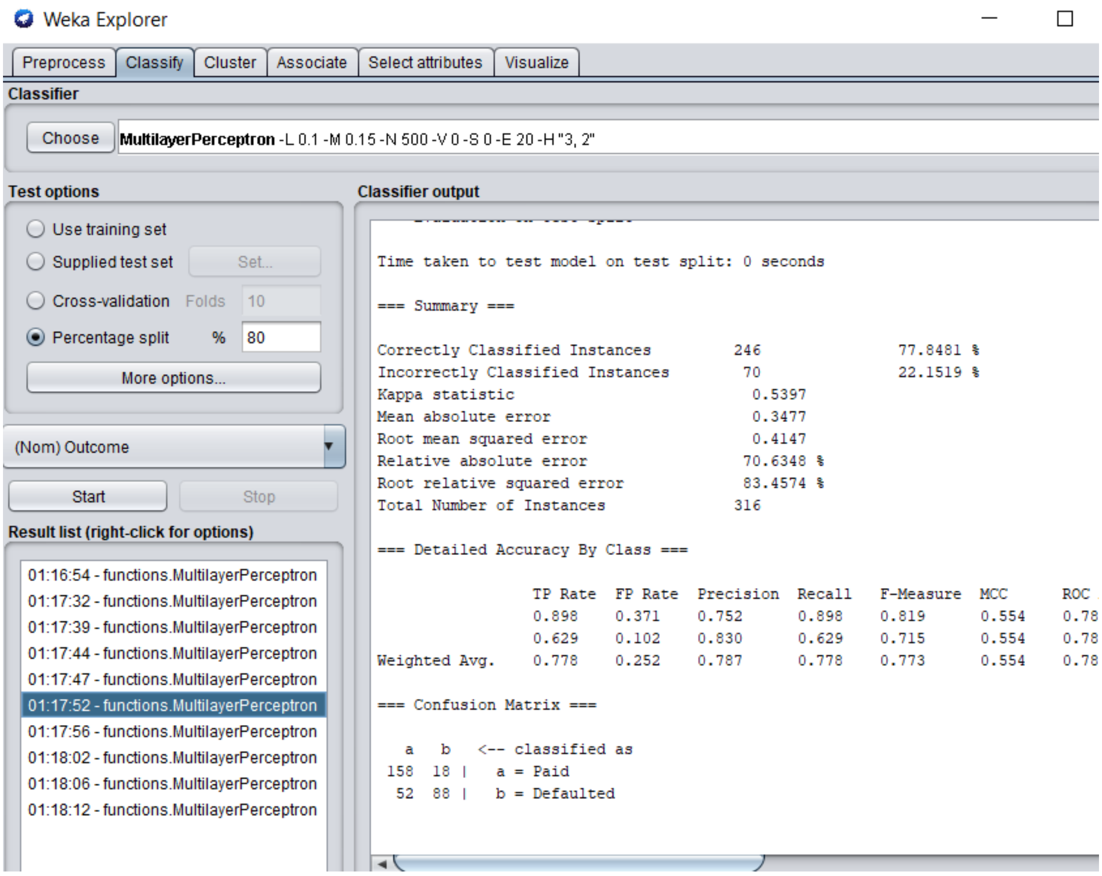


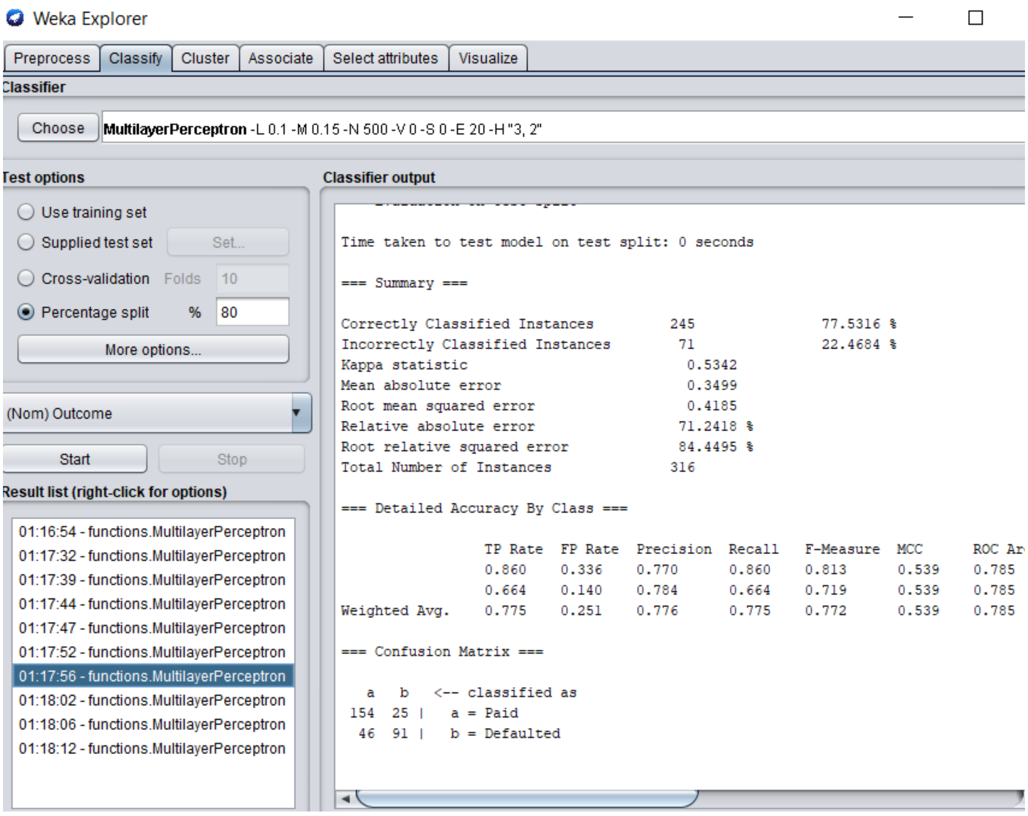


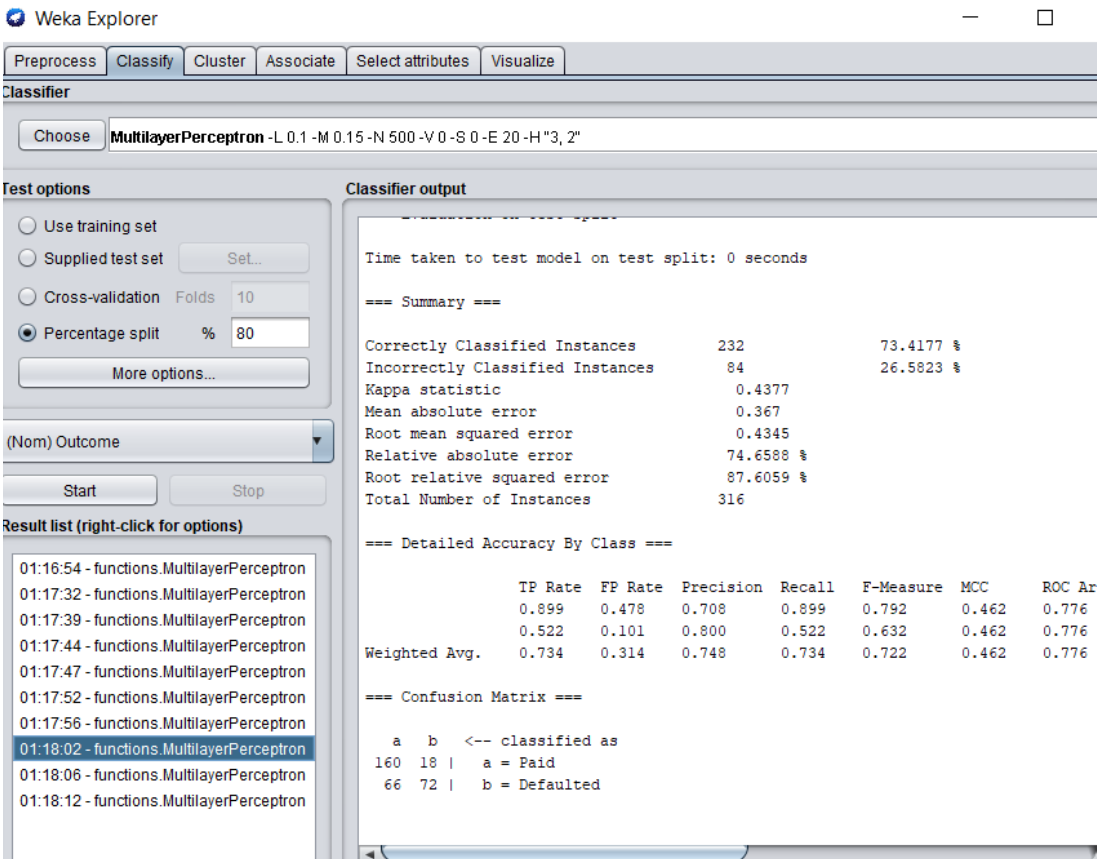

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 241 76.2658 %
Incorrectly Classified Instances 75 23.7342 %
Kappa statistic 0.4925
Mean absolute error 0.35
Root mean squared error 0.4202
Relative absolute error 71.6462 %
Root relative squared error 85.4191 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Ar
0.857 0.378 0.771 0.857 0.812 0.498 0.774
0.622 0.143 0.745 0.622 0.678 0.498 0.774
Weighted Avg. 0.763 0.283 0.761 0.763 0.758 0.498 0.774
=== Confusion Matrix ===
a b <-- classified as
162 27 | a = Paid
48 79 | b = Defaulted

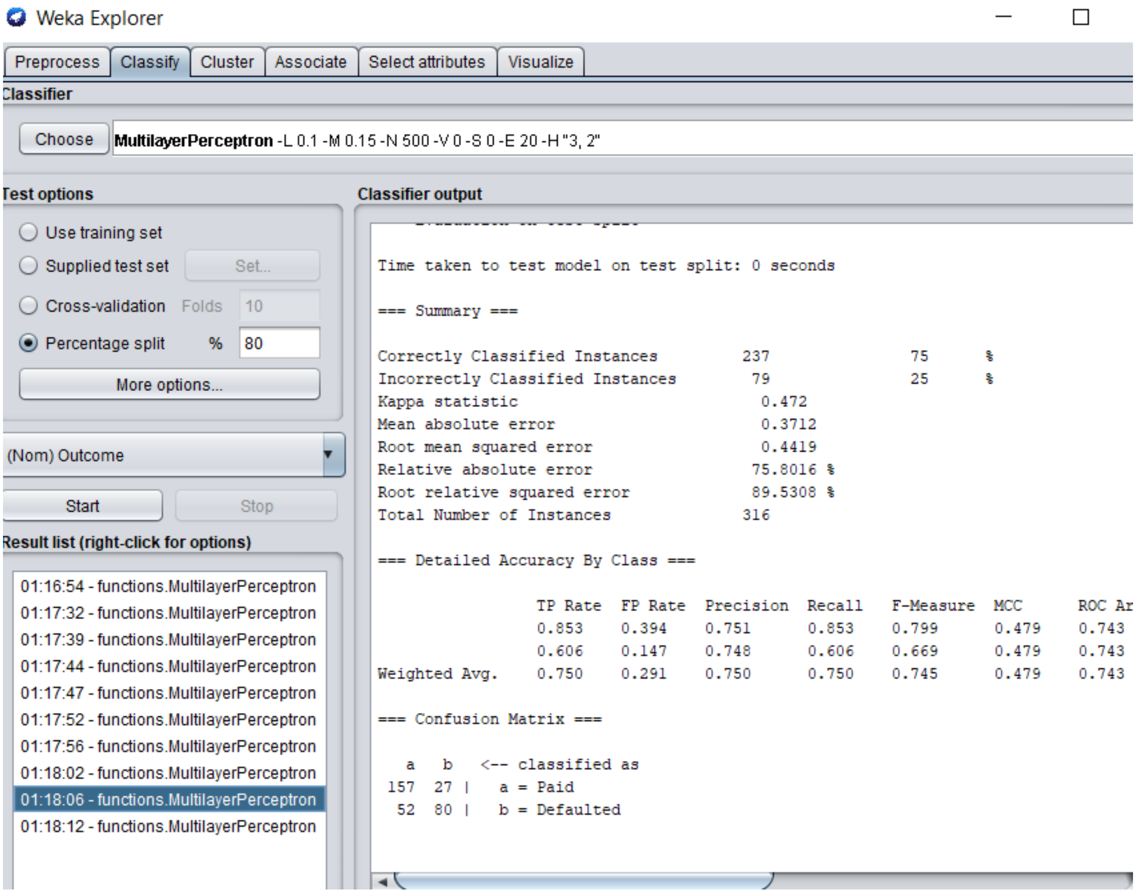

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 237 75 %
Incorrectly Classified Instances 79 25 %
Kappa statistic 0.4923
Mean absolute error 0.3638
Root mean squared error 0.4346
Relative absolute error 73.2529 %
Root relative squared error 86.4723 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.903 0.417 0.703 0.903 0.790 0.516 0.75
0.583 0.097 0.846 0.583 0.690 0.516 0.75
Weighted Avg. 0.750 0.264 0.771 0.750 0.743 0.516 0.75
=== Confusion Matrix ===
a b <-- classified as
149 16 | a = Paid
63 88 | b = Defaulted

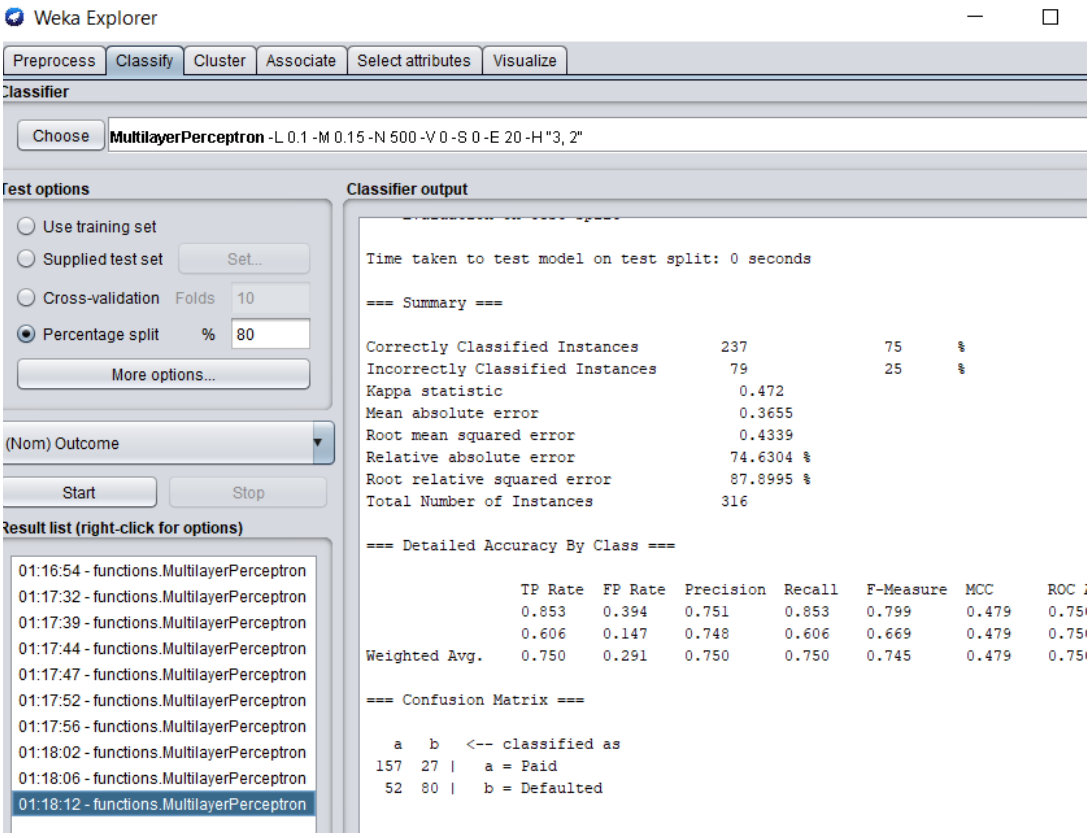

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 248 78.481 %
Incorrectly Classified Instances 68 21.519 %
Kappa statistic 0.5539
Mean absolute error 0.3403
Root mean squared error 0.4087
Relative absolute error 69.2352 %
Root relative squared error 82.3987 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC A
0.876 0.333 0.772 0.876 0.821 0.561 0.788
0.667 0.124 0.807 0.667 0.730 0.561 0.788
Weighted Avg. 0.785 0.242 0.787 0.785 0.781 0.561 0.788
=== Confusion Matrix ===
a b <-- classified as
156 22 | a = Paid
46 92 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 246 77.8481 %
Incorrectly Classified Instances 70 22.1519 %
Kappa statistic 0.5397
Mean absolute error 0.3477
Root mean squared error 0.4147
Relative absolute error 70.6348 %
Root relative squared error 83.4574 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC
0.898 0.371 0.752 0.898 0.819 0.554 0.78
0.629 0.102 0.830 0.629 0.715 0.554 0.78
Weighted Avg. 0.778 0.252 0.787 0.778 0.773 0.554 0.78
=== Confusion Matrix ===
a b <-- classified as
158 18 | a = Paid
52 88 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 245 77.5316 %
Incorrectly Classified Instances 71 22.4684 %
Kappa statistic 0.5342
Mean absolute error 0.3499
Root mean squared error 0.4185
Relative absolute error 71.2418 %
Root relative squared error 84.4495 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Ar
0.860 0.336 0.770 0.860 0.813 0.539 0.785
0.664 0.140 0.784 0.664 0.719 0.539 0.785
Weighted Avg. 0.775 0.251 0.776 0.775 0.772 0.539 0.785
=== Confusion Matrix ===
a b <-- classified as
154 25 | a = Paid
46 91 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 232 73.4177 %
Incorrectly Classified Instances 84 26.5823 %
Kappa statistic 0.4377
Mean absolute error 0.367
Root mean squared error 0.4345
Relative absolute error 74.6588 %
Root relative squared error 87.6059 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Ar
0.899 0.478 0.708 0.899 0.792 0.462 0.776
0.522 0.101 0.800 0.522 0.632 0.462 0.776
Weighted Avg. 0.734 0.314 0.748 0.734 0.722 0.462 0.776
=== Confusion Matrix ===
a b <-- classified as
160 18 | a = Paid
66 72 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 237 75 %
Incorrectly Classified Instances 79 25 %
Kappa statistic 0.472
Mean absolute error 0.3712
Root mean squared error 0.4419
Relative absolute error 75.8016 %
Root relative squared error 89.5308 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Ar
0.853 0.394 0.751 0.853 0.799 0.479 0.743
0.606 0.147 0.748 0.606 0.669 0.479 0.743
Weighted Avg. 0.750 0.291 0.750 0.750 0.745 0.479 0.743
=== Confusion Matrix ===
a b <-- classified as
157 27 | a = Paid
52 80 | b = Defaulted


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
MultilayerPerceptron -L 0.1 -M 0.15 -N 500 -V 0 -S 0 -E 20 -H "3, 2"
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 80
More options...
(Nom) Outcome
Start
Stop
Result list (right-click for options)
01:16:54 - functions.MultilayerPerceptron
01:17:32 - functions.MultilayerPerceptron
01:17:39 - functions.MultilayerPerceptron
01:17:44 - functions.MultilayerPerceptron
01:17:47 - functions.MultilayerPerceptron
01:17:52 - functions.MultilayerPerceptron
01:17:56 - functions.MultilayerPerceptron
01:18:02 - functions.MultilayerPerceptron
01:18:06 - functions.MultilayerPerceptron
01:18:12 - functions.MultilayerPerceptron
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 237 75 %
Incorrectly Classified Instances 79 25 %
Kappa statistic 0.472
Mean absolute error 0.3655
Root mean squared error 0.4339
Relative absolute error 74.6304 %
Root relative squared error 87.8995 %
Total Number of Instances 316
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC
0.853 0.394 0.751 0.853 0.799 0.479
0.606 0.147 0.748 0.606 0.669 0.479
Weighted Avg. 0.750 0.291 0.750 0.750 0.745 0.479
=== Confusion Matrix ===
a b <-- classified as
157 27 | a = Paid
52 80 | b = Defaulted