

Name - Pratik Bhujade

1. Description of your dataset(s) and findings

1) **Title:** German Credit Data

2) **Data description:**

o **The problem domain** - Credit classification, The original dataset contains 1000 entries with 20 categorial/symbolic attributes prepared by Prof. Hofmann. In this dataset, each entry represents a person who takes a credit by a bank. Each person is classified as good or bad credit risks according to the set of attributes.

o **The source of the data** -

UCI Machine Learning Repository

Professor Dr. Hans Hofmann

Institut f"ur Statistik und "Okonometrie

Universit"at Hamburg

FB

Wirtschaftswissenschaften

Von-Melle-Park 5

2000 Hamburg 13

o **The agencies working with the data**

- Open Knowledge Foundation Germany , AlgorithmWatch

o **The intended use of the data**

- This dataset represents entries of people taking a credit by a bank. Good or Bad credit risk is analysed for each person based on the set of attributes.

o The attribute types of the data-

Attributes:	Type:
Status of existing of checking account	Nominal
Duration in months	Numeric
Credit history	Nominal
Purpose	Nominal
Credit amount	Numeric
Saving accounts/bonds	Nominal
Present employment since	Nominal
Installment rate in percentage of disposable income	Numeric
Personal and Sex	Nominal
Other debtors/guarantors	Nominal
Present residence since	Numeric
property	Nominal
Age in years	Numeric
Other installment plans	Nominal
Housing	Nominal
No of existing credits at this bank	Numeric
job	Nominal
No of people being liable to provide maintaince for	Numeric
Telephone	Nominal
Foreign worker	Nominal

Description of Attributes

Attribute 1: (qualitative)

Status of existing checking account

A11 : ... < 0 DM

A12 : $0 \leq \dots < 200$ DM

A13 : ... ≥ 200 DM / salary assignments for at least 1 year

A14 : no checking account

Attribute 2: (numerical)

Duration in month

Attribute 3: (qualitative)

Credit history

A30 : no credits taken/ all credits paid back duly

A31 : all credits at this bank paid back duly

A32 : existing credits paid back duly till now

A33 : delay in paying off in the past

A34 : critical account/ other credits existing (not at this bank)

Attribute 4: (qualitative)

Purpose

A40 : car (new)

A41 : car (used)

A42 : furniture/equipment

A43 : radio/television

A44 : domestic appliances

A45 : repairs

A46 : education

A47 : (vacation - does not exist?)

A48 : retraining

A49 : business

A410 : others

Attribute 5: (numerical)

Credit amount

Attribute 6: (qualitative)

Savings account/bonds

A61 : ... < 100 DM

A62 : 100 <= ... < 500 DM

A63 : 500 <= ... < 1000 DM

A64 : .. >= 1000 DM

A65 : unknown/ no savings account

Present employment since

Attribute 7: (qualitative)

Present employment since

A71 : unemployed

A72 : ... < 1 year

A73 : 1 <= ... < 4 years

A74 : 4 <= ... < 7 years

A75 : .. >= 7 years

Attribute 8: (numerical)

Instalment rate in percentage of disposable income

Attribute 9: (qualitative)

Personal status and sex

A91 : male : divorced/separated

A92 : female : divorced/separated/married

A93 : male : single

A94 : male : married/widowed

A95 : female : single

Attribute 10: (qualitative)

Other debtors / guarantors A101 : none

A102 : co-applicant A103
: guarantor

Attribute 11: (numerical)

Present residence since

Attribute 12: (qualitative)

Property

A121 : real estate

A122 : if not A121 : building society savings agreement/ life insurance

A123 : if not A121/A122 : car or other, not in attribute 6

A124 : unknown / no property

Attribute 13: (numerical)

Age in years

Attribute 14: (qualitative)

Other instalment plans

A141 : bank

A142 : stores

A143 : none

Attribute 15: (qualitative)

Housing A151 : rent

A152 : own

A153 : for free

Attribute 16: (numerical)

Number of existing credits at this bank

Attribute 17: (qualitative)

Job

A171 : unemployed/ unskilled - non-resident

A172 : unskilled - resident

A173 : skilled employee / official

A174 : management/ self-employed/ highly qualified employee/ officer

Attribute 18: (numerical)

Number of people being liable to provide maintenance for

Attribute 19: (qualitative)

TelephoneA19

1 : noneA192 : yes, registered under the customer's name

Attribute 20: (qualitative)

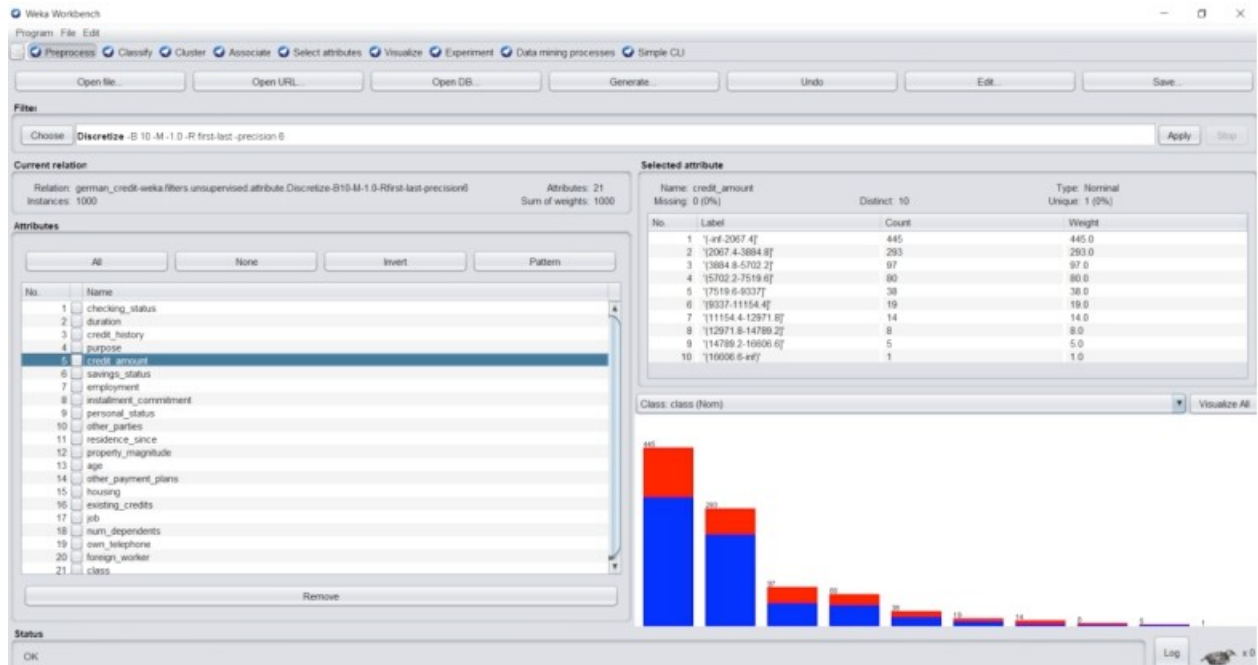
foreign worker

A201 : yes

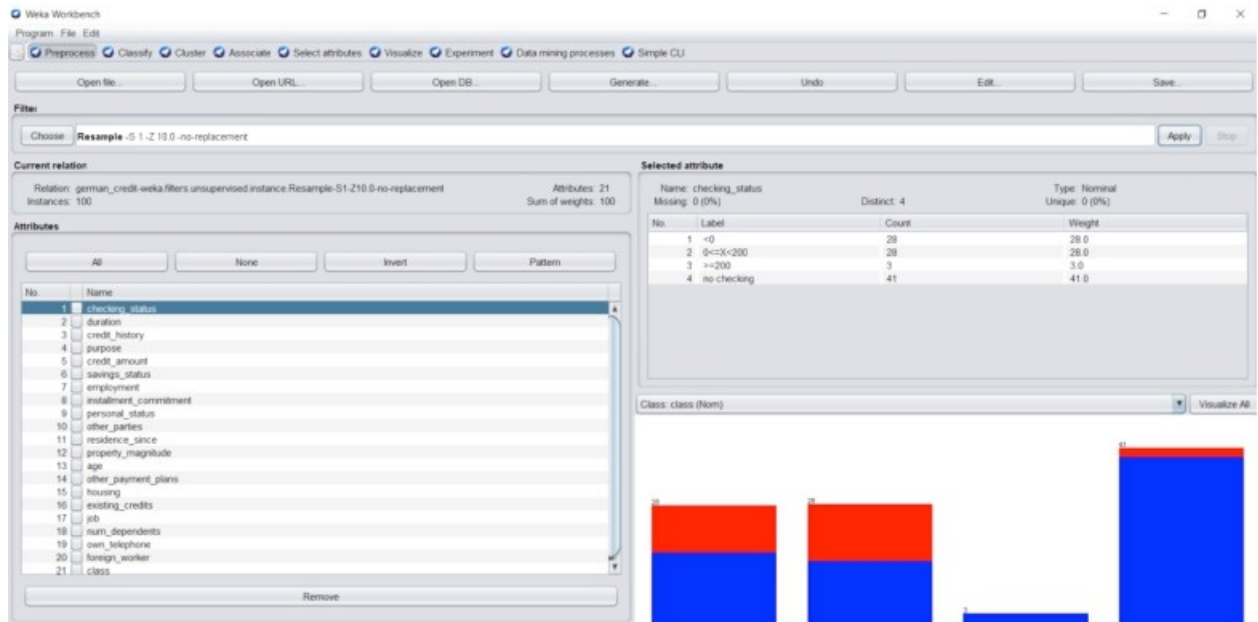
A202 : no

Summary of Dataset: The original dataset is made up of 1000 entries with 20 categorical/ symbolic attributes prepared by Prof. Hofmann. This dataset, each entry represents a person who takes a credit by a bank. Each person is classified as good or bad credit risks according to the set of attributes.

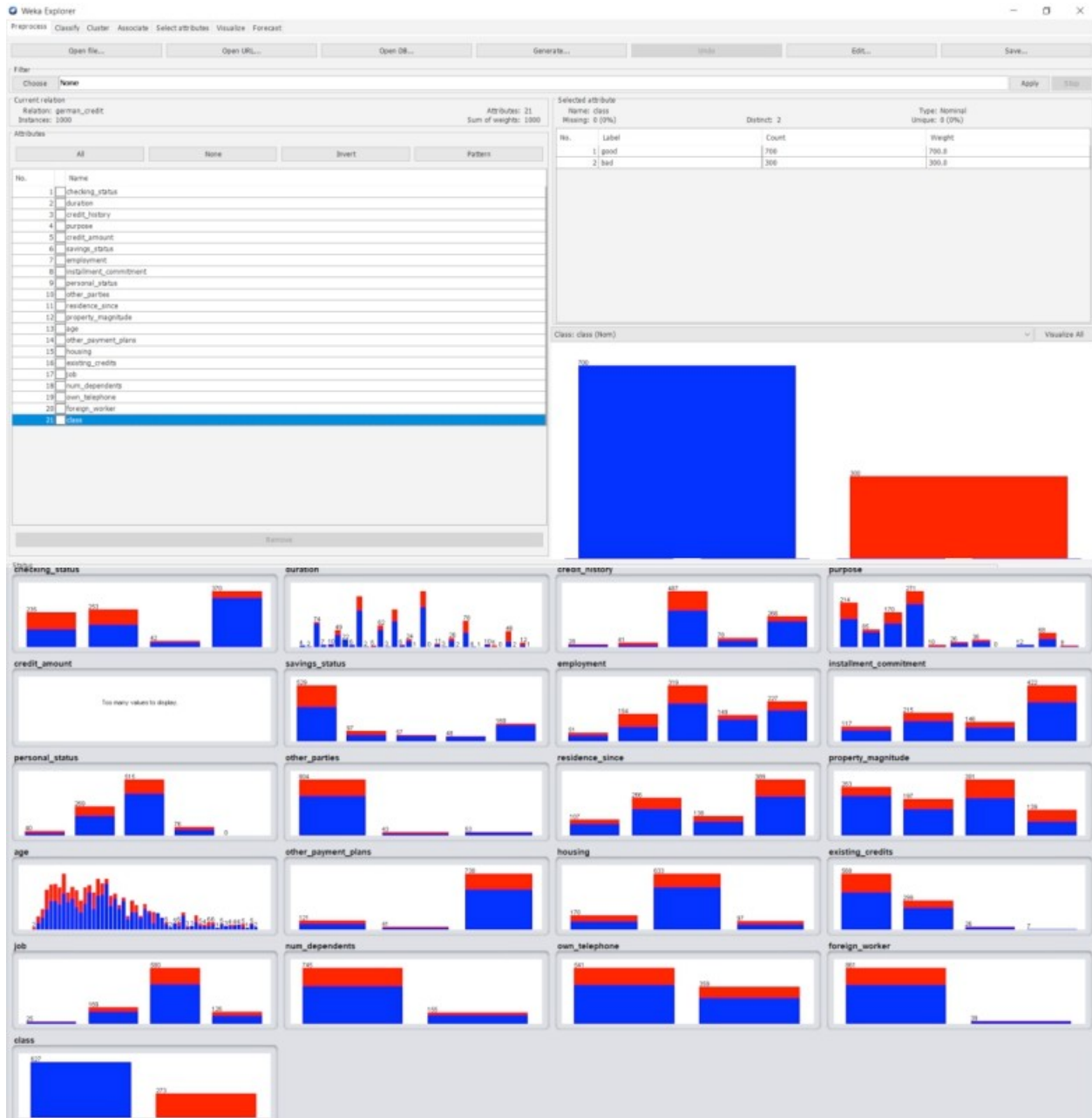
All Attributes:



Visualisation



Visualisation of Attributes:



Summary on Weka:

```
%  
% 1. Title: German Credit data  
%  
% 2. Source Information  
%  
% Professor Dr. Hans Hofmann  
% Institut für Statistik und Ökonometrie  
% Universität Hamburg  
% FB Wirtschaftswissenschaften  
% Von-Melle-Park 5  
% 2000 Hamburg 13  
%  
% 3. Number of Instances: 1000  
%  
% Two datasets are provided. the original dataset, in the form provided  
% by Prof. Hofmann, contains categorical/symbolic attributes and  
% is in the file "german.data".  
%  
% For algorithms that need numerical attributes, Strathclyde University  
% produced the file "german.data-numeric". This file has been edited  
% and several indicator variables added to make it suitable for  
% algorithms which cannot cope with categorical variables. Several  
% attributes that are ordered categorical (such as attribute 17) have  
% been coded as integer. This was the form used by Statlog.  
%  
%  
% 6. Number of Attributes german: 20 (7 numerical, 13 categorical)  
%    Number of Attributes german.numer: 24 (24 numerical)  
%  
%  
% 7. Attribute description for german
```

```

%
% Attribute 1: (qualitative)
%      Status of existing checking account
%      A11 :      ... <   0 DM
%      A12 : 0 <= ... < 200 DM
%      A13 :      ... >= 200 DM /
%      salary assignments for at least 1 year
%      A14 : no checking account
%
% Attribute 2: (numerical)
%      Duration in month
%
% Attribute 3: (qualitative)
%      Credit history
%      A30 : no credits taken/
%      all credits paid back duly
%      A31 : all credits at this bank paid back duly
%      A32 : existing credits paid back duly till now
%      A33 : delay in paying off in the past
%      A34 : critical account/
%      other credits existing (not at this bank)
%
% Attribute 4: (qualitative)
%      Purpose
%      A40 : car (new)
%      A41 : car (used)
%      A42 : furniture/equipment
%      A43 : radio/television
%      A44 : domestic appliances
%      A45 : repairs
%      A46 : education
%      A47 : (vacation - does not exist?)
%      A48 : retraining
%      A49 : business
%      A410 : others
%
% Attribute 5: (numerical)

```

Objective:

To identify fraudulent Credit Card Transactions so that the customer isn't charged for items they didn't purchase.

Summary of Findings:

The dataset is preprocessed using Numeric to Nominal filter as most of the data is numeric and qualitative. The DataMining techniques used are J48 Tree which is a Classification Technique giving 98% accuracy by varying the parameters. Similarly, Voted Perceptron an advanced machine learning technique available in WEKA did not produce satisfactory results as the maximum accuracy stood around 88%. Classes were also used to Cluster evaluation technique which was part of WEKA software. In Conclusion, using only the credit amount attribute , fraudulent transactions can be identified.

2. Preprocessing

The dataset did not consist of any missing or duplicate values

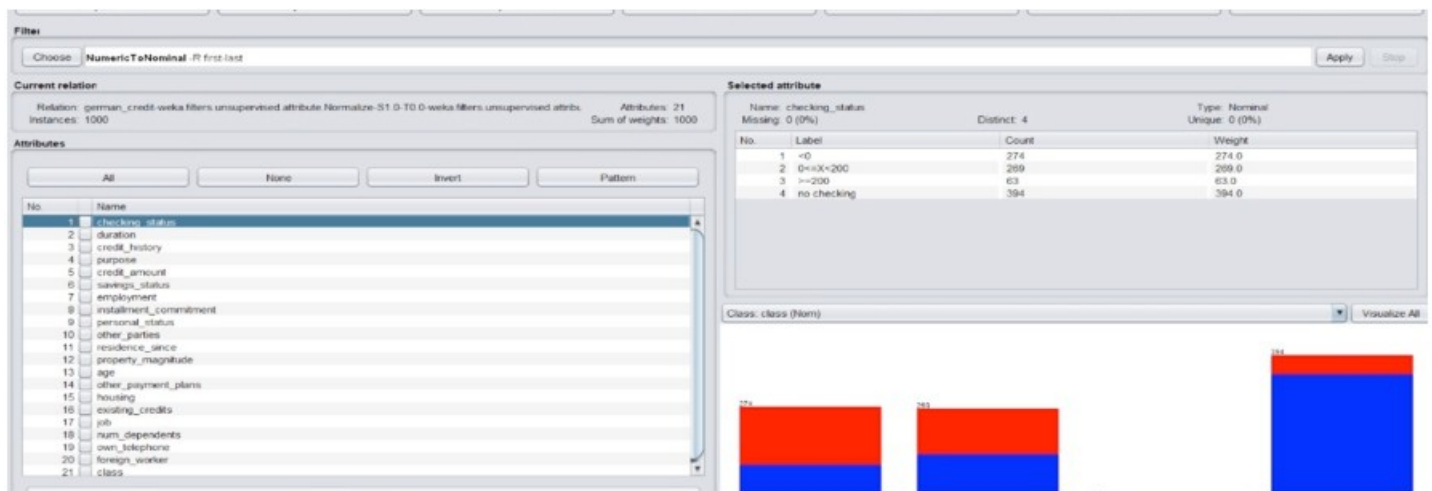
The Set of preprocessing techniques analysed are:

- 1) Numeric to Nominal
- 2) Nominal to Binary
- 3) Normalise
- 4) Discretise

The dataset consists of nominal and numeric values but class output is nominal-good or bad.

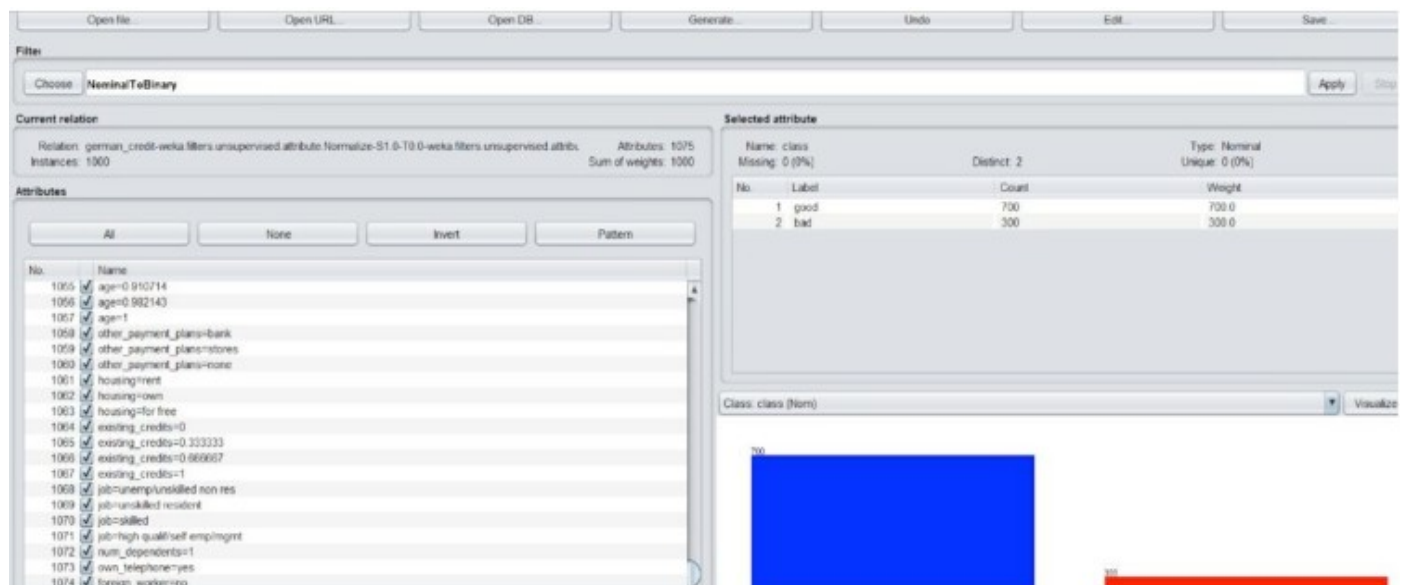
In this case we will have to analyse the dataset after applying the above mentioned preprocessing techniques.

The preprocessing technique that gives best result in the form of Nominal good or bad. Many preprocessing techniques were explored to clean the data to get desired results.



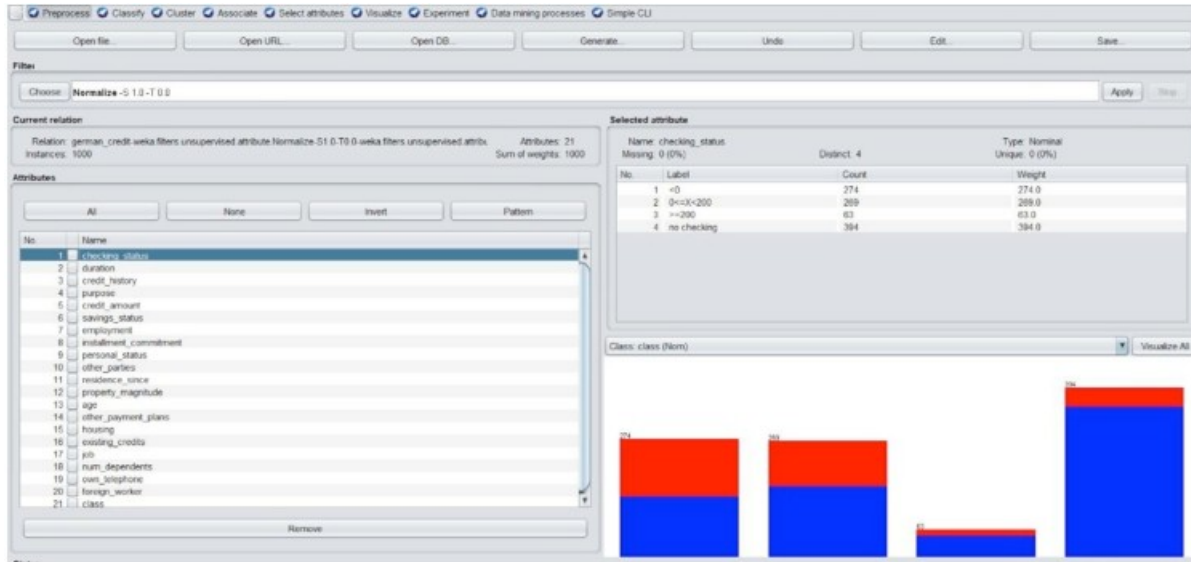
NUMERIC TO NOMINAL

Numeric to nominal is used as all attribute values are converted to nominal form which is the technique being used.



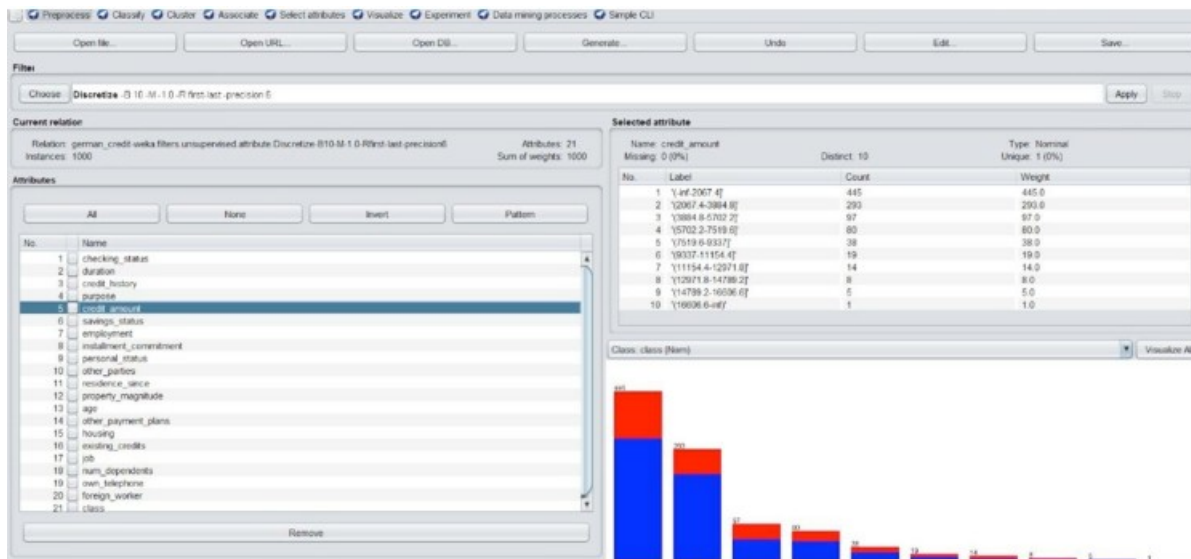
NOMINAL TO BINARY

Nominal to binary is also not good enough as it results in more attributes.



NORMALIZE

Normalise is another technique which was used but did not produce desired results.

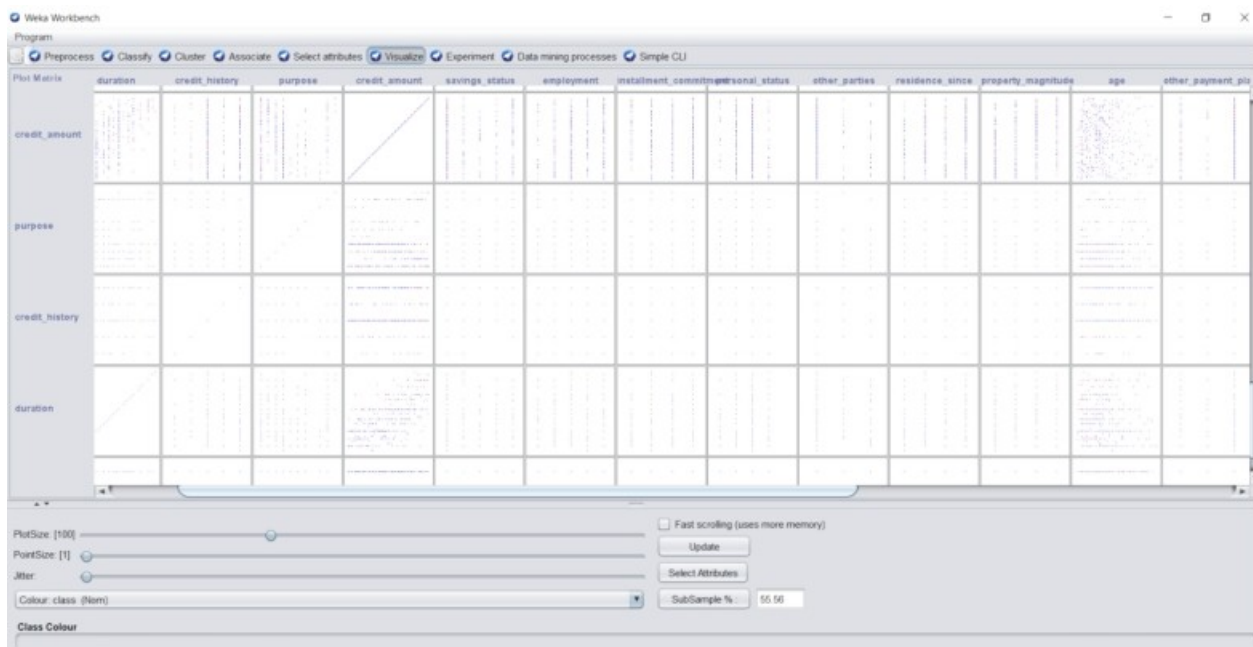
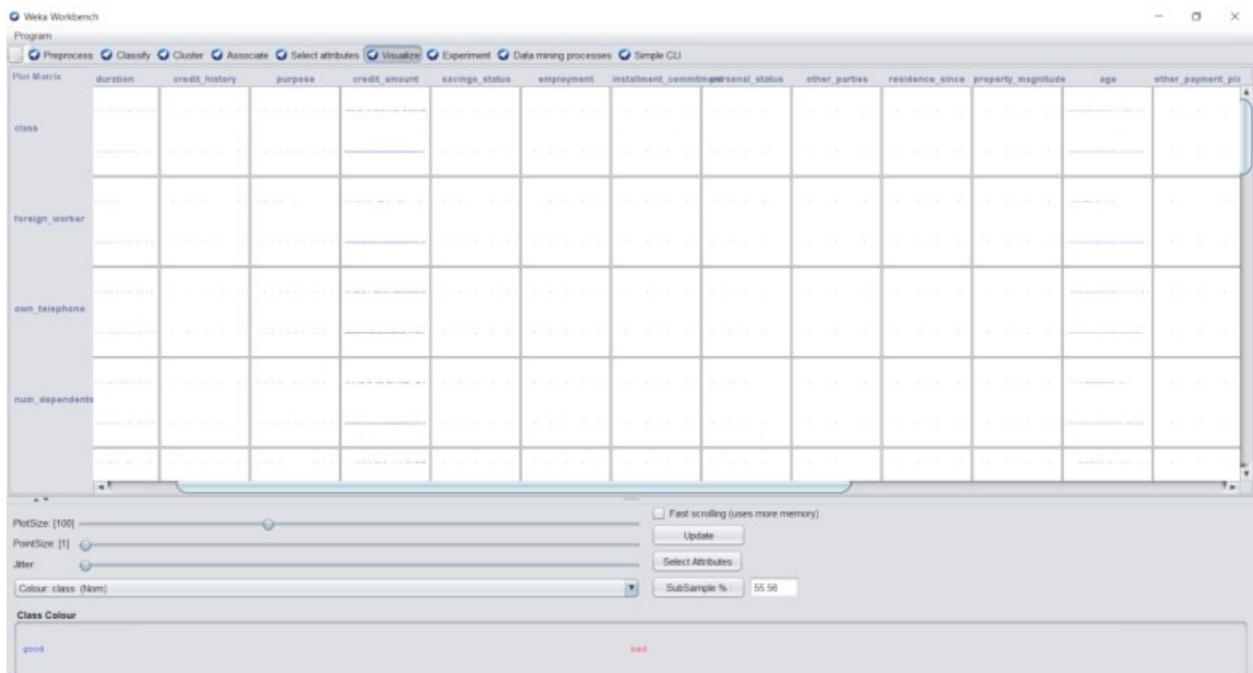


DISCRETISE

Discretise does not work as labels are changed that leads to misinformation.

- 1) The outcome is given in dataset.arff present in the zip file.
Normalize in the range $[-1,1]$. For ML algorithms most of the cases normalization is important as attribute values can differ in order of magnitudes.
- 2) Discretize all numerical values to 3 nominals. This will allow us to use J48.

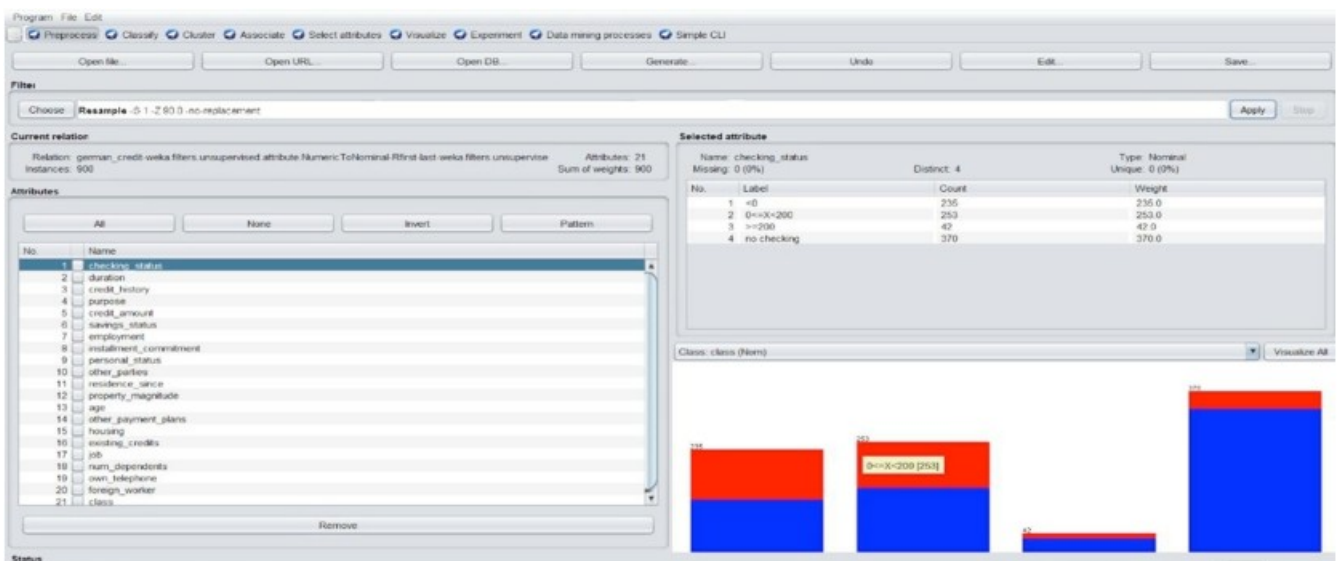
Visualisation of dataset



3. Divided dataset into training and test set

The dataset is divided into training and test set into (9:1)

ratio. **Training Set**



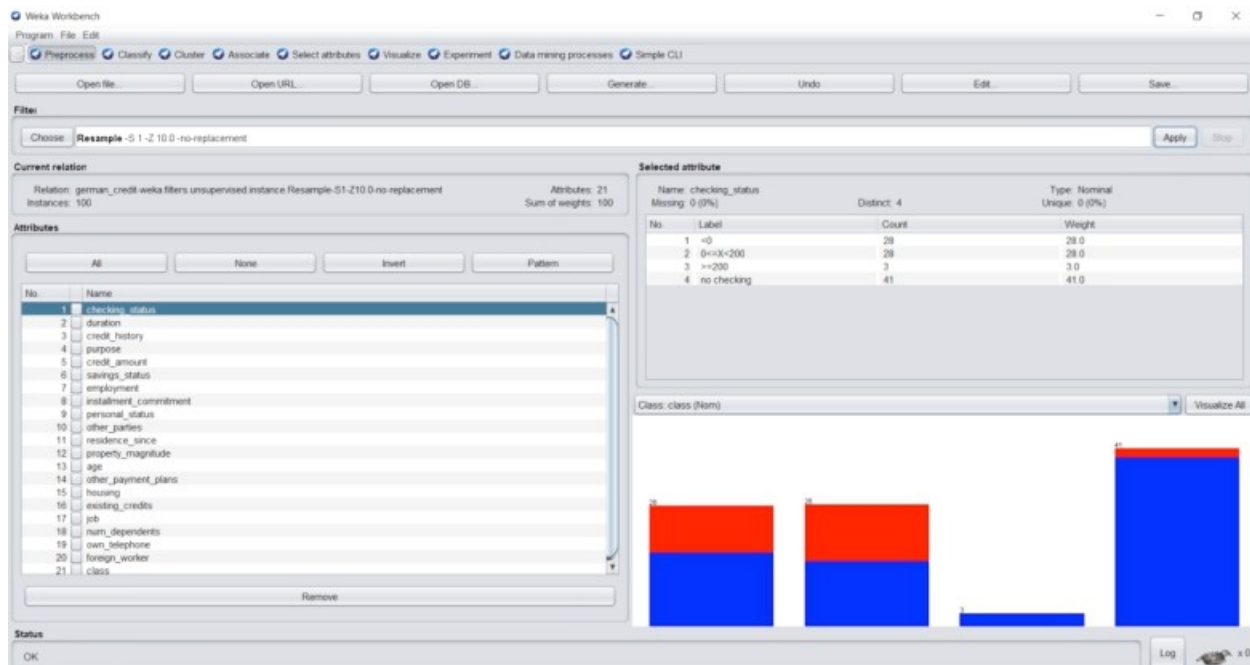
Raw view of dataset(Training)

Viewer

Relation: german_credit-weka.filters.unsupervised.attribute.NumericToNominal-R1rst-last-weka.filters.unsupervised.attribute.NumericToNominal-R1rst-last-weka.filters.unsupervised.instance.Resample-S1-Z100-0-weka.filters.unsupervised.instance.Resample-S1-Z100-0-no-replacement

No.	1: checking Nominal	2: duration Nominal	3: credit_history Nominal	4: purpose Nominal	5: credit_amount Nominal	6: savings_status Nominal	7: employment Nominal	8: installment_commitment Nominal	9: personal_status Nominal	10: other_parties Nominal	11: residence_since Nominal	12: property_magnitude Nominal	13: age Nominal	14: other_payment_plans Nominal	15: housing Nominal
...	no checking	24	existing paid	furniture/...	3062	500=>X(1000	=7	4	male single	none	3	no known property	32	none	rent
(0	critical/other e...	18	critical/other e...	furniture/...	1049	(100	(1	4	female div/deptm...	none	4	life insurance	21	none	rent
...	no checking	18	critical/other e...	radio/hv	6070	(100	=7	3	male single	none	4	car	33	none	own
(0	...	12	existing paid	furniture/...	652	(100	=7	4	female div/deptm...	none	4	life insurance	24	none	rent
...	no checking	18	existing paid	new car	2962	no known savin...	4(=X(7	4	male single	none	3	life insurance	32	none	own
...	no checking	18	critical/other e...	furniture/...	...	(100	(1	3	male div/deptm...	none	2	car	35	none	own
...	no checking	24	existing paid	radio/hv	3181	(100	(1	4	female div/deptm...	none	4	life insurance	26	none	own
...	no checking	15	existing paid	radio/hv	1386	no known savin...	1(=X(4	4	male mar/wid	none	2	real estate	40	none	own
0(=X(200	...	24	delayed previo...	furniture/...	2064	(100	unemployed	3	female div/deptm...	none	2	life insurance	34	none	rent
...	no checking	12	critical/other e...	radio/hv	2311	no known savin...	=7	1	male single	co applicant	4	real estate	49	none	own
...	no checking	6	existing paid	furniture/...	2978	500=>X(1000	1(=X(4	1	male single	none	2	car	32	none	own
(0	...	18	all paid	new car	1442	(100	4(=X(7	4	male single	none	4	no known property	32	none	for free
(0	...	36	existing paid	new car	1842	(100	(1	4	female div/deptm...	none	4	car	34	none	own
(0	...	12	existing paid	new car	2579	(100	(1	4	male single	none	1	real estate	33	none	own
...	no checking	18	existing paid	radio/hv	433	(100	unemployed	3	female div/deptm...	co applicant	4	real estate	22	none	rent
(0	...	16	critical/other e...	new car	2625	(100	=7	2	male single	guarantor	4	life insurance	43	bank	rent
0(=X(200	...	48	existing paid	furniture/...	9960	(100	(1	1	female div/deptm...	none	2	car	26	none	own
...	no checking	30	delayed previo...	business	4272	100=>X(500	1(=X(4	2	male single	none	2	life insurance	26	none	own
...	no checking	48	existing paid	business	3914	no known savin...	1(=X(4	4	male div/deptm...	none	2	real estate	38	bank	own
...	no checking	36	existing paid	business	7409	no known savin...	=7	3	male single	none	2	life insurance	37	none	own
0(=X(200	...	9	existing paid	furniture/...	2030	no known savin...	4(=X(7	2	male single	none	1	car	24	none	own
...	no checking	18	existing paid	radio/hv	2051	(100	(1	4	male single	none	1	real estate	33	none	own
...	no checking	21	existing paid	business	1572	=1000	=7	4	female div/deptm...	none	4	real estate	36	bank	own
0(=X(200	...	21	critical/other e...	business	3652	(100	4(=X(7	2	male single	none	3	life insurance	27	none	own
(0	...	36	existing paid	radio/hv	2302	(100	1(=X(4	4	male div/deptm...	none	4	car	31	none	rent
(0	...	18	critical/other e...	new car	3966	(100	=7	1	female div/deptm...	none	4	real estate	33	bank	rent
...	...	8	critical/other e...	new car	731	(100	=7	4	male single	none	4	real estate	47	none	own
0(=X(200	...	24	existing paid	radio/hv	5084	no known savin...	=7	2	female div/deptm...	none	4	car	42	none	own
(0	...	12	critical/other e...	used car	1526	(100	=7	4	male single	none	4	no known property	66	none	for free
0(=X(200	...	48	no credits/all p...	business	3844	100=>X(500	4(=X(7	4	male single	none	4	no known property	34	none	for free
...	no checking	18	critical/other e...	radio/hv	1169	no known savin...	1(=X(4	4	male single	none	3	life insurance	29	none	own
(0	...	30	no credits/all p...	furniture/...	4583	(100	1(=X(4	2	male div/deptm...	guarantor	2	real estate	32	none	own
(0	...	12	all paid	new car	697	(100	(1	4	male single	none	2	car	46	bank	own
...	no checking	18	critical/other e...	radio/hv	1800	(100	1(=X(4	4	male single	none	2	car	24	none	own
0(=X(200	...	6	all paid	new car	931	100=>X(500	(1	1	female div/deptm...	none	1	life insurance	32	stores	own
(0	...	24	existing paid	furniture/...	7721	no known savin...	(1	1	female div/deptm...	none	2	life insurance	30	none	own
...	no checking	36	critical/other e...	business	6304	no known savin...	=7	4	male single	none	4	real estate	36	none	own
...	no checking	15	critical/other e...	education	1532	100=>X(500	1(=X(4	4	female div/deptm...	none	3	car	31	none	own
...	no checking	12	critical/other e...	new car	682	100=>X(500	4(=X(7	4	female div/deptm...	none	3	car	51	none	own
...	no checking	12	critical/other e...	radio/hv	1934	(100	=7	2	male single	none	2	no known property	26	none	own
...	no checking	36	critical/other e...	furniture/...	7127	(100	(1	2	female div/deptm...	none	4	life insurance	23	none	rent
(0	...	24	existing paid	furniture/...	2996	no known savin...	1(=X(4	2	male mar/wid	none	4	car	20	none	own
(0	...	9	existing paid	radio/hv	1364	(100	4(=X(7	3	male single	none	4	real estate	59	none	own

Test Set



Raw view dataset (Test Set)

Viewer

Relation: german_credit-weka filters unsupervised attribute Numeric ToNominal filter-last-weka filters unsupervised instance Resample-S1-210.0

10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
checking_status	duration	credit_history	purpose	credit_amount	savings_status	employment	installment_commitment	personal_status	other_partners	residence_since	property_magnitude	age	other_payment_plans	housing	
Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	
58	0	24	existing paid	furniture/	3149	(100	(1	4	male single	none	1	no known property	22	bank	for free
59	0	24	existing paid	new car	1207	(100	(1	4	female div/depm...	none	4	life insurance	24	none	rent
60	0<=X/200	24	existing paid	furniture/...	4057	(100	4<=X/7	3	male div/sep	none	3	car	43	none	own
61	0<=X/200	7	existing paid	radio/v	2329	(100	(1	1	female div/depm...	guarantor	1	real estate	45	none	own
62	no checking	10	existing paid	radio/v	1924	(100	1<=X/4	1	male single	none	4	life insurance	38	none	own
63	no checking	12	existing paid	radio/v	3077	(100	1<=X/4	2	male single	none	4	car	52	none	own
64	no checking	24	critical/other e.	radio/v	5103	(100	(1	3	male mar/wid	none	3	no known property	47	none	for free
65	no checking	18	critical/other e.	new car	2775	(100	4<=X/7	2	male single	none	2	life insurance	31	bank	own
66	0	30	existing paid	used car	3857	(100	1<=X/4	4	male div/sep	none	4	life insurance	40	none	own
67	no checking	15	delayed previo.	used car	3594	(100	(1	1	female div/depm...	none	2	life insurance	40	none	own
68	no checking	24	existing paid	new car	2255	no known sawn...	(1	4	male single	none	1	life insurance	54	none	own
69	no checking	15	existing paid	furniture/...	2221	500<=X/1000	1<=X/4	2	female div/depm...	none	4	car	20	none	rent
70	0	36	existing paid	radio/v	2302	(100	1<=X/4	4	male div/sep	none	4	car	31	none	rent
71	no checking	9	existing paid	new car	3577	100<=X/500	1<=X/4	1	male single	guarantor	2	real estate	26	none	rent
72	no checking	30	critical/other e.	radio/v	3077	no known sawn...	>=7	3	male single	none	2	car	40	none	own
73	0<=X/200	48	no credits/all p.	business	12204	no known sawn...	1<=X/4	2	male single	none	2	car	48	bank	own
74	>=200	18	existing paid	new car	1961	(100	>=7	3	female div/depm...	none	2	car	23	none	own
75	0	48	no credits/all p.	furniture/...	7119	(100	1<=X/4	3	male single	none	4	no known property	53	none	for free
76	0<=X/200	36	existing paid	education	12612	100<=X/500	1<=X/4	1	male single	none	4	no known property	47	none	for free
77	0<=X/200	27	existing paid	business	3915	(100	1<=X/4	4	male single	none	2	car	36	none	own
78	0	24	existing paid	radio/v	1987	(100	1<=X/4	2	male single	none	4	real estate	21	none	rent
79	0<=X/200	48	existing paid	radio/v	5951	(100	1<=X/4	2	female div/depm...	none	2	real estate	22	none	own
80	no checking	12	critical/other e.	furniture/...	1258	(100	(1	2	female div/depm...	none	4	life insurance	22	none	rent
81	no checking	9	existing paid	radio/v	2753	100<=X/500	>=7	3	male single	<= applicant	4	car	35	none	own
82	>=200	6	delayed previo.	radio/v	683	(100	(1	2	female div/depm...	none	1	life insurance	29	bank	own
83	no checking	6	existing paid	radio/v	1595	(100	4<=X/7	3	male single	none	2	life insurance	51	none	own
84	0<=X/200	9	existing paid	radio/v	1206	(100	>=7	4	female div/depm...	none	4	real estate	25	none	own
85	0<=X/200	6	delayed previo.	new car	1209	(100	unemployed	4	male single	none	4	life insurance	47	none	own
86	no checking	36	existing paid	radio/v	2294	no known sawn...	1<=X/4	4	female div/depm...	none	4	car	25	none	own
87	no checking	6	critical/other e.	new car	362	100<=X/500	1<=X/4	4	female div/depm...	none	4	car	52	none	own
88	no checking	6	existing paid	used car	1236	500<=X/1000	1<=X/4	2	male single	none	4	life insurance	50	none	rent
89	0	12	critical/other e.	new car	691	(100	>=7	4	male single	none	3	life insurance	35	none	own
90	0<=X/200	8	existing paid	business	907	(100	(1	3	male mar/wid	none	2	real estate	26	none	own
91	0	6	critical/other e.	new car	3676	(100	1<=X/4	1	male single	none	3	real estate	37	none	rent
92	no checking	15	critical/other e.	education	1532	100<=X/500	1<=X/4	4	female div/depm...	none	3	car	31	none	own
93	0<=X/200	60	all paid	other	14782	100<=X/500	>=7	3	female div/depm...	none	4	no known property	60	bank	for free
94	0<=X/200	21	critical/other e.	business	3652	(100	4<=X/7	2	male single	none	3	life insurance	27	none	own
95	no checking	18	existing paid	used car	3378	no known sawn...	1<=X/4	2	male single	none	1	life insurance	31	none	own
96	0	21	critical/other e.	new car	1602	(100	>=7	4	male mar/wid	none	3	car	30	none	own
97	0<=X/200	36	existing paid	radio/v	2323	(100	4<=X/7	4	male single	none	4	car	24	none	rent
98	0<=X/200	9	existing paid	furniture/...	918	(100	1<=X/4	4	female div/depm...	none	1	life insurance	30	none	own
99	no checking	6	existing paid	repairs	660	500<=X/1000	4<=X/7	2	male mar/wid	none	4	real estate	23	none	rent
100	0	48	existing paid	education	7476	(100	4<=X/7	4	male single	none	1	no known property	50	none	for free

4. Classification/ Association: J48 Tree or Association Rules J48

Model Parameters

Consist of batch size

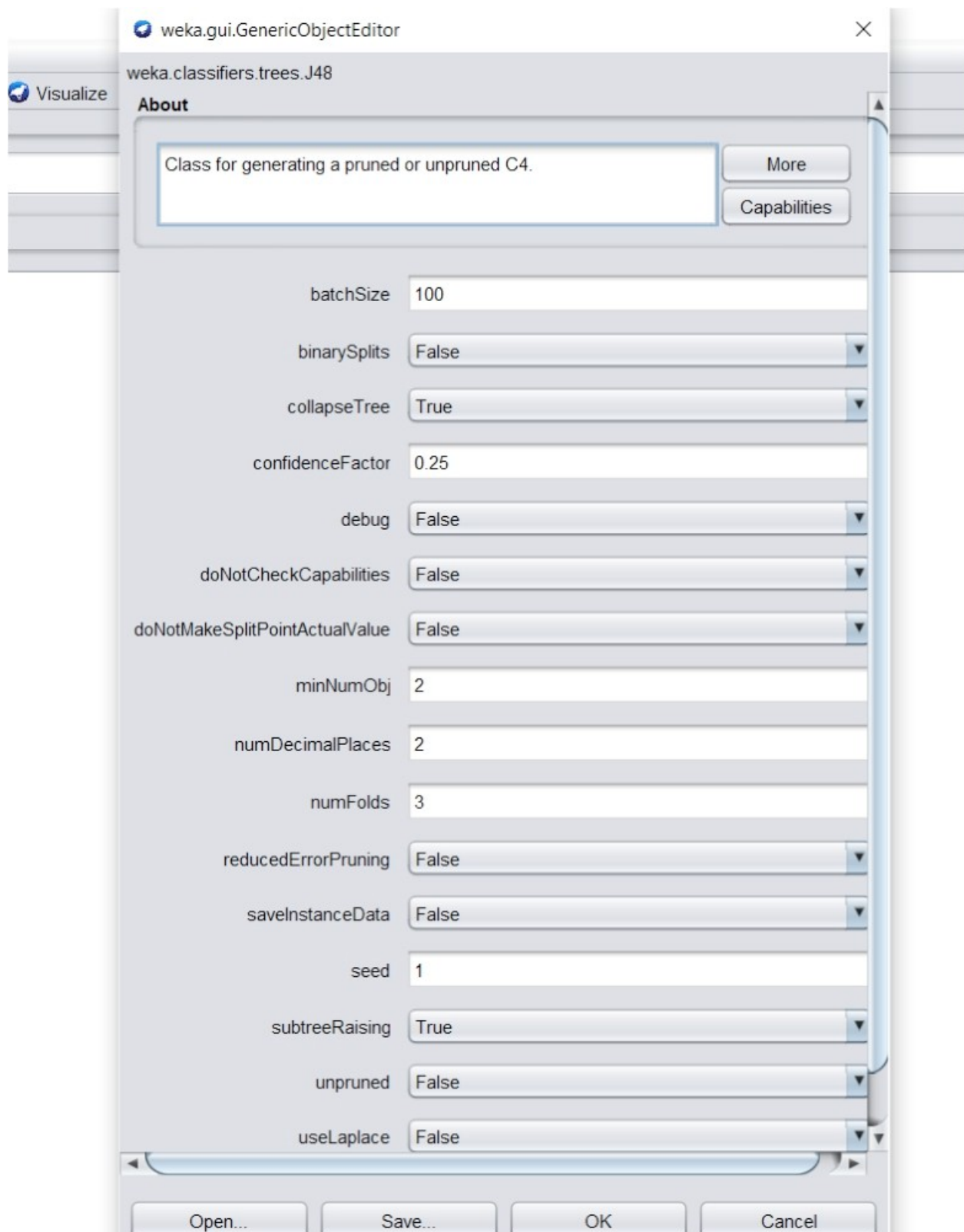
Subtree raising

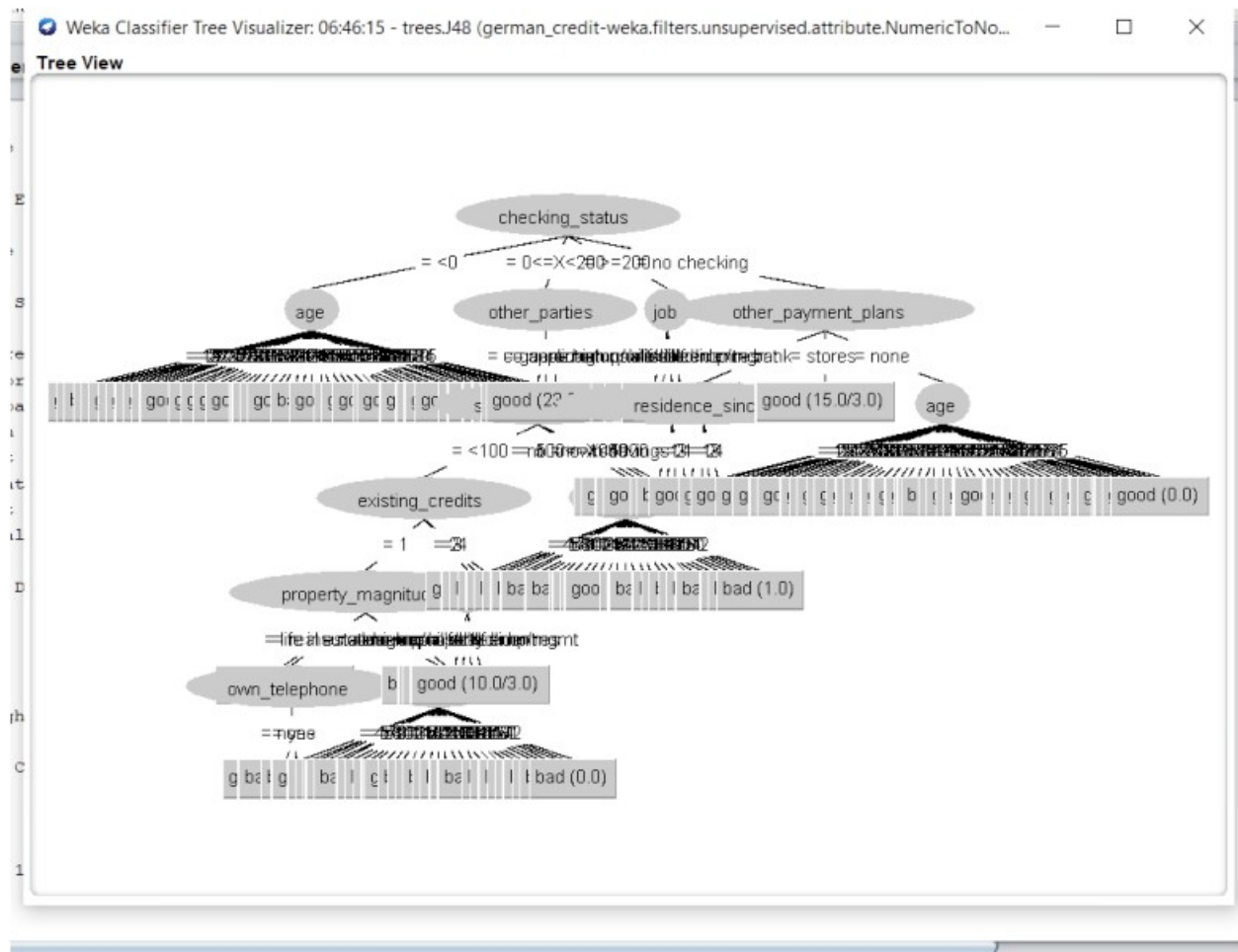
No of decimals

No of folds

Collapse tree

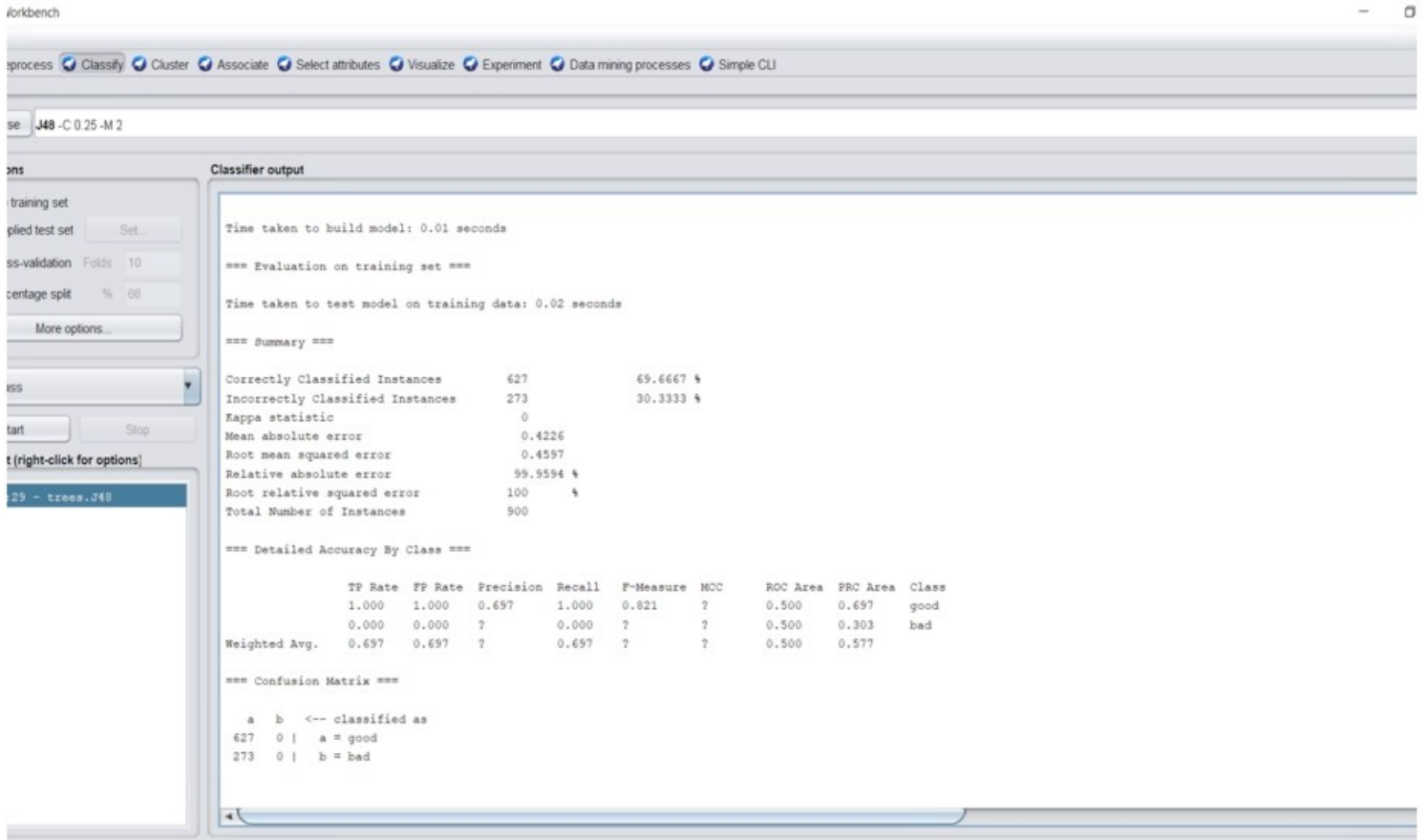
Minimum no of objects





Tree

Training J48 using 900 instances and using confidence factors and number of objects as parameters also include other parameters like number of folds and seed.



Train C=0.25,M=2

Detailed Run output with confusion matrix and tree architecture:

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: german_credit-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-
 weka.filters.unsupervised.attribute.NumericToNominal-
 Rfirst-last- weka.filters.unsupervised.instance.Resample-S1-
 Z100.0- weka.filters.unsupervised.instance.Resample-S1-
 Z90.0-no-replacement

Instances: 900

Attributes: 21

checking_status

duration

credit_history

purpose

credit_amount

savings_status

employment

installment_commitment

personal_status

other_parties

residence_since

property_magnitude

age

other_payment_plans

housing

existing_credits

job

num_dependents

own_telephone

foreign_worker

class

Test mode: evaluate on training data

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

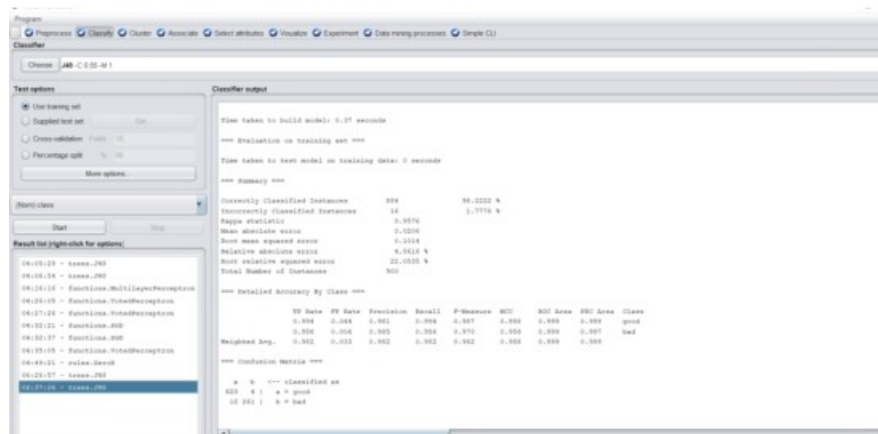
J48 pruned tree

: good (900.0/273.0)

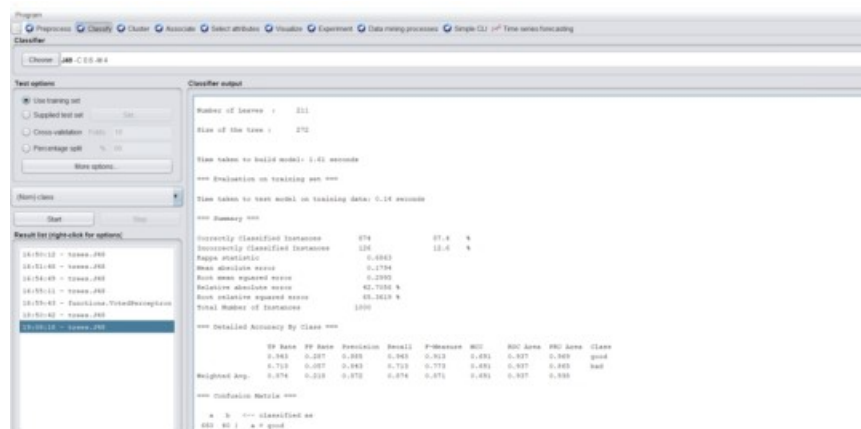
Number of Leaves : 1

Size of the tree : 1

Train : $C=0.55, M=1$



Train: $C= 0.6, M=5$



When we increase the confidence values or confidence factor(C) and the minimum number of objects(M) denoted by minNumObj the correctly classified instances increases. These are the parameters of the algorithm that have been varied.

When we took $M = 1$ and $C = 0.55$

correctly classified instances give

98.2%

incorrectly classified instances are 1.7778%

When we took $M = 2$ and $C = 0.25$

correctly classified instances give

69.66%

incorrectly classified instances are 30.33%

When we took $M = 4$ and $C = 0.6$

correctly classified instances give

87.4%

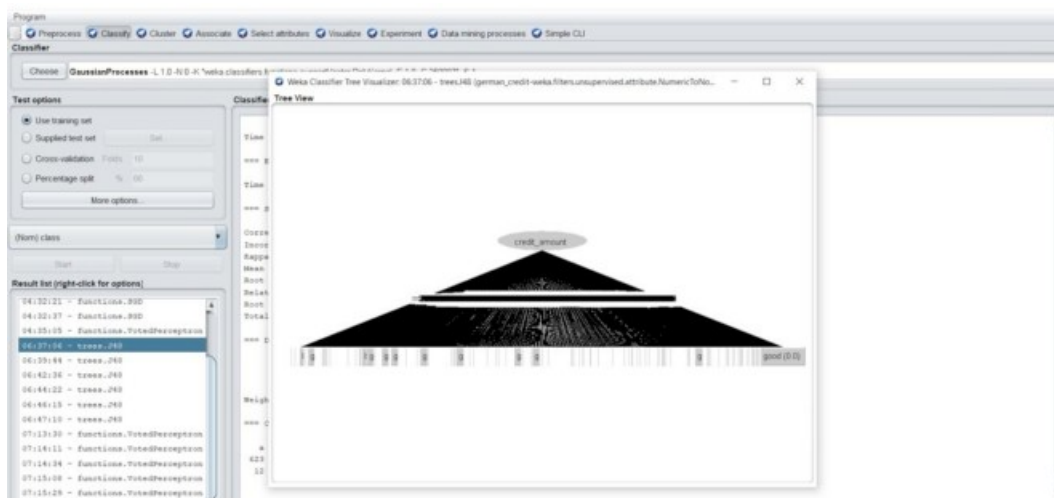
incorrectly classified instances are 12.6%

Visualisation of the training model

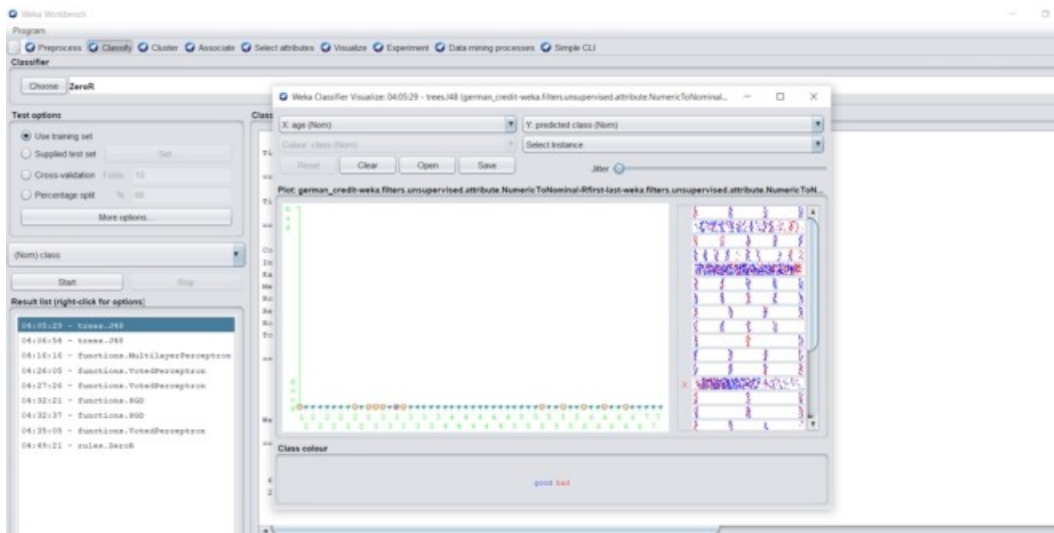
This helps to view relationship of different attributes in the model as well as the changes caused by tuning the parameters.



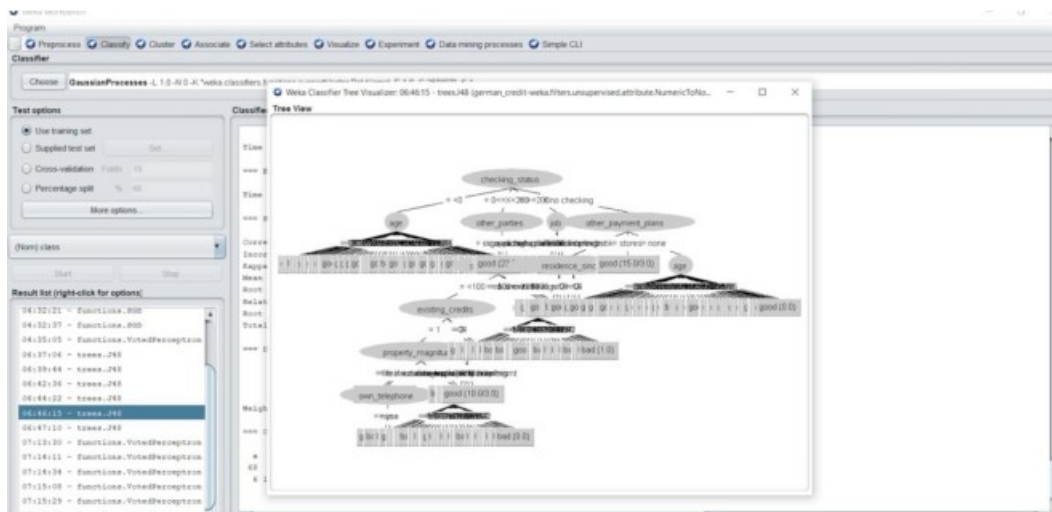
Train $C=0.25, M=1$



Train: $C = 0.55, M=1$



TRAIN $C = 0.6$, $M = 4$



TRAIN $C = 0.25$, $M = 2$

Testing the model

After training the dataset on 900 instances we test the dataset for given set of parameters.

For a $C = 0.25$, $M = 2$ correctly classified instances are 75% and incorrectly classified instances are 25%.

Optimal performance is given when $C = 0.75$, $M = 4$ which gives classified instances are 98% and incorrectly classified instances are 2%.

For a $C = 0.9$, $M = 7$ correctly classified instances are 87% and incorrectly classified instances are 13%.

For a $C = 0.8$, $M = 4$ correctly classified instances are 75% and incorrectly classified instances are 25%.

Classifier output

Time taken to build model: 0.34 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Metric	Value	%
Correctly Classified Instances	98	98 %
Incorrectly Classified Instances	2	2 %
Kappa statistic	0.9652	
Mean absolute error	0.0261	
Root mean squared error	0.1123	
Relative absolute error	6.4061 %	
Root relative squared error	25.7394 %	
Total Number of Instances	100	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	1.000	0.000	0.974	1.000	0.987	0.947	0.998	0.999	good
	0.920	0.000	1.000	0.920	0.958	0.947	0.998	0.999	bad

=== Confusion Matrix ===

```

a b  <-- classified as
75 0 | a = good
 2 2 | b = bad

```

TEST $C = 0.75$, $M = 4$

Classifier output

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Metric	Value	%
Correctly Classified Instances	75	75 %
Incorrectly Classified Instances	25	25 %
Kappa statistic	0	
Mean absolute error	0.4017	
Root mean squared error	0.4363	
Relative absolute error	59.9457 %	
Root relative squared error	59.9877 %	
Total Number of Instances	100	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	1.000	1.000	0.750	1.000	0.857	?	0.500	0.750	good
	0.000	0.000	?	0.000	?	?	0.500	0.250	bad

=== Confusion Matrix ===

```

a b  <-- classified as
75 0 | a = good
25 0 | b = bad

```

$C = 0.9$, $M = 7$

$C = 0.8$, $M = 4$

Classifier output

Time taken to build model: 0.35 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Metric	Value	%
Correctly Classified Instances	87	87 %
Incorrectly Classified Instances	13	13 %
Kappa statistic	0.6579	
Mean absolute error	0.1989	
Root mean squared error	0.3037	
Relative absolute error	45.4799 %	
Root relative squared error	69.6102 %	
Total Number of Instances	100	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.907	0.240	0.919	0.907	0.913	0.658	0.928	0.974	good
	0.760	0.093	0.731	0.760	0.745	0.658	0.928	0.816	bad

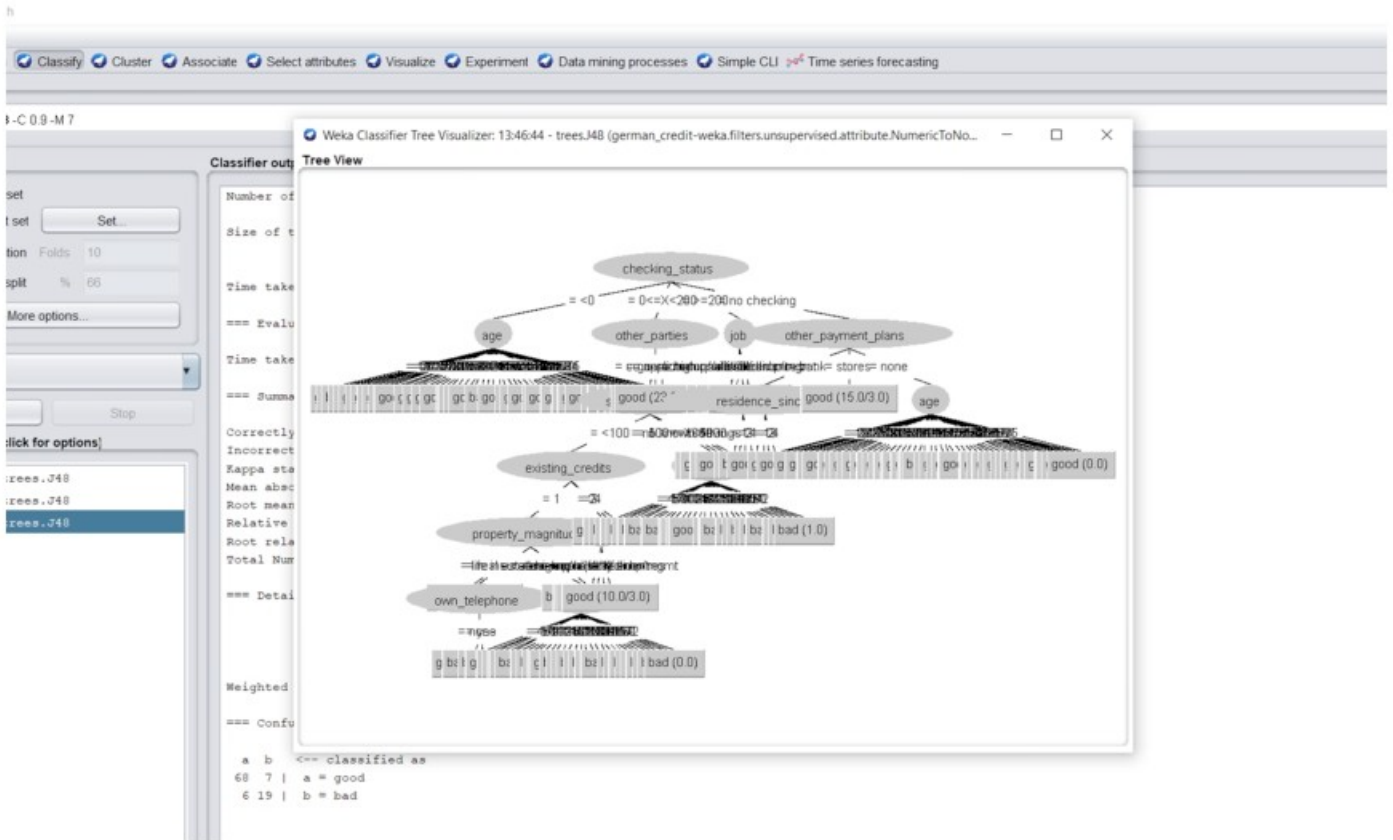
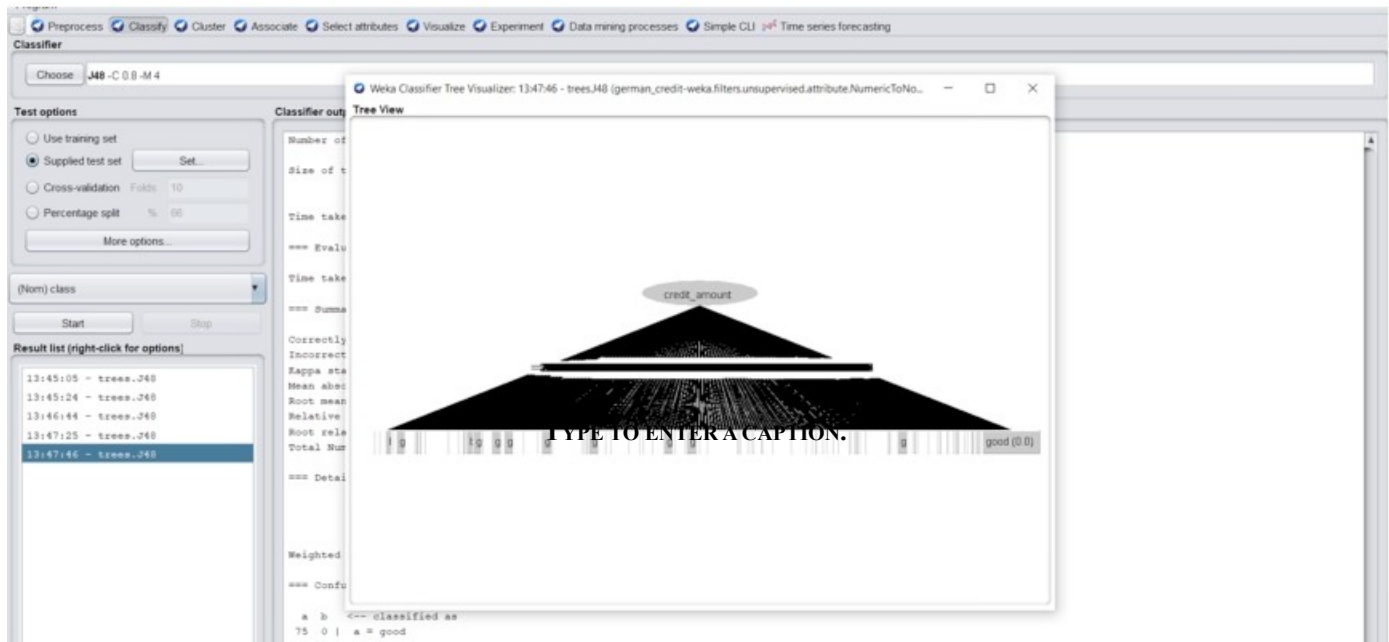
=== Confusion Matrix ===

```

a b  <-- classified as
87 7 | a = good
 6 19 | b = bad

```


Visualisation of the tested model , $C=0.8, M=4$



$C=0.9, M=0.7$



Confusion matrices:

=== Confusion Matrix ===

a	b		<-- classified as
660	40		a = good
86	214		b = bad

ADDITIONAL OUTPUT

=== Confusion Matrix ===

a	b		<-- classified as
75	0		a = good
2	23		b = bad

Additional Output

URL link of resources

1) <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>

This link was helpful in understanding the different Machine Learning algorithms and draw comparisons with the algorithms used in this report.

2) https://www.researchgate.net/figure/Explanation-of-WEKA-J48-parameters-Parameter-Description-Status_tbl2_258283784

This research paper was an analysis of J48 Tree and its usage in WEKA.

5. Classification: MLP or a similar advanced technique from Weka

Similar advanced technique used is voted perceptron.

Voted perceptron is based on the perceptron algorithm of Rousenblatt and Frank. The algorithm takes advantage of data that are linearly separable with large margins.

Training of the model

In Voted Perceptron, the parameters are number of iterations(I) and batch size. For the Classification model we chose an advanced technique called Voted Perceptron which comes under Perceptron Algorithm available in weka.

The optimal performance is given when I = 4 and batch size = 500 correctly classified instances are 87.25% and incorrectly classified instances are 12.75%

When I = 2 and batch size = 200 correctly classified instances are 80% and incorrectly classified instances are 20%

When I = 1 and batch size = 40 correctly classified instances are 70% and incorrectly classified instances are 30%

The screenshot shows the Weka Classifier window with the 'VotedPerceptron' model selected. The 'Test options' section has 'Use training set' selected. The 'Classifier output' section displays the following summary:

```

Time taken to build model: 0.6 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.07 seconds
=== Summary ===
Correctly Classified Instances      80      80 %
Incorrectly Classified Instances    20      20 %
Kappa statistic                    0.6000
Mean absolute error                 0.2
Root mean squared error            0.4472
Relative absolute error            45.7035 %
Root relative squared error       102.4924 %
Total Number of Instances         100
  
```

The 'Detailed Accuracy By Class' table is as follows:

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	AUC Area	PRC Area	Class
Weighted Avg.	0.800	0.200	0.800	0.800	0.800	0.481	0.764	0.547	good
									bad

The 'Confusion Matrix' is:

```

a b -- classified as
44 11 | a = good
44 11 | a = bad
  
```

Train I = 2, batch-size=200, I = 4, batch-size=400, I = 3, batch-size=400

The screenshot shows the Weka Classifier window with the 'VotedPerceptron' model selected. The 'Test options' section has 'Use training set' selected. The 'Classifier output' section displays the following summary:

```

Time taken to build model: 2.21 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 1.03 seconds
=== Summary ===
Correctly Classified Instances      795      87.2222 %
Incorrectly Classified Instances    115      12.7778 %
Kappa statistic                    0.687
Mean absolute error                 0.128
Root mean squared error            0.3575
Relative absolute error            30.2933 %
Root relative squared error       77.7494 %
Total Number of Instances         900
  
```

The 'Detailed Accuracy By Class' table is as follows:

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	AUC Area	PRC Area	Class
Weighted Avg.	0.935	0.275	0.888	0.935	0.911	0.690	0.839	0.881	good
									bad

The 'Confusion Matrix' is:

```

a b -- classified as
588 41 | a = good
74 139 | b = bad
  
```

The screenshot shows the Weka Workbench interface with the 'Classifier' window open. The 'Test options' section has 'Use training set' selected. The 'Classifier output' section displays the following summary:

```

Time taken to build model: 2.21 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 1.03 seconds
=== Summary ===
Correctly Classified Instances      795      87.2222 %
Incorrectly Classified Instances    115      12.7778 %
Kappa statistic                    0.687
Mean absolute error                 0.128
Root mean squared error            0.3575
Relative absolute error            30.2933 %
Root relative squared error       77.7494 %
Total Number of Instances         900
  
```

The 'Detailed Accuracy By Class' table is as follows:

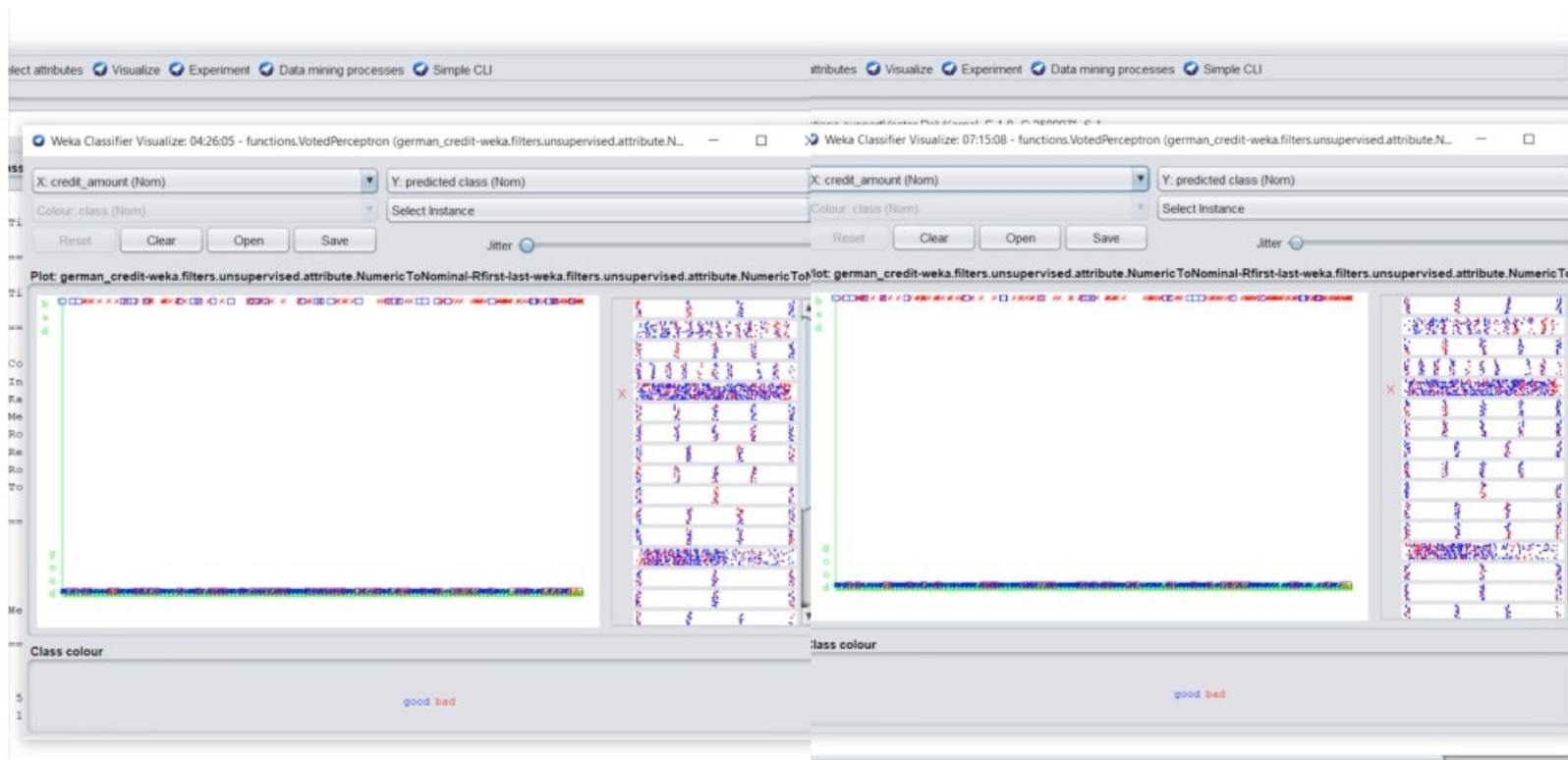
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	AUC Area	PRC Area	Class
Weighted Avg.	0.935	0.275	0.888	0.935	0.911	0.690	0.839	0.881	good
									bad

The 'Confusion Matrix' is:

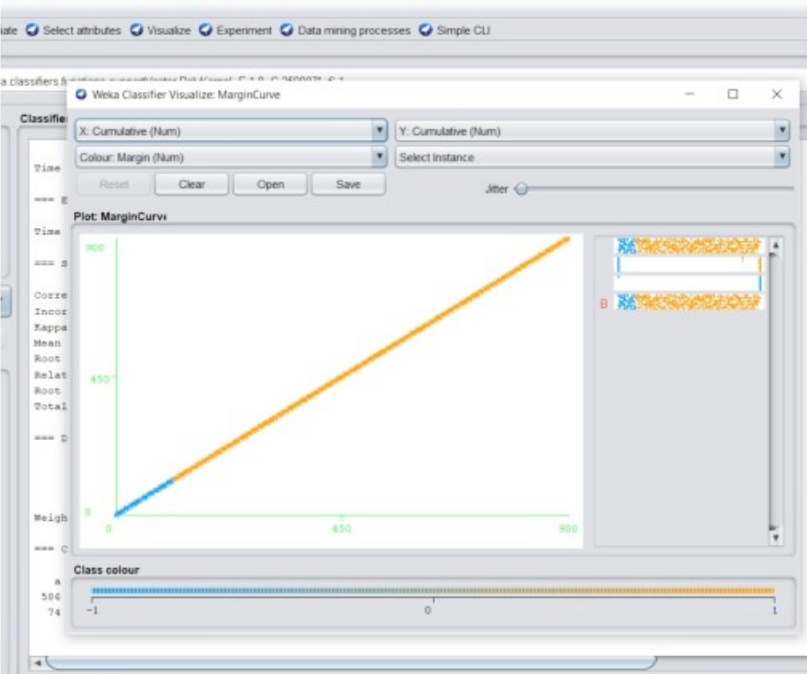
```

a b -- classified as
588 41 | a = good
74 139 | b = bad
  
```


Visualisation of training model

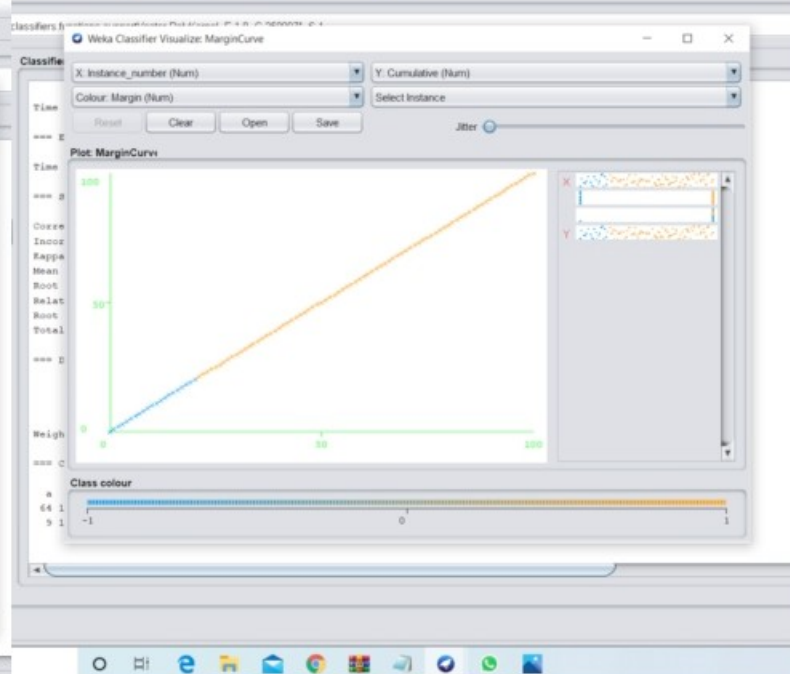


TRAIN: I = 2, BATCH-SIZE=200



TRAIN : I = 3, BATCH-SIZE = 400

TYPE TO ENTER A CAPTION.



TRAIN: I =4, BATCH-SIZE = 400

Testing the model

After training the model we test the model using the test data.

Optimal performance, when I = 5 and batch size = 400

correctly classified instances are 86%

incorrectly classified instances are 14%

When I = 2 and batch size = 400

correctly classified instances are 81%

incorrectly classified instances are 19%

When I = 4 and batch size = 300

correctly classified instances are 85%

incorrectly classified instances are 15%.

=== Run information ===

Scheme: weka.classifiers.functions.VotedPerceptron -I 1 -E 1.0 -S 1 -M 10000

Relation: german_credit-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-

weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-

weka.filters.unsupervised.instance.Resample-S1-Z100.0-

weka.filters.unsupervised.instance.Resample-S1-Z90.0-no-replacement

Instances: 100

Attributes: 21

checking_status

duration

credit_history

purpose

credit_amount

savings_status

employment

installment_commitment

personal_status

other_parties

residence_since

property_magnitude

age

other_payment_plans

housing

existing_credits

job

num_dependents

own_telephone

foreign_worker

class

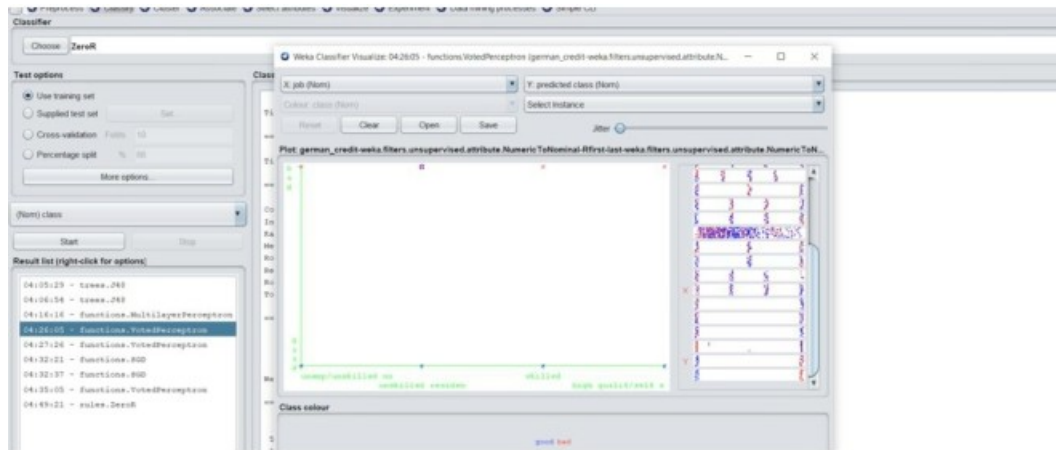
Test mode: evaluate on training data

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

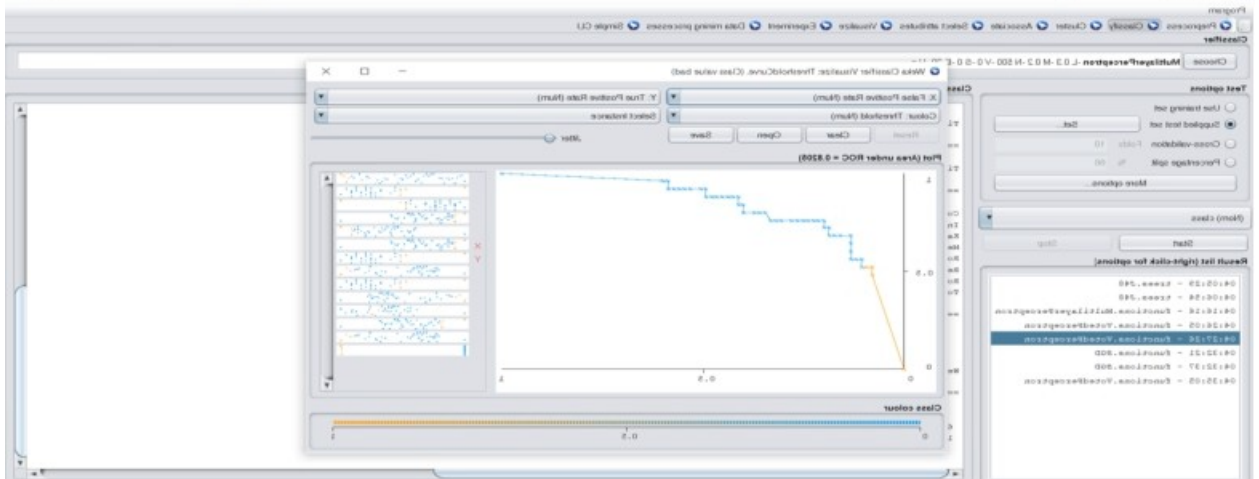
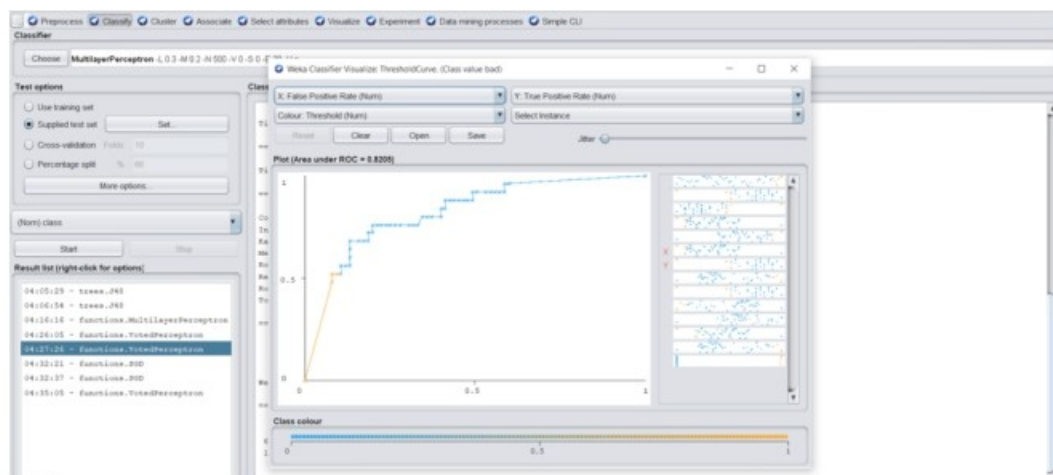
VotedPerceptron: Number of perceptrons=268

Time taken to build model: 0.24 seconds

Visualisation of tested model Confusion matrices:-



I = 5 and batch size = 400



I = 2 and batch size = 400

```

=== Confusion Matrix ===      === Confusion Matrix ===

  a  b  <-- classified as  a  b  <-- classified as
586 41 |  a = good        67  8 |  a = good
 74 199 |  b = bad        7 18 |  b = bad

```

Additional Outputs

URL links to online resources

2)<https://en.wikipedia.org/wiki/>

Perceptron

A reference to Voted Perceptron, an advanced technique from WEKA.

2)https://www.researchgate.net/publication/289318021_Real-time_training_of_Voted_Perceptron_for_classification_of_EEG_data

This link is to a research paper based on Voted Perceptron that was used to gain better insight in the topic.

6. Clustering: K-Means or DBSCAN

Clustering technique used for this dataset is K-Means Algorithm. K-Means Algorithm is an example of unsupervised learning. It starts with a group of selected centroids which are used to form clusters of points and finally we perform the iterations.

The parameters that are being varied to produce different outcomes are Maximum number of iterations(maxIterations) and number of clusters (numClusters).

When maxIterations = 500 and numClusters = 2

then the clustered instances are divided into 65% and 30%

When maxIterations = 600 and numClusters = 3 then the clustered instances are divided into 54%, 29% and 18%

I tried several configurations for k-means clustering. Changing the distance function between euclidean and Manhattan did not change clusters very much. The initialization method had a very big effect on cluster formation. When choosing random, resulting 2 clusters did not correspond appropriate clusters, but choosing farthest first, k-means++ or canopy did give the same clusters corresponding class with very high accuracy. The following results are produced.

The screenshot shows the Weka Workbench interface with the 'Cluster' tab selected. The 'Clusterer' dropdown is set to 'SimpleKMeans'. The 'Clusterer output' pane displays the following data:

checking_status	no checking	no checking	no checking
duration	24	12	24
credit_history	existing paid	existing paid	critical/other existing credit
purpose	radio/tv	radio/tv	new car
credit_amount	1386	1386	1169
savings_status	<100	<100	<100
employment	1<=X<4	1<=X<4	>=7
installment_commitment	4	4	4
personal_status	male single	male single	male single
other_parties	none	none	none
residence_since	4	4	4
property_magnitude	car	car	life insurance
age	26	26	36
other_payment_plan	none	none	none
housing	own	own	own
existing_credits	1	1	2
job	skilled	skilled	skilled
num_dependents	1	1	1
own_telephone	none	none	yes
foreign_worker	yes	yes	yes
class	good	good	good

Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances

0	633 (70%)
1	267 (30%)

Manhattan and Euclidean distance

The screenshot shows the Weka Workbench interface with the 'Cluster' tab selected. The 'Clusterer' dropdown is set to 'SimpleKMeans'. The 'Clusterer output' pane displays the following data:

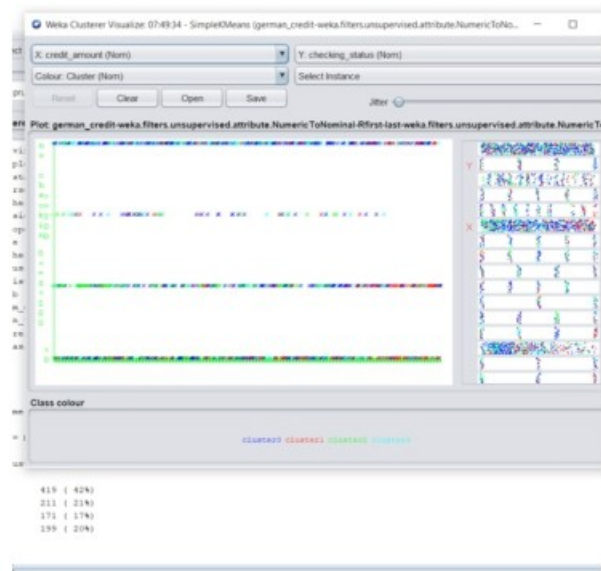
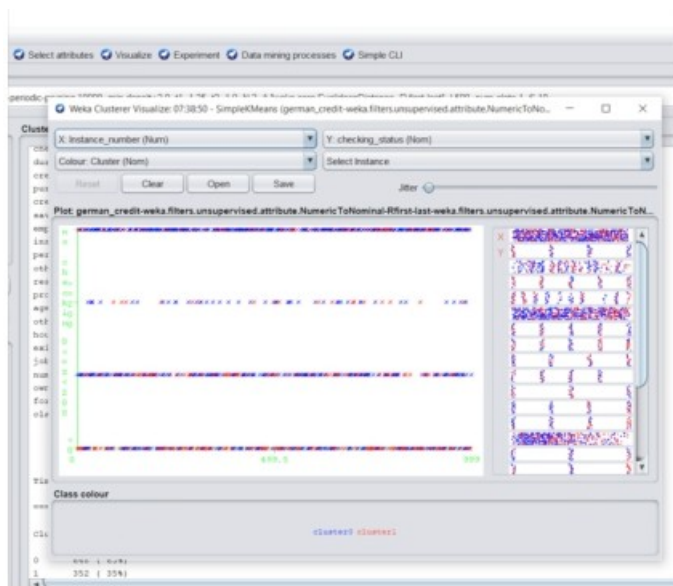
checking_status	no checking	no checking	no checking
duration	24	12	24
credit_history	existing paid	existing paid	critical/other existing credit
purpose	radio/tv	radio/tv	new car
credit_amount	1386	1386	1169
savings_status	<100	<100	<100
employment	1<=X<4	1<=X<4	>=7
installment_commitment	4	4	4
personal_status	male single	male single	male single
other_parties	none	none	none
residence_since	4	4	4
property_magnitude	car	car	life insurance
age	26	26	36
other_payment_plan	none	none	none
housing	own	own	own
existing_credits	1	1	2
job	skilled	skilled	skilled
num_dependents	1	1	1
own_telephone	none	none	yes
foreign_worker	yes	yes	yes
class	good	good	good

Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

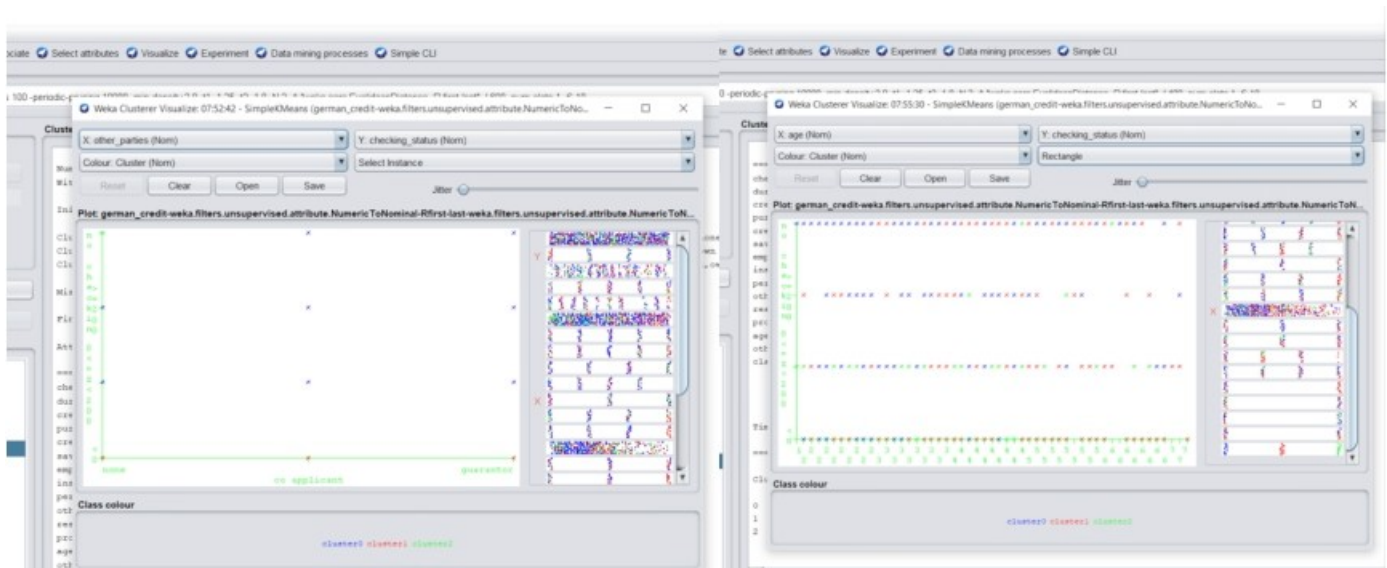
Clustered Instances

0	633 (70%)
1	267 (30%)



maxIterations = 500 and numClusters = 2

maxIterations = 400 and numClusters = 3



maxIterations = 600 and numClusters = 3

The screen shots given below consist of following values of parameters:

1)maxIterations = 500 and numClusters = 2

2) maxIterations = 400 and numClusters = 3 3) maxIterations = 600 and numClusters = 3

Choose **SimpleKMeans** -int 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka core EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- ☒ Use training set
- ☐ Supplied test set
- ☐ Percentage split %
- ☐ Classes to clusters evaluation (None) class
- ☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 07:38:50 - SimpleKMeans

Clusterer output

checking_status	no checking	no checking	<0
duration	24	12	24
credit_history	existing paid	existing paid	existing paid
purpose	radio/tv	radio/tv	new car
credit_amount	1250	1250	727
savings_status	<100	<100	<100
employment	1<=X<4	1<=X<4	>=7
installment_commitment	4	4	4
personal_status	male single	male single	male single
other_parties	none	none	none
residence_since	4	2	4
property_magnitude	car	real estate	no known property
age	27	27	36
other_payment_plans	none	none	none
housing	own	own	own
existing_credits	1	1	1
job	skilled	skilled	skilled
num_dependents	1	1	1
own_telephone	none	none	yes
foreign_worker	yes	yes	yes
class	good	good	good

Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Preprocess Classify Cluster Associate Select attributes Visualize Experiment Data mining processes Simple CLI

Clusterer

Choose **SimpleKMeans** -int 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka core EuclideanDistance -R first-last" -I 600 -num-slots 1 -S 10

Cluster mode

- ☒ Use training set
- ☐ Supplied test set
- ☐ Percentage split %
- ☐ Classes to clusters evaluation (None) class
- ☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 07:38:50 - SimpleKMeans
- 07:47:36 - SimpleKMeans
- 07:49:34 - SimpleKMeans
- 07:51:26 - SimpleKMeans
- 07:52:04 - SimpleKMeans
- 07:52:42 - SimpleKMeans

Clusterer output

credit_amount	1250	1293	727	1154
savings_status	<100	<100	<100	<100
employment	1<=X<4	1<=X<4	>=7	>=7
installment_commitment	4	4	4	4
personal_status	male single	male single	male single	male single
other_parties	none	none	none	none
residence_since	4	2	4	4
property_magnitude	car	car	no known property	real estate
age	27	27	36	23
other_payment_plans	none	none	none	none
housing	own	own	own	own
existing_credits	1	1	1	1
job	skilled	skilled	skilled unskilled resident	
num_dependents	1	1	1	1
own_telephone	none	none	yes	none
foreign_worker	yes	yes	yes	yes
class	good	good	good	good

Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0	537 (54%)
1	287 (29%)
2	176 (18%)

**Additional output of the clustering process
For the above results produced**

Choose SimpleKMeans -int 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 4 -A "weka core EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- ☒ Use training set
- ☐ Supplied test set
- ☐ Percentage split
- ☐ Classes to clusters evaluation
- ☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 07:38:50 - SimpleKMeans
- 07:47:36 - SimpleKMeans
- 07:49:34 - SimpleKMeans

Clusterer output

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (1000.0)	Cluster# 0 (419.0)	1 (211.0)	2 (171.0)	3 (115.0)
checking_status	no checking	no checking	<0	0<=K<200	no che
duration	24	24	24	12	12
credit_history	existing paid	existing paid	existing paid	existing paid	existing
purpose	radio/tv	new car	used car	radio/tv	radio
credit_amount	1258	1275	1159	709	
savings_status	<100	<100	<100	<100	
employment	1<=K<4	1<=K<4	>=7	<1	
installment_commitment	4	4	4	4	
personal_status	male single	female div/dep/mar	male single	male single	male si
other_parties	none	none	none	none	
residence_since	4	2	4	3	
property_magnitude	car	car	no known property	real estate	life insur
age	27	25	35	23	
other_payment_plans	none	none	none	none	
housing	own	own	own	own	
existing_credits	1	1	1	1	
job	skilled	skilled high qualif/self emp/mgmt	unskilled resident		ski
num_dependents	1	1	1	1	
own_telephone	none	none	yes	none	

Preprocess Classify Cluster Associate Select attributes Visualize Experiment Data mining processes Simple CLI

Choose SimpleKMeans -int 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka core EuclideanDistance -R first-last" -I 400 -num-slots 1 -S 10

Cluster mode

- ☒ Use training set
- ☐ Supplied test set
- ☐ Percentage split
- ☐ Classes to clusters evaluation
- ☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 07:38:50 - SimpleKMeans
- 07:47:36 - SimpleKMeans
- 07:49:34 - SimpleKMeans
- 07:51:26 - SimpleKMeans
- 07:52:04 - SimpleKMeans
- 07:52:42 - SimpleKMeans
- 07:55:10 - SimpleKMeans
- 07:56:30 - SimpleKMeans

Clusterer output

	(1000.0)	(472.0)	(399.0)	(129.0)
checking_status	no checking	no checking	<0	0<=K<200
duration	24	12	24	12
credit_history	existing paid	existing paid	existing paid	existing paid
purpose	radio/tv	new car	radio/tv	radio/tv
credit_amount	1258	1258	717	1424
savings_status	<100	<100	<100	<100
employment	1<=K<4	1<=K<4	>=7	<1
installment_commitment	4	4	4	4
personal_status	male single	female div/dep/mar	male single	male mar/wid
other_parties	none	none	none	none
residence_since	4	2	4	3
property_magnitude	car	car	car	real estate
age	27	27	34	24
other_payment_plans	none	none	none	none
class	good	good	good	good

Time taken to build model (full training data) : 0.01 seconds

== Model and evaluation on training set ==

Clustered Instances

	0	1	2
0	472 (47%)		
1	399 (40%)		
2	129 (13%)		

Choose SimpleKMeans -int 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka core EuclideanDistance -R first-last" -I 400 -num-slots 1 -S 10

Cluster mode

- ☒ Use training set
- ☐ Supplied test set
- ☐ Percentage split
- ☐ Classes to clusters evaluation
- ☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 07:38:50 - SimpleKMeans
- 07:47:36 - SimpleKMeans
- 07:49:34 - SimpleKMeans
- 07:51:26 - SimpleKMeans
- 07:52:04 - SimpleKMeans
- 07:52:42 - SimpleKMeans
- 07:55:10 - SimpleKMeans
- 07:56:30 - SimpleKMeans

Clusterer output

Number of iterations: 4

Within cluster sum of squared errors: 7679.0

Initial starting points (random):

Cluster 0: 'no checking',34,'critical/other existing credit','new car',7855,<100,1<=K<4,4,'female div/dep/mar',none,2,'real estate',25,stores,bad

Cluster 1: '<0,24,'critical/other existing credit','used car',6615,<100,unemployed,2,'male single',none,4,'no known property',75,none,good

Cluster 2: '0<=K<200,12,'existing paid',radio/tv,1155,<100,>=7,3,'male mar/wid',guarantor,3,'real estate',40,bank,good

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (1000.0)	Cluster# 0 (472.0)	1 (399.0)	2 (129.0)
checking_status	no checking	no checking	<0	0<=K<200
duration	24	12	24	12
credit_history	existing paid	existing paid	existing paid	existing paid
purpose	radio/tv	new car	radio/tv	radio/tv
credit_amount	1258	1258	717	1424
savings_status	<100	<100	<100	<100
employment	1<=K<4	1<=K<4	>=7	<1
installment_commitment	4	4	4	4
personal_status	male single	female div/dep/mar	male single	male mar/wid

Url links

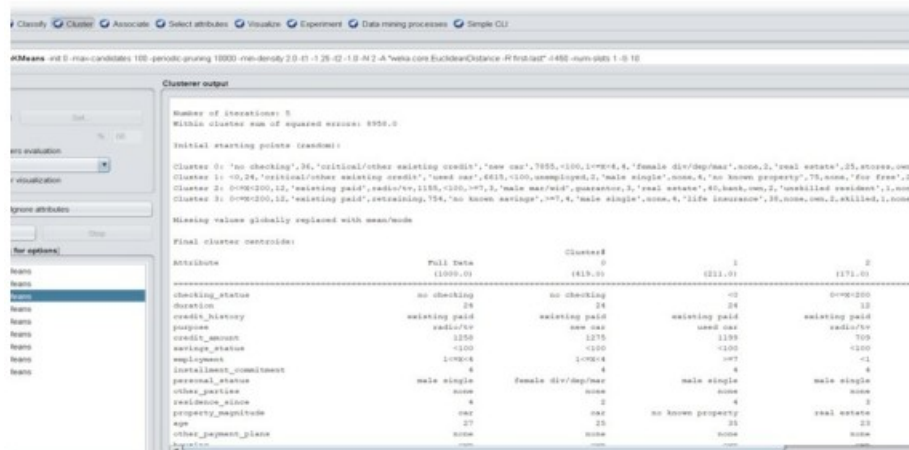
- 1) <http://facweb.cs.depaul.edu/mobasher/classes/ect584/WEKA/k-means.html>

This link references to K-Means Clustering in WEKA. Illustrations have been used to explain the concept in depth

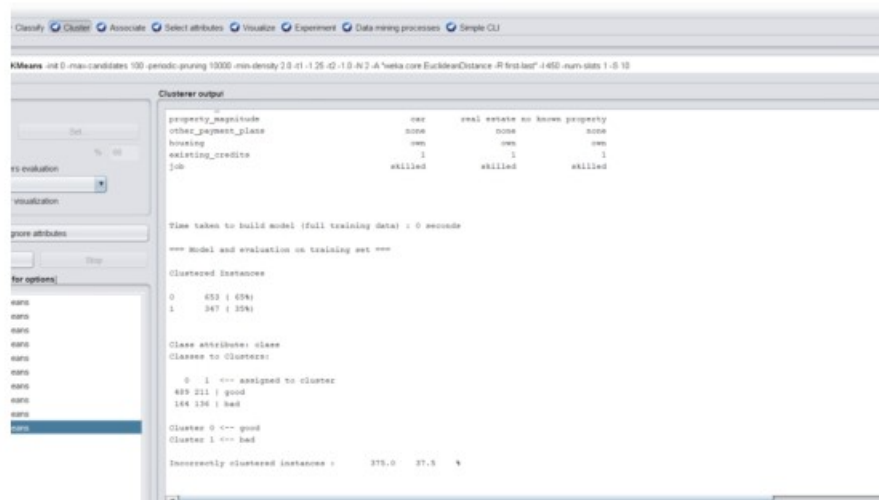
2) <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>

This article is an explanation of the K-Means Clustering Algorithm with examples of the same.

Evaluate the clusters using the classes to clusters evaluation



maxIterations = 100 and numClusters = 2



maxIterations = 600 and numClusters = 2

Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Relation: german_credit-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.instance.Resample-S1-Z100.0-weka.filters.unsupervised.instance.Resample-S1-Z90.0-no-replacement

Instances: 900

Attributes: 21

checking_status

duration

credit_history

purpose

credit_amount

savings_status

employment

installment_commitment

personal_status

other_parties

residence_since

property_magnitude

age

other_payment_plans

housing

existing_credits

job

num_dependents

own_telephone

foreign_worker

Ignored:

class

Test mode: Classes to clusters evaluation on training data

=== Clustering model (full training set) ===

kMeans

=====

Number of iterations: 4

Within cluster sum of squared errors: 8123.0

Initial starting points (random):

Cluster 0: <0,12,'existing paid',education,684,<100,1<=X<4,4,'male single',none,4,car,40,none,rent,1,'unskilled resident',2,none,yes

Cluster 1: 0<=X<200,18,'critical/other existing credit',furniture/equipment,7374,<100,unemployed,4,'male single',none,4,'life insurance',40,stores,own,2,'high qualif/self emp/mgmt',1,yes,yes

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Cluster#		
	Full Data	0	1
	(900.0)	(630.0)	(270.0)
=====			
checking_status	no checking	no checking	no
checking	24	12	24
credit_history	existing paid	existing paid	critical/other existing
credit purpose	radio/tv	radio/tv	new car
credit_amount	1386	1386	1169
savings_status	<100	<100	<100
employment	1<=X<4	1<=X<4	>=7
installment_commitment.	4	4	4
personal_status	male single	male single	male single
other_parties	none	none	none
residence_since	4	4	4
property_magnitude	car	real estate	life insurance
age	26	26	36
other_payment_plans	none	none	none
housing	own	own	own
existing_credits	1	1	2
job	skilled	skilled	skilled
num_dependents	1	1	1

own_telephone	none	none	yes
foreign_worker	yes	yes	yes

duration

7. TimeSeries Forecasting

In Time Series Forecasting prediction about the future is done using a method called Extrapolation using time series data. We are basically analysing the existing trends and predicting if the current scenario will continue in the future.

In this dataset the Time Series Forecasting uses 7 attributes:-

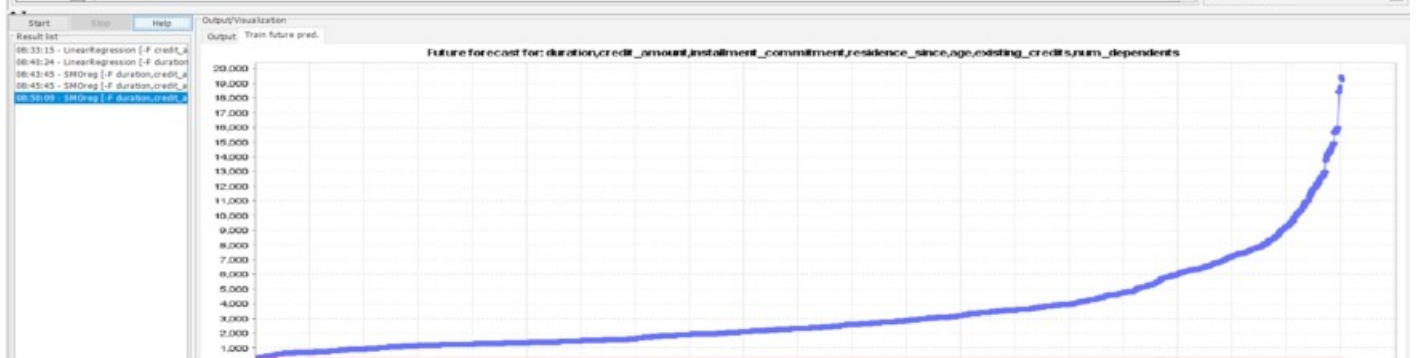
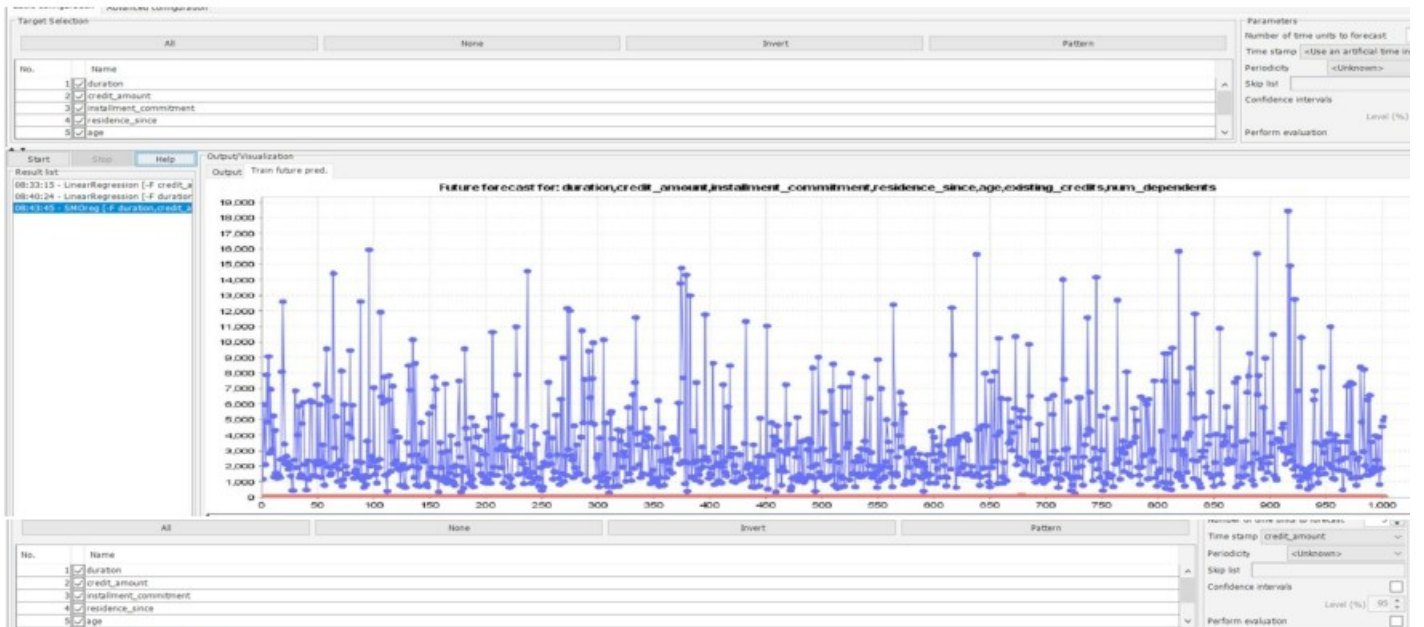
- 1) Duration
- 2) Credit_Amount
- 3) Installment_Commitment
- 4) Residence_Since
- 5) Age
- 6) Existing_Credits
- 7) Num_dependents

Different regression methods are used for time series forecasting

- 1) Linear Regression
- 2) SMOReg
- 3) Gaussian Process
- 4) Additive Regression

Only the attribute Credit_Amount can be used to predict a foul transaction.

Diagrams and Visualization historical Values and predictions:1)Linear Regression



Results

AllNoneInvertPattern

No.	Name
1	<input checked="" type="checkbox"/> duration
2	<input checked="" type="checkbox"/> credit_amount
3	<input checked="" type="checkbox"/> installment_commitment
4	<input checked="" type="checkbox"/> residence_since
5	<input checked="" type="checkbox"/> age

Time stamp: <Use an artificial time inde...>
Periodicity: <Unknown>
Skip list:
Confidence intervals: ☐
Level (%): 95
Perform evaluation: ☐

StartStopHelp

Output/Visualization

Result list

Output	Train	future	pred.
08:33:15 - LinearRegression [-F credit_a	576	24	1258
08:40:24 - LinearRegression [-F duration	577	6	753
	578	18	2427
	579	24	2538
	580	15	1264
	581	30	8386
	582	48	4844
	583	21	2923
	584	36	8229
	585	24	2028
	586	15	1433
	587	42	6289
	588	13	1409
	589	24	6579
	590	24	1743
	591	12	3565
	592	15	1569
	593	18	1936
	594	36	3959
	595	12	2390
	596	12	1736
	597	30	3857

Basic configurationAdvanced configuration

Target Selection

AllNoneInvertPattern

No.	Name
1	<input checked="" type="checkbox"/> credit_amount
2	<input checked="" type="checkbox"/> installment_commitment
3	<input checked="" type="checkbox"/> residence_since
4	<input checked="" type="checkbox"/> age
5	<input checked="" type="checkbox"/> existing_credits

Parameters
Number of time units to forecast: 2
Time stamp: <Use an artificial time inde...>
Periodicity: <Unknown>
Skip list:
Confidence intervals: ☐
Level (%): 95
Perform evaluation: ☐

StartStopHelp

Output/Visualization

Result list

Output	Train	future	pred.
08:33:15 - LinearRegression [-F credit_a	576	3	57
08:39:41 - LinearRegression [-F credit_a	577	2	64
	578	4	42
	579	4	47
	580	2	25
	581	2	49
	582	3	33
	583	1	28
	584	2	26
	585	2	30
	586	4	25
	587	2	33
	588	4	64

Target Selection

AllNoneInvertPattern

No.	Name
1	<input checked="" type="checkbox"/> duration
2	<input checked="" type="checkbox"/> credit_amount
3	<input checked="" type="checkbox"/> installment_commitment
4	<input checked="" type="checkbox"/> residence_since
5	<input checked="" type="checkbox"/> age

Parameters
Number of time units to forecast: 1
Time stamp: <None>
Periodicity: <Unknown>
Skip list:
Confidence intervals: ☒
Level (%): 95
Perform evaluation: ☒

StartStopHelp

Output/Visualization

Result list

Output	Train	future	pred.
08:33:15 - LinearRegression [-F credit_a	594	60	14782
08:40:24 - LinearRegression [-F duration	595	6	14896
08:43:45 - SMOReg [-F duration,credit_a	596	60	15653
08:45:45 - SMOReg [-F duration,credit_a	597	48	15672
08:50:09 - SMOReg [-F duration,credit_a	598	36	15857
08:53:54 - GaussianProcesses [-F durati	599	54	15945
08:55:10 - AdditiveRegression [-F durati	1000	48	18424
08:56:04 - AdditiveRegression [-F durati	1001*	57.4295	18597.4992
08:59:38 - AdditiveRegression [-F durati			1.494
09:01:01 - LinearRegression [-F duration			
09:02:27 - GaussianProcesses [-F durati			
09:03:42 - LinearRegression [-F credit_a			
09:04:02 - LinearRegression [-F duration			
09:04:29 - LinearRegression [-F duration			
09:04:47 - LinearRegression [-F duration			

=== Evaluation on training data ===

Target

1-step-ahead

=====

duration

N

988

Mean absolute error

7.0231

Root mean squared error

9.226

credit_amount

N

988

Mean absolute error

24.1828

Root mean squared error

76.1078

existing_credits

N

988

Mean absolute error

0.5061

- - -

1) LINEAR REGRESSION AND ITS PREDICTIONS

All										Number of time units to forecast	
None										Time stamp credit_amount	
Invert										Periodicity Daily	
Pattern										Skip list	
No.	Name										Confidence intervals
3	installment_commitment										Level (%) 95
4	residence_since										Perform evaluation
5	age										
6	existing_credits										
7	num_dependents										

Start	Stop	Help	Output/Visualization									
Result list			Output Train future pred.									
08:33:15	LinearRegression [-F credit_a		17987.3874	39	11760	2	3	32	1	1		
08:40:24	LinearRegression [-F duration		18005.5796	45	11816	2	4	29	2	1		
08:43:45	SMOreg [-F duration,credit_a		18023.7718	24	11938	2	3	25	2	2		
08:45:45	SMOreg [-F duration,credit_a		18041.964	30	11998	1	1	34	1	1		
08:50:09	SMOreg [-F duration,credit_a		18060.1562	48	12169	4	4	36	1	1		
08:53:54	GaussianProcesses [-F durati		18078.3483	48	12204	2	2	48	1	1		
08:55:18	AdditiveRegression [-F durati		18096.5405	36	12389	1	4	37	1	1		
			18114.7327	24	12579	4	2	44	1	1		
			18132.9249	36	12612	1	4	47	1	2		
			18151.1171	21	12680	4	4	30	1	1		
			18169.3093	48	12749	4	1	37	1	1		
			18187.5015	18	12976	3	4	38	1	1		
			18205.6937	60	13756	2	4	63	1	1		
			18223.8859	60	14027	4	2	27	1	1		
			18242.0781	39	14179	4	4	30	2	1		
			18260.2703	36	14318	4	2	57	1	1		
			18278.4625	48	14421	2	2	25	1	1		
			18296.6547	6	14555	1	2	23	1	1		
			18314.8469	60	14782	3	4	60	2	1		
			18333.039	6	14896	1	4	68	1	1		
			18351.2312	60	15653	2	4	21	2	1		
			18369.4234	48	15672	2	2	23	1	1		
			18387.6156	36	15857	2	3	43	1	1		
			18405.8078	54	15945	3	4	58	1	1		

ADDITIVE REGRESSION AND SMOREG

All										Number of time units to forecast	
None										Time stamp credit_amount	
Invert										Periodicity <Unknown>	
Pattern										Skip list	
No.	Name										Confidence intervals
3	installment_commitment										Level (%) 95
4	residence_since										Perform evaluation
5	age										
6	existing_credits										
7	num_dependents										

Start	Stop	Help	Output/Visualization									
Result list			Output Train future pred.									
08:33:15	LinearRegression [-F credit_a		18023.7718	24	11938	2	3	39	2	2		
08:40:24	LinearRegression [-F duration		18041.964	30	11998	1	1	34	1	1		
08:43:45	SMOreg [-F duration,credit_a		18060.1562	48	12169	4	4	36	1	1		
08:45:45	SMOreg [-F duration,credit_a		18078.3483	48	12204	2	2	48	1	1		
08:50:09	SMOreg [-F duration,credit_a		18096.5405	36	12389	1	4	37	1	1		
08:53:54	GaussianProcesses [-F durati		18114.7327	24	12579	4	2	44	1	1		
08:55:18	AdditiveRegression [-F durati		18132.9249	36	12612	1	4	47	1	2		
			18151.1171	21	12680	4	4	30	1	1		
			18169.3093	48	12749	4	1	37	1	1		
			18187.5015	18	12976	3	4	38	1	1		
			18205.6937	60	13756	2	4	63	1	1		
			18223.8859	60	14027	4	2	27	1	1		
			18242.0781	39	14179	4	4	30	2	1		
			18260.2703	36	14318	4	2	57	1	1		
			18278.4625	48	14421	2	2	25	1	1		
			18296.6547	6	14555	1	2	23	1	1		
			18314.8469	60	14782	3	4	60	2	1		
			18333.039	6	14896	1	4	68	1	1		
			18351.2312	60	15653	2	4	21	2	1		
			18369.4234	48	15672	2	2	23	1	1		
			18387.6156	36	15857	2	3	43	1	1		
			18405.8078	54	15945	3	4	58	1	1		
			18424	2	16424	1	2	32	1	1		
			18442.1922*	4	15792.0208	2.5142	2.4514	55.4449	1.4387	1.4147		
			18460.3844*	26	15792.0209	2.5142	1.4847	56.9424	1.4391	1.4147		
			18478.5766*	39	15792.0208	2.0896	2.2791	26.5462	1.4376	1.0282		
			18496.7688*	39	15792.0208	2.0896	1.9492	26.5462	1.4104	1.0282		

All										Number of time units to forecast	
None										Time stamp <Use an artificial time inde...	
Invert										Periodicity <Unknown>	
Pattern										Skip list	
No.	Name										Confidence intervals
1	duration										Level (%) 95
2	credit_amount										Perform evaluation
3	installment_commitment										
4	residence_since										
5	age										

Start	Stop	Help	Output/Visualization									
Result list			Output Train future pred.									
08:33:15	LinearRegression [-F credit_a		976	24	1258	3	3	57	1	1		
08:40:24	LinearRegression [-F duration		977	6	753	2	3	44	1	1		
08:43:45	SMOreg [-F duration,credit_a		978	18	2427	4	2	42	2	1		
			979	24	2538	4	4	47	2	2		
			980	15	1264	2	2	25	1	1		
			981	30	8386	2	2	49	1	1		
			982	48	4844	3	2	33	1	1		
			983	21	2923	1	1	28	1	1		
			984	36	8229	2	2	26	1	2		
			985	24	2028	2	2	30	2	1		
			986	15	1433	4	3	25	2	1		
			987	42	4289	2	1	33	2	1		
			988	13	1409	2	4	44	1	1		
			989	24	6579	4	2	29	1	1		
			990	24	1743	4	2	48	2	1		
			991	12	3565	2	1	37	2	2		
			992	15	1569	4	4	24	1	2		
			993	18	1936	2	4	23	2	1		
			994	36	3955	4	3	50	1	1		
			995	12	2390	4	3	50	1	1		
			996	12	1736	3	4	31	1	1		
			997	30	3857	4	4	40	1	1		
			998	12	804	4	4	38	1	1		
			999	45	1845	4	4	23	1	1		
			1000	45	4576	3	4	27	1	1		
			1001*	33.6613	4961.4472	1.7471	2.5222	25.8719	1.22	1.001		
			1002*	31.104	5162.4715	0.726	3.4155	43.0355	0.7977	1.0011		

Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. A [scatterplot](#) can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the [correlation coefficient](#), which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form $Y = a + bX$, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b , and a is the intercept (the value of y when $x = 0$).

Least-Squares Regression

The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values.

Example

The dataset "Televisions, Physicians, and Life Expectancy" contains, among other variables, the number of people per television set and the number of people per physician for 40 countries. Since both variables probably reflect the level of wealth in each country, it is reasonable to assume that there is some positive association between them. After removing 8 countries with missing values from the dataset, the remaining 32 countries have a correlation coefficient of 0.852 for number of people per television set and number of people per physician. The r^2 value is 0.726 (the square of the correlation coefficient), indicating that 72.6% of the variation in one variable may be explained by the other (Note: see [correlation](#) for more detail.) Suppose we choose to consider number of people per television set as the explanatory variable, and number of people per physician as the dependent variable. Using the MINITAB "REGRESS" command gives the following results:

The regression equation is $\text{People.Phys.} = 1019 + 56.2 \text{ People.Tel.}$

To view the fit of the model to the observed data, one may plot the computed regression line over the actual data points to evaluate the results. For this example, the plot appears to the right, with number of individuals per television set (the explanatory variable) on the x-axis and number of individuals per physician (the dependent variable) on the y-axis. While most of the data points are clustered towards the lower left corner of the plot (indicating relatively few individuals per television set and per physician), there are a few points which lie far away from the main cluster of the data. These points are known as *outliers*, and depending on their location may have a major impact on the regression line (see below).



REGRESSION EQUATION

URL Link to Online Resources

- 1) <https://machinelearningmastery.com/time-series-forecasting/>

A detailed article explaining the concept of Time Series Forecasting and more information on Time Series as a whole.

8) Research Publication Summary and relevance / potential relevance to your work

A] Publication & Researchers

Citation

Gurram Sai Kumar. " Credit Card Fraud Detection System Based On Machine Learning Techniques." IOSR Journal of Computer Engineering (IOSR-JCE) 21.3 (2019): 45-52.

Publication - IOSR Journal of Computer Engineering (IOSR-JCE)

Researchers - Gurram Sai Kumar, Madala Vekaiah Naidu, Dr. Mandugula Sujatha

B] Dataset

- German Credit Card dataset obtained from the UCI (University of California, Irvine) machine learning repository

C] Technique (mention any adaptations)

In this research, ensemble models were the majority concerned.

Bagging is an ensemble algorithm that is used to improve factors such as stability and accuracy of a machine learning algorithm. Random Forest is another ensemble algorithm that helps to identify relevant predictor variables to make feature selection easier.

eXtreme Gradient Boosting (XGBoost) is a kind of GBM model that follows the principle of gradient boosting. The differences in modelling details that exist are that XGBoost uses a more regularized model formalization to control over-fitting which helps achieve better performance.

Light Gradient Boosting Machine (LightGBM) is a gradient boosting framework that works upon tree-based algorithms. Given its highly efficient and scalable behaviour it is capable of supporting many different GBM Algorithms. It is several times faster than most existing implementations of gradient boosting trees which is backed by its fully greedy tree-growth method, histogram-based memory and computation optimization.

Adaptive Boosting(AdaBoost) has been used as part of the implementation method in order to boost the performance of decision tree and has been implemented in WEKA(Waikato Environment for Knowledge Analysis). This boosting algorithm can be applied to any classifier's learning algorithm.

D] Major Findings

Two different forms of experimental results have been provided, which are:-

- 1) Experimental results of decision tree without any boosting techniques
- 2) Experimental results of the decision tree together with AdaBoost.

The proposed algorithm is a clear cut enhancement of the already available algorithms. Application of boosting algorithms in combination with these algorithms gives much higher accuracy. On analysis of various other features such as Sensitivity and Specificity, it can clearly be derived that the proposed algorithm does a better job than any existing technique. The only limitation

encountered in the proposed algorithm is its non applicability to Linear data.

Fraud Type : Wrong CVV						
ID	Card Number	User Name	Bank Name	Fraud Amount	WebSite	Date
5	350881406571	Praniti	Canara Bank	18000	Flipkart	31/10/2018 18:34:55
9	536470266101	Roshan	Indian Bank	10000	Flipkart	01/11/2018 11:55:17
10	537785904513	Shanmukh	Indian Bank	18000	Flipkart	01/11/2018 12:02:32
21	537785904513	Shanmukh	Indian Bank	18000	Flipkart	01/11/2018 12:21:12
24	646597512025	Sujan	SBI Bank	18000	Flipkart	01/11/2018 12:35:34
25	642855074991	Ashwin	SBI Bank	10000	Flipkart	01/11/2018 12:38:27

Fig3. Fraud Result with Wrong CVV

Fraud Type : Expired Card						
ID	Card Number	User Name	Bank Name	Fraud Amount	WebSite	Date
2	536470266101	Roshan	Indian Bank	10000	Flipkart	31/10/2018 18:32:54
6	320622743637	Sanjay	Corporation Bank	10000	Flipkart	01/11/2018 11:28:27
7	320622743637	Sanjay	Corporation Bank	10000	Flipkart	01/11/2018 11:30:20
11	537785904513	Shanmukh	Indian Bank	10000	Flipkart	01/11/2018 12:03:33
16	537785904513	Shanmukh	Indian Bank	18000	Flipkart	01/11/2018 12:08:28
22	537785904513	Shanmukh	Indian Bank	18000	Flipkart	01/11/2018 12:22:09
32	649942232755	Shivaji	SBI Bank	35000	Flipkart	01/11/2018 13:38:08

Fig4. Fraud Result with Expired Card

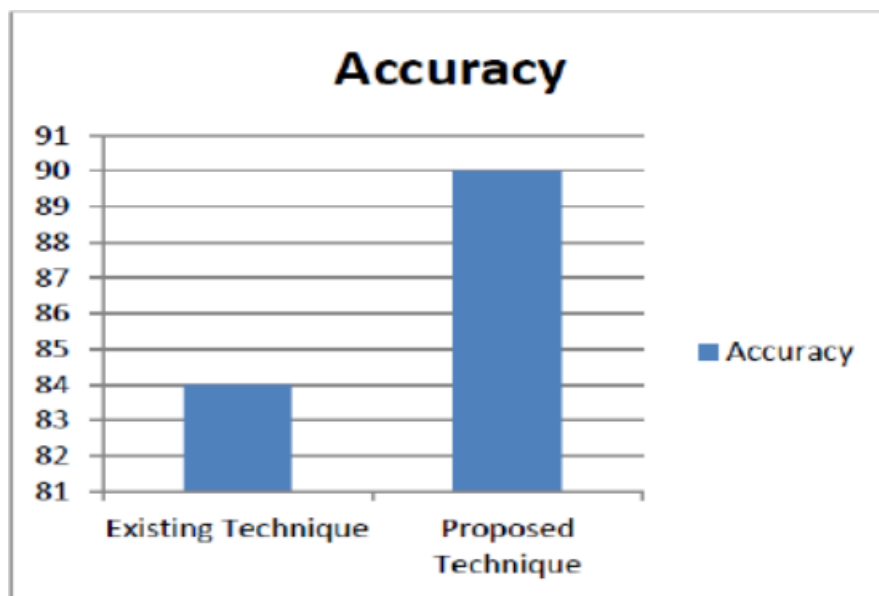


Figure 5: Accuracy Comparison

	KNN	Random Tree	Proposed Algorithm
Accuracy	0.9691	0.9432	0.9824
Sensitivity	0.8835	0	0.9767
Specificity	0.9711	0	0.9824
Limitations	Cannot detect the fraud at the time of transaction	Not suitable for Randomness in dataset	Not Applied for non-Linear data

TABLE: COMPARISION OF ALGORITHMS

E] Relevance / potential relevance to your work

The research paper gives an insight into making the existing algorithms better by addition of boosting algorithms. Since, we have used the basic techniques such as J48 tree, Voted Perceptrons, K-Means and Time Series Forecasting, we have only looked into the basic existing techniques that are possible to solve the problem of Credit Card Fraud. Adding of Boosting Algorithms to the techniques analysed in this report would be a beneficial task. We can further investigate into ways to enhance the current models and come up with the best approach after analysing it on various factors of accuracy , sensitivity and specificity.

REFERENCE - <http://www.iosrjournals.org/iosr-jce/papers/Vol21-issue3/Series-5/H2103054552.pdf>