

Data Mining Lab File



Submitted to - MRS Rashmi Chaudary

Name - Shubham Jha

Branch - COE

Section - 3

Semester - 5

Roll No - 2019UCO1730

Index

S.No.	Practical
1	Understanding Data Mining & Visualizing data
2	Analysis of Data - Know the type of data - nominal, ordinal, ratio, interval
3	Statistical Data Analysis - Find the mean, median, variance and standard deviation of data
4	Proximity Measures - Calculate dissimilarity matrix in any programming language of choice
5	Data Preprocessing - (a) Handling missing values using various techniques, finding outliers in data, discretisation (b) Normalization and standardization of data
6	Dimensionality reduction using Principal components (PCA)
7	Classification using Decision trees
8	Regression Tasks - Implement linear, logistic regression. Calculate correlation matrix for numerical attributes.
9	Association rule mining - Apriori, FP growth
10	Visualization Techniques - Explore data using various visualization techniques available in Python.
11	Use Bayesian Learning for classification
12	Implement various clustering algorithms - K-means, Hierarchical, EM clustering

1) Understanding Data Mining & Visualizing data

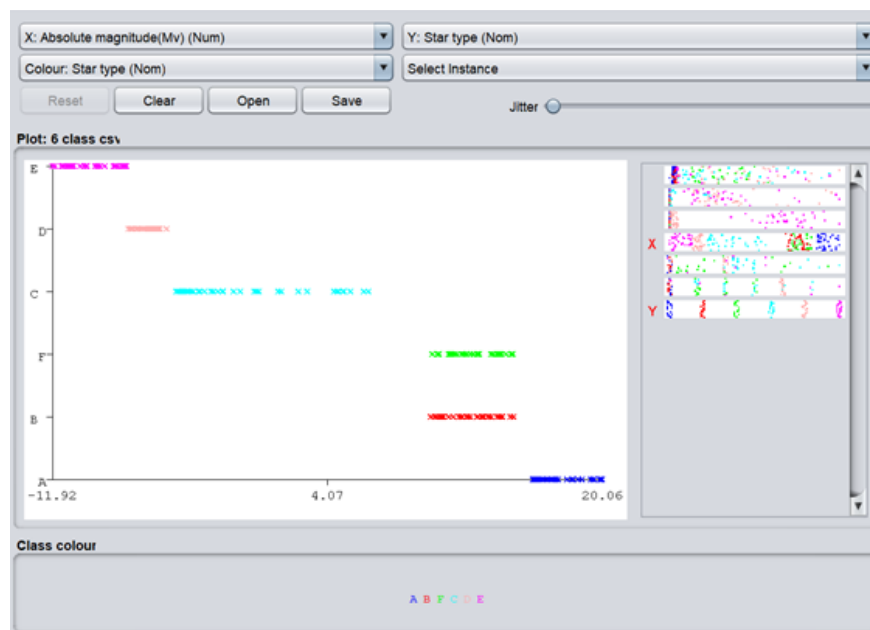
Data Mining

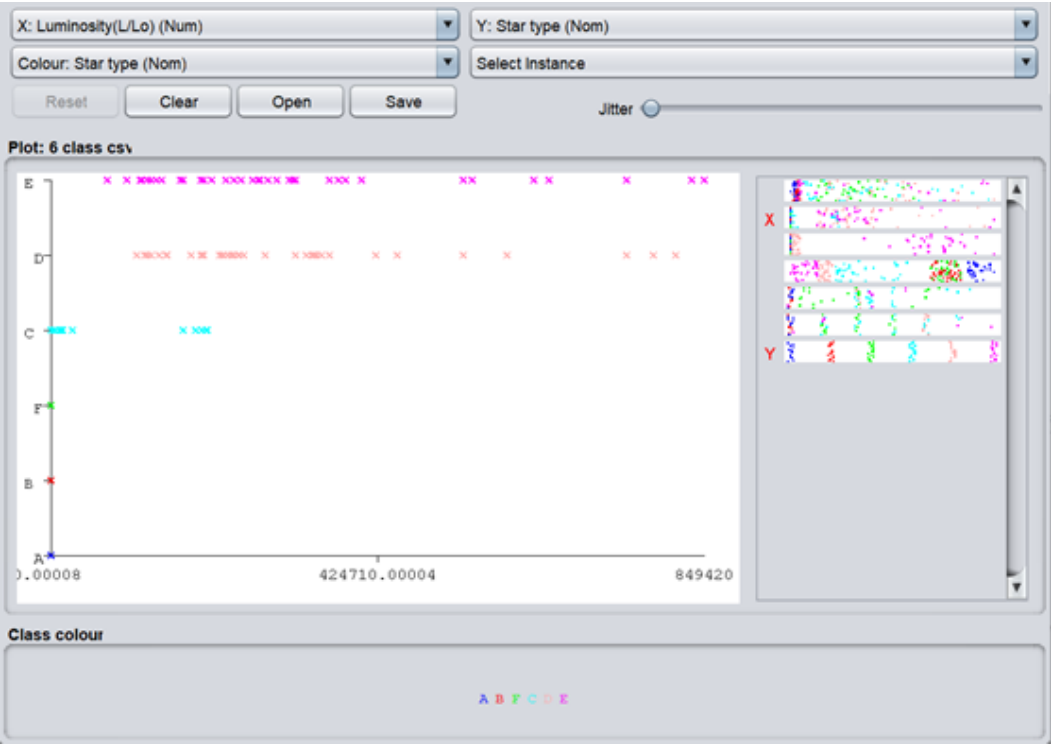
Data mining, also known as knowledge discovery in data, is the process of extracting patterns and other useful information from large data sets (KDD). The usage of data mining techniques has grown in recent decades, assisting organisations in turning raw data into valuable knowledge, thanks to advancements in data warehousing technologies and the rise of big data. Executives still face scalability and automation difficulties, despite the fact that technology is constantly advancing to manage enormous amounts of data.

Data mining has improved corporate decision-making through clever data analytics. The data mining approaches employed in these research can be divided into two groups: those that describe the target dataset and those that use machine learning algorithms to anticipate results. These tactics are used to organise and filter data, providing the most important information, from fraud detection to user behaviours, bottlenecks, and even security breaches.

When integrated with data analytics and visualisation technologies like Apache Spark, delving into the domain of data mining has never been easier, and collecting significant insights has never been faster. Artificial intelligence advances are hastening implementation in a variety of industries.

Visualising Data





2) Analysis of Data - nominal, ordinal, ratio, interval

Nominal

A nominal scale is used to express a variable that has no natural order or ranking. If you choose, you can code nominal variables with numbers, but the sequence is arbitrary, and any computations, such as estimating a mean, median, or standard deviation, would be useless. Examples of nominal variables include: genotype, blood type, zip code, gender, race, eye color, political party

Ordinal

An ordinal scale is one where the order matters but not the difference between values. Examples of ordinal variables include: socio economic status ("low income", "middle income", "high income"), education level ("high school", "BS", "MS", "PhD"), income level ("less than 50K", "50K-100K", "over 100K"), satisfaction rating ("extremely dislike", "dislike", "neutral", "like", "extremely like").

Interval

An interval scale is one where there is order and the difference between two values is meaningful. Examples of interval variables include: temperature (Fahrenheit), temperature (Celsius), pH, SAT score (200-800), credit score (300-850).

Ratio

A ratio variable, has all the properties of an interval variable, and also has a clear definition of 0.0. When the variable equals 0.0, there is none of that variable.

Examples of ratio variables include: enzyme activity, dose amount, reaction rate, flow rate, concentration, pulse, weight, length, temperature in Kelvin (0.0 Kelvin really does mean "no heat"), survival time.

3) Find the mean, median, variance and standard deviation of data

```
#include <bits/stdc++.h>

using namespace std;

double meancal(double *arr, int n){
    double sum=accumulate(arr, arr+n, 0.0);
    return sum/n;
}

double median(double *arr, int n){
    if(n%2==1)
        return (double)(arr[n/2]);
    return (double)((arr[n/2]+arr[n/2-1])/2);
}

double variance(double *arr, int n, double mean){
    double t=0.0;
    for(int i=0; i<n; i++){
        double temp=abs(arr[i]-mean);
        t+=temp*temp;
    }
    if(n>50)
        return t/(n-1);
    else
        return t/n;
}

int main() {
```

```

cout<<"Enter size of data: ";

int n;

cin>>n;

cout<<"\nEnter data: ";

double arr[n];

for(int i=0; i<n; i++)

{
    cin>>arr[i];
}

sort(arr, arr+n);

cout<<endl<<"Entered data after sorting is:\n";

for(int i=0; i<n; i++)

cout<<arr[i]<<" ";

cout<<endl;

double mean;

mean=meancal(arr, n);

cout<<"\nMean of dataset is: "<<mean<<endl;

cout<<"Median of dataset is: "<<median(arr, n)<<endl;

double var=variance(arr, n, mean);


cout<<"\nVariance of dataset is: "<<var<<endl;

cout<<"\nStandard deviation of dataset is: "<<sqrt(var)<<endl;

return 0;

}

```

 **OnlineGDB** beta

online compiler and debugger for c/c++

code, compile, run, debug, share.

IDE

My Projects




Classroom new

Learn Programming

Programming Questions








Sign Up

Login

   124K

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy

© 2016 - 2021 GDB Online

 Run  Debug  Stop  Share  Save  Beautify 

main.cpp

```
40     cin>>arr[i];
41 }
42 sort(arr, arr+n);
43 cout<<endl<<"Entered data after sorting is:\n";
```

input

Enter size of data: 17

Enter data: 1 2 1 9.3 2.2 3 4 2.0 2.9 2.5 8 5 5.5 4.3 2 9.9 7.5

Entered data after sorting is:

1 1 2 2 2 2.2 2.5 2.9 3 4 4.3 5 5.5 7.5 8 9.3 9.9

Mean of dataset is: 4.24118

Median of dataset is: 3

Variance of dataset is: 7.71772

Standard deviation of dataset is: 2.77808

...Program finished with exit code 0

Press ENTER to exit console.

4) Proximity Measures - Calculate dissimilarity matrix

The pairwise differentiation between M items is described by the dissimilarity matrix (also known as the distance matrix). It's a square symmetrical MxM matrix with the value of a chosen measure of differentiation between the (i)th and (j)th object as the (ij)th member. The diagonal elements are either ignored or equal to zero, implying that the difference between an item and itself is assumed to be zero.

The similarity matrix is a similarly related and opposing idea. For the same data, both sorts of descriptions are frequently utilised.

Any valid measure of dissimilarity, including subjective dissimilarity scores, may be employed. The only stipulation is that the bigger the difference between two objects, the higher the measure of dissimilarity's worth.

Code:

```
#include<bits/stdc++.h>
#define N 3
#define M 4
using namespace std;
void printDistance(int mat[N][M])
{
    int ans[N][M];
    for (int i = 0; i < N; i++)
        for (int j = 0; j < M; j++)
            ans[i][j] = INT_MAX;
    for (int i = 0; i < N; i++)
        for (int j = 0; j < M; j++)
        {
            for (int k = 0; k < N; k++)
                for (int l = 0; l < M; l++)
                {
                    if (mat[k][l] == 1)
                        ans[i][j] = min(ans[i][j], abs(i-k) + abs(j-l));
                }
        }
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < M; j++)
            cout << ans[i][j] << " ";
        cout << endl;
    }
}
int main()
{
```

```
int mat[N][M] =  
{  
    0, 0, 0, 1,  
    0, 0, 1, 1,  
    0, 1, 1, 0  
};  
printDistance(mat);  
return 0;  
}
```

5) Data Preprocessing

Preprocessing data is a crucial task. It is a data mining approach that converts unstructured data into a format that is more comprehensible, usable, and efficient.

Tasks in data preprocessing -

- 1) Data Cleaning
- 2) Data Integration
- 3) Data Transformation
- 4) Data Reduction
- 5) Data Discretization

Handling Missing Values -

In a data set, missing values cannot be checked. They must be dealt with. A number of models also don't like missing values. There are numerous methods for dealing with missing data, and picking the proper one is crucial. The strategy used to deal with missing data is determined by the problem domain and the data mining process's purpose. The different ways to handle missing data are:

- 1) Ignore the data row
- 2) Fill the missing values manually
- 3) Use a global constant to fill in for missing values
- 4) Use attribute mean or median
- 5) Use forward fill or backward fill method
- 6) Use a data-mining algorithm to predict the most probable value

Handling Noise & Outliers -

Noise in data can be created as a result of data gathering errors, data entry problems, or data transmission errors, among other things. Different sorts of noise and outliers include unknown encoding (Example: Marital Status — Q), out of range numbers (Example: Age — -10), inconsistent data (Example: DoB — 4th Oct 1999, Age — 50), inconsistent formats (Example: DoJ — 13th Jan 2000, DoL — 10/10/2016), and so on.

Binning can be used to deal with noise. Sorted data is arranged into bins or buckets using this method. Equal-width (distance) or equal-depth (frequency) partitioning can be used to produce bins. Smoothing can be done on these bins. Bin mean, bin median, and bin borders are all options for smoothing.

Binning and then smoothing can be used to smooth outliers. Visual analysis or box plots can be used to detect them. Clustering can be used to find outlier data groups. The detected outliers may be smoothed or removed.

Normalization -

Normalization or Min-Max Scaling is used to transform features to be on a similar scale. The new point is calculated as:

$$X_{\text{new}} = (X - X_{\text{min}}) / (X_{\text{max}} - X_{\text{min}})$$

This scales the range to [0, 1] or sometimes [-1, 1]. Geometrically speaking, transformation squishes the n-dimensional data into an n-dimensional unit hypercube. Normalization is useful when there are no outliers as it cannot cope up with them. Usually, we would scale age and not incomes because only a few people have high incomes but the age is close to uniform.

Standardization -

Standardization or Z-Score Normalization is the transformation of features by subtracting from mean and dividing by standard deviation. This is often called a Z-score.

$$X_{\text{new}} = (X - \text{mean}) / \text{Std}$$

Standardization can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Geometrically speaking, it translates the data to the mean vector of original data to the origin and squishes or expands the points if std is 1 respectively. We can see that we are just changing mean and standard deviation to a standard normal distribution which is still normal thus the shape of the distribution is not affected.

Standardization does not get affected by outliers because there is no predefined range of transformed features.

6) Dimensionality reduction using Principal components (PCA)

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Attribute Evaluation

Choose PrincipalComponents -R 0.95 -A 5

Search Method

Choose Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

☒ Use full training set
☐ Cross-validation Folds 10 Seed 1

(Nom) Star type

Start Stop

Result list (right-click for options)

16.26.59 - Ranker + CorrelationAttrib
16.28.01 - Ranker + InfoGainAttribute
16.29.43 - Ranker + PrincipalCompo

Attribute selection output

```
0.5255 6 -0.422Spectral Class=A+0.339Spectral Class=B+0.311Radius(R/Ro)=-0.284Star color=White=0.259Spectral Class=O...
0.4886 7 0.528Star color=Blue White=0.468Star color=Blue-white+0.438Star color=Orange+0.311Star color=White=0.182Star color=yellow...
0.4539 8 0.517Star color=yellow-white=0.446Star color=Yellowish White=0.388Star color=white+0.366Star color=Whitish+0.23 Star colo...
0.4197 9 0.808Spectral Class=0-0.429Star color=Orange=0.166Star color=Blue white=0.154Star color=yellow-white+0.139Star color=Whit...
0.3858 10 -0.624Star color=Blue white=0.539Star color=White=0.313Spectral Class=0-0.262Star color=white=0.226Star color=Whitish...
0.352 11 0.667Star color=white=0.383Star color=Yellowish White=0.334Star color=Blue white =0.268Star color=Blue White=0.227Star co...
0.3182 12 -0.648Star color=Whitish+0.442Star color=Blue white=0.407Star color=Yellowish White=0.315Star color=Blue-White+0.24 Star...
0.2845 13 -0.455Star color=Blue white -0.422Star color=Blue -0.389Star color=white=0.365Star color=Yellowish White=0.326Star color...
0.251 14 -0.542Star color=White-Yellow=0.535Star color=Pale yellow orange=0.436Star color=Blue +0.305Star color=Yellowish White=0...
0.2174 15 0.681Star color=yellowish=0.486Star color=Yellowish=0.486Star color=Orange-Red=0.159Star color=White-Yellow=0.159Star col...
0.1839 16 -0.733Star color=Blue-White+0.373Star color=Whitish=0.287Star color=Blue white -0.279Star color=Blue -0.224Star color=Whit...
0.1504 17 0.708Star color=Blue white -0.629Star color=Blue -0.171Star color=Pale yellow orange=0.149Star color=White-Yellow=0.148Star...
0.1169 18 -0.704Star color=Yellowish+0.704Star color=Orange-Red=0.065Star color=Pale yellow orange=0.064Star color=White-Yellow=0.0...
0.0835 19 -0.704Star color=White-Yellow=0.704Star color=Pale yellow orange=0.064Star color=Yellowish=0.064Star color=Orange-Red=0.0...
0.0576 20 -0.596Star color=Orange=0.395Radius(R/Ro)=0.392Spectral Class=0+0.258Star color=Blue White=0.239Star color=Blue white...
0.0372 21 0.422Star color=Blue White=0.405Spectral Class=B-0.388Star color=White=0.378Spectral Class=A-0.255Star color=Blue-White...
```

Selected attributes: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21 : 21

Status

OK Log x 0

7) Classification using Decision trees

The screenshot shows the Weka Explorer window with the REPTree classifier selected. The 'Test options' section on the left indicates 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' section on the right displays the following results:

Root relative squared error: 34.5457 %
 Total Number of Instances: 240

==== Detailed Accuracy By Class ====

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PNC Area	Class
1.000	0.035	0.031	1.000	0.920	0.906	1.000	1.000	1.000	A
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	B
0.900	0.010	0.947	0.900	0.923	0.909	0.996	0.977	0.977	F
0.825	0.015	0.917	0.825	0.868	0.845	0.976	0.906	0.906	C
0.975	0.020	0.907	0.975	0.980	0.928	0.985	0.968	0.968	D
0.875	0.005	0.932	0.875	0.921	0.908	0.991	0.944	0.944	E
Weighted Avg.	0.929	0.014	0.932	0.929	0.929	0.916	0.991	0.966	

==== Confusion Matrix ====

	a	b	c	d	e	f	C-- classified as
40	0	0	0	0	0	0	a = A
0	40	0	0	0	0	0	b = B
4	0	34	0	0	0	0	c = F
3	0	0	33	3	1	1	d = C
0	0	0	1	39	0	0	e = D
0	0	2	2	1	35	1	f = E

The 'Result list' on the left shows '11:58:15 - trees.REPTree'.

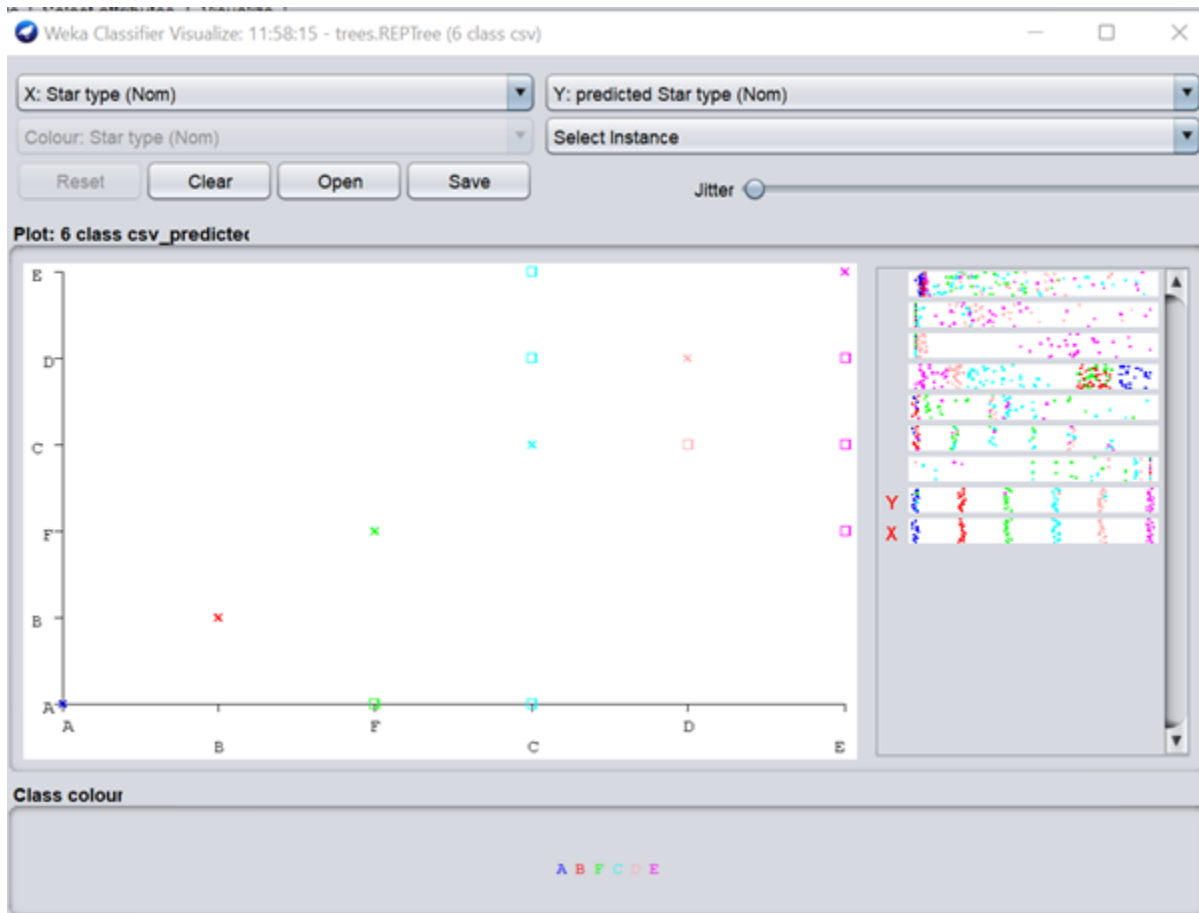
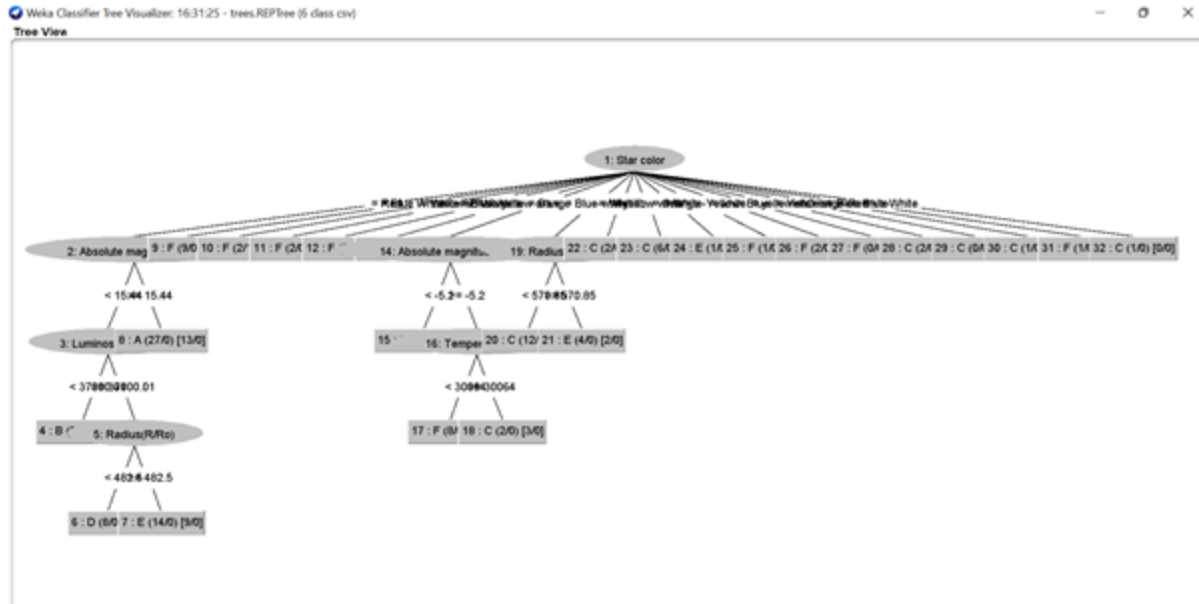
Size of the tree : 32

Time taken to build model: 0.01 seconds

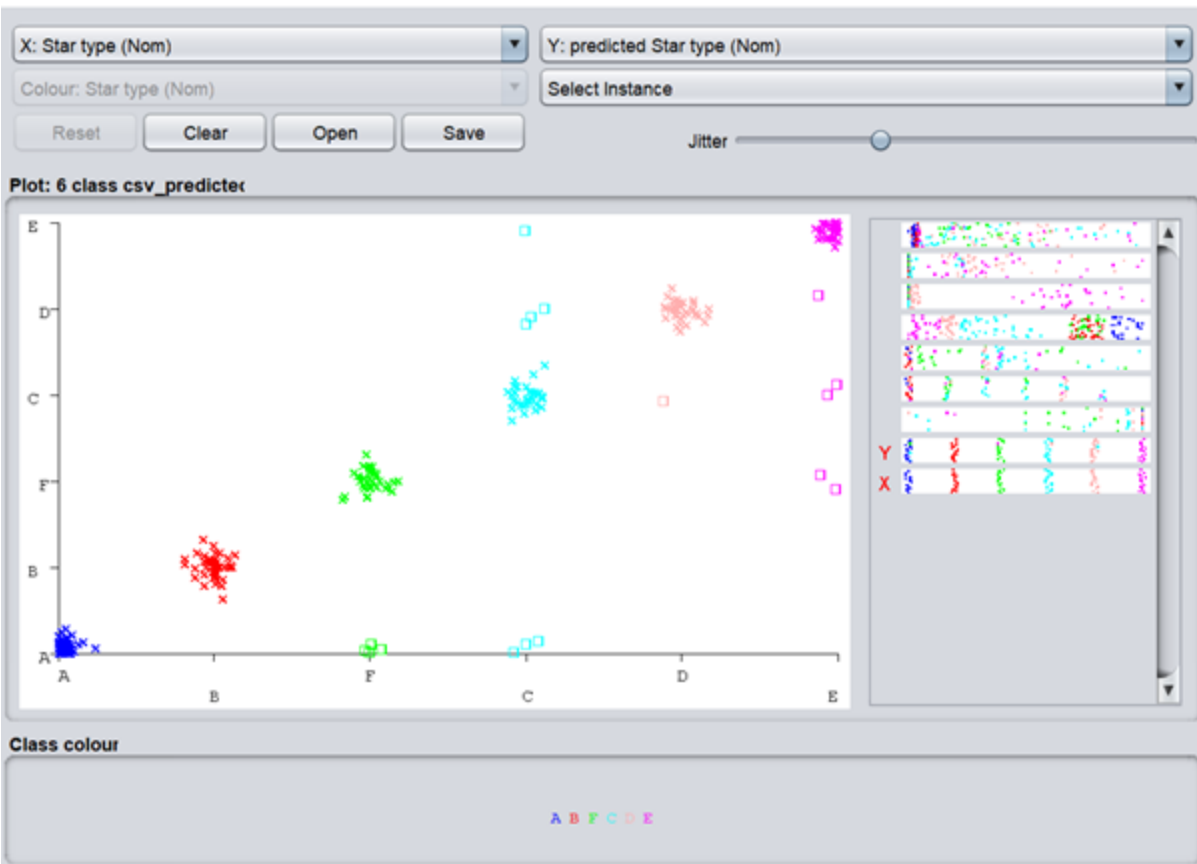
==== Stratified cross-validation ====

==== Summary ====

Correctly Classified Instances	223	92.9167 %
Incorrectly Classified Instances	17	7.0833 %
Kappa statistic	0.915	
Mean absolute error	0.0271	
Root mean squared error	0.1287	
Relative absolute error	9.7411 %	
Root relative squared error	34.5457 %	
Total Number of Instances	240	

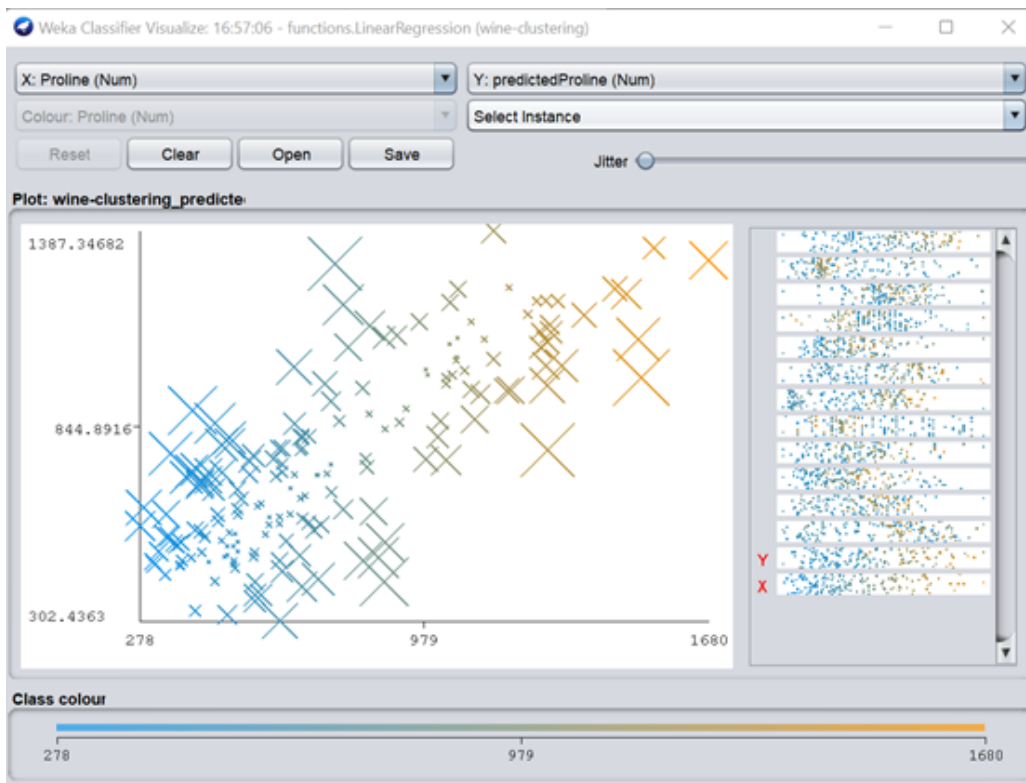
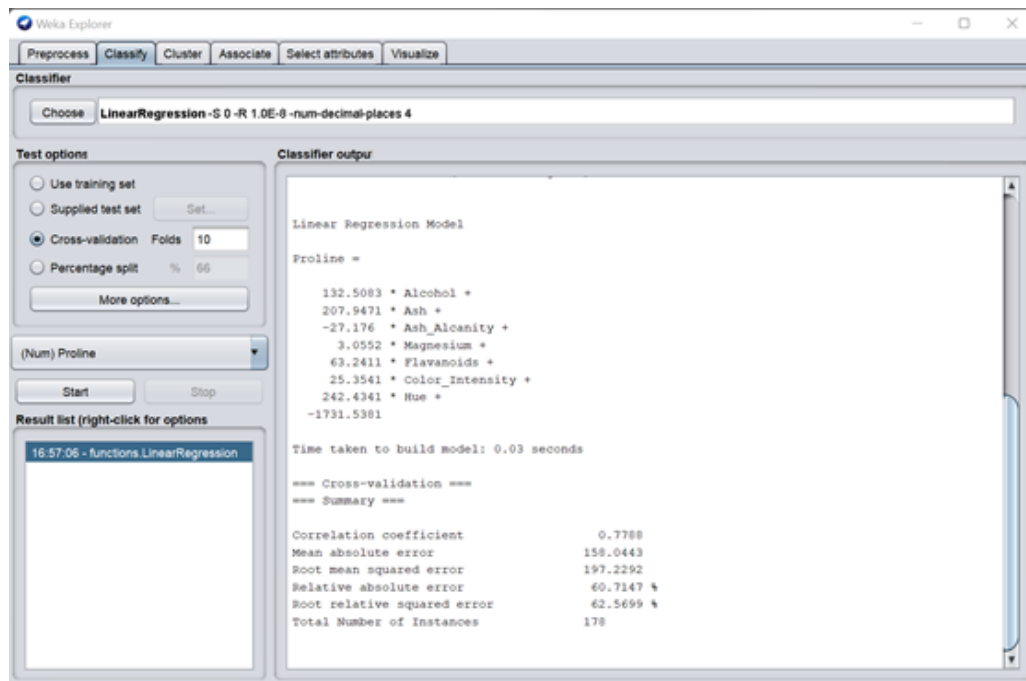


Using some Jitter



8) Regression Tasks

Implement linear Regression



Implement logistic regression

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...

Variable	Class	A	B	F	C	D
Temperature (K)		-0.0028	0.0019	-0.0009	0.0032	0.0006
Luminosity (L/L _o)		0.0001	-0.0002	-0	-0.0001	-0
Radius (R/R _o)		0.1025	-0.1182	-0.0179	-0.0538	-0.1071
Absolute magnitude (M _v)		18.175	-2.6781	1.7887	-0.2151	-3.5152
Star color=Red		-49.108	15.1471	-19.9149	7.7397	1.7505
Star color=Blue White		-50.8312	4.6768	43.0445	2.4603	26.0063
Star color=White		-90.3027	22.1454	25.5743	33.1626	4.101
Star color=Yellowish White		-97.4297	58.121	30.3236	-5.9311	39.3589
Star color=Blue white		-77.1746	19.056	15.2599	45.1564	22.2079
Star color=Pale yellow orange		-32.3932	9.5621	2.0310	2.9255	42.2033
Star color=Blue		38.0603	3.3557	29.6014	-38.8306	15.7708
Star color=Blue-white		74.3603	-61.8432	-18.2862	24.1565	-54.1701
Star color=Whitish		107.8768	-35.8509	-46.5252	44.1919	-44.6546
Star color=Yellow-white		99.9836	-17.4586	-37.7106	14.5591	-34.1429
Star color=Orange		32.6449	114.1015	12.4706	36.6664	71.0603
Star color=White-Yellow		-33.1173	10.1595	1.6009	3.5162	42.1698
Star color=White		-45.1716	20.6891	21.3986	-6.718	34.3057
Star color=Blue		35.4133	-3.6265	23.1439	-32.7696	2.412
Star color=Yellowish		41.7433	-7.8328	-52.226	-14.3923	58.5373
Star color=Yellowish		73.6844	-46.5547	25.9194	-14.6628	-12.537
Star color=Orange-Red		60.7389	-17.1273	-5.7423	4.2106	-16.2879
Star color=Blue white		-51.6584	8.3071	13.1004	42.1464	12.8028
Star color=Blue-White		161.2902	-63.9489	-23.3488	28.7284	-53.5763
Spectral Class=M		-52.7066	13.1837	-20.5939	0.0058	8.9949
Spectral Class=B		-15.0796	0.9229	20.2896	-30.5629	16.0907
Spectral Class=A		18.3257	15.1348	7.562	-17.1997	13.4759
Spectral Class=F		-22.6404	-2.322	33.4145	9.6999	7.9858
Spectral Class=O		97.4909	-39.437	-4.9862	23.6415	-38.7934
Spectral Class=K		49.4133	27.3546	-9.9775	76.9718	-36.9831
Spectral Class=G		204.4089	105.9766	38.8751	45.6425	12.4383
Intercept		-158.6944	19.9491	11.1024	-13.5823	33.3988

Odds Ratios...

Variable	Class	A	B	F	C	D
Temperature (K)		0.9972	1.0019	0.9991	1.0032	1.0006
Luminosity (L/L _o)		1.0001	0.9998	1	0.9999	1
Radius (R/R _o)		1.1079	0.8885	0.9823	0.9477	0.8984
Absolute magnitude (M _v)		78214314.869	0.0687	5.9818	0.8065	0.0297
Star color=Red		0	3786892.4543	0	2297.7035	5.7573
Star color=Blue White		0	107.4268	4.9431022087017892818	11.7089	1.9696071613771902811
Star color=White		0	4146155172.5596	1.2787412894255319811	2.5254171831256975814	60.4021
Star color=Yellowish White		0	1.74441242205408825	1.4769319397642275813	0.0029	1.239794705972536817
Star color=Blue white		0	188768460.1109	4239257.7703	4.08483232212777819	6413331396.4746
Star color=Pale yellow orange		0	14216.2942	7.4282	18.6428	2.13134759772614835818
Star color=Blue		3.3837358189599456816	28.467	7.173729618032664812	0	7066125.1417
Star color=Blue-white		1.9691211371105947832	0	0	3.0977825480072365810	0
Star color=Whitish		7.084293821732493846	0	0	1.5571024462786872819	0
Star color=Yellow-white		2.6443512922561983843	0	0	2103376.7282	0
Star color=Orange		1.5049278680187647814	3.578174428348214849	260570.3537	8.395270004822912815	7.26239890348172830
Star color=White-Yellow		0	25835.0213	4.9577	33.4579	2.0411377933550154818
Star color=White		0	96433832.7463	1964766591.9054	0.0012	7.92129603430797814
Star color=Blue		2.397772664438689815	0.0266	1.1252500145968239810	0	11.1559
Star color=Yellowish		1.3454610729638313818	0.0004	0	0	2.64492570650806825
Star color=Yellowish		1.0017005877141715832	0	1.8058038515579355811	0	0
Star color=Orange-Red		2.3909030714286743826	0	0	0.0032	67.3941
Star color=Blue white		0	4052.5748	489145.4356	2.01344444273628237818	363247.076
Star color=Blue-White		1.1154772243142857870	0	0	2.9961979571408804812	0
Spectral Class=M		0	531644.7882	0	1.0058	8061.7006
Spectral Class=B		0	2.5166	648135832.9227	0	9729722.1494
Spectral Class=A		90936060.187	3740583.728	1923.4786	0	712082.5254
Spectral Class=F		0	0.0981	3.248941960639533814	16315.3958	2938.9515
Spectral Class=O		2.1865854272401227842	0	0.0068	1.8509032531859875810	0
Spectral Class=K		2.883654558255802821	7.586823824815914811	0	2.681868461359114833	0
Spectral Class=G		5.938218652112033888	1.0594127670079418846	7.6428382858989812816	6.641547982137212819	308123.8245

Time taken to build model: 0.17 seconds

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

Correctly Classified Instances	229	95.4167 %
Incorrectly Classified Instances	11	4.5833 %
Kappa statistic	0.945	
Mean absolute error	0.016	
Root mean squared error	0.1233	
Relative absolute error	5.7643 %	
Root relative squared error	33.076 %	
Total Number of Instances	240	

==== Detailed Accuracy By Class ====

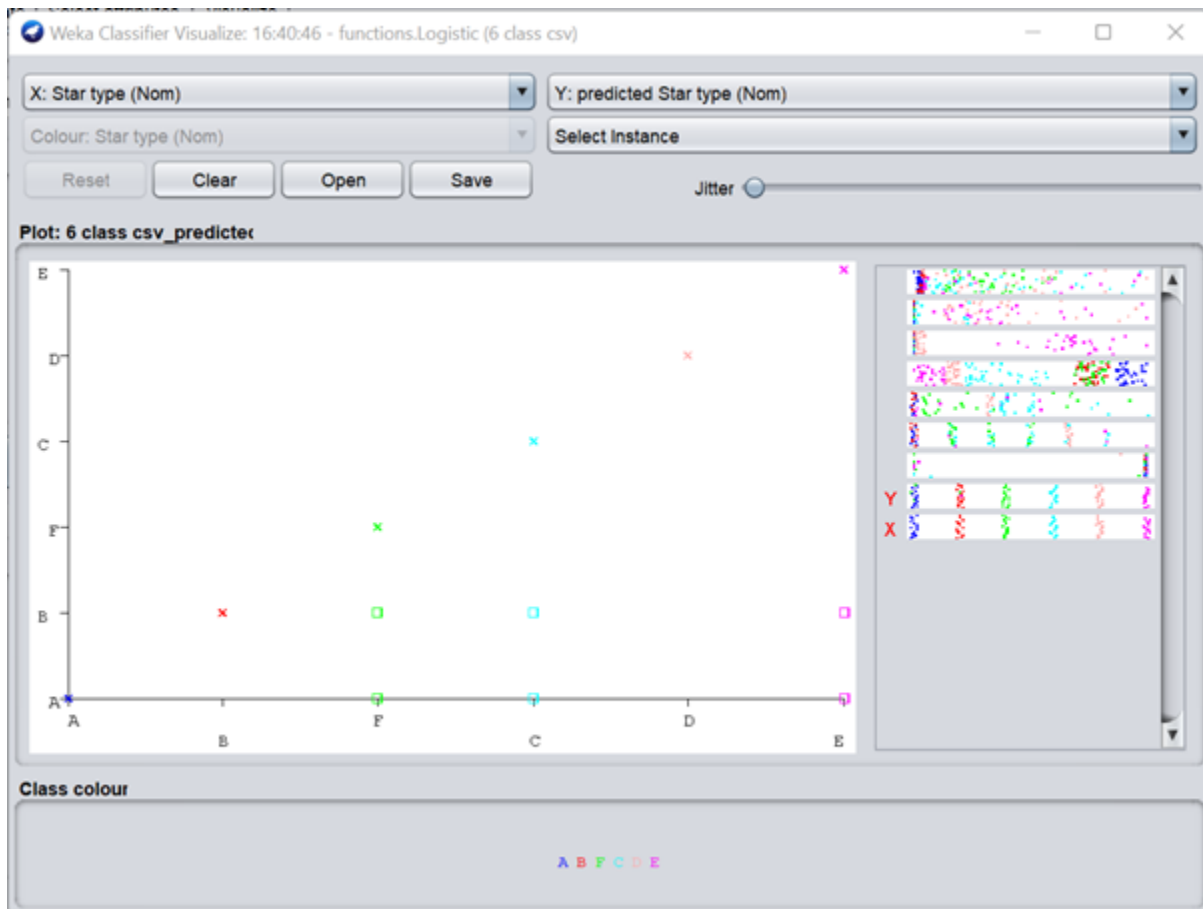
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.035	0.851	1.000	0.920	0.906	0.995	0.976	A
	1.000	0.020	0.909	1.000	0.952	0.944	0.996	0.967	B
	0.875	0.000	1.000	0.875	0.933	0.924	0.993	0.979	F
	0.925	0.000	1.000	0.925	0.961	0.955	0.986	0.971	C
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	D
	0.925	0.000	1.000	0.925	0.961	0.955	0.978	0.959	E
Weighted Avg.	0.954	0.009	0.960	0.954	0.955	0.947	0.991	0.975	

==== Confusion Matrix ====

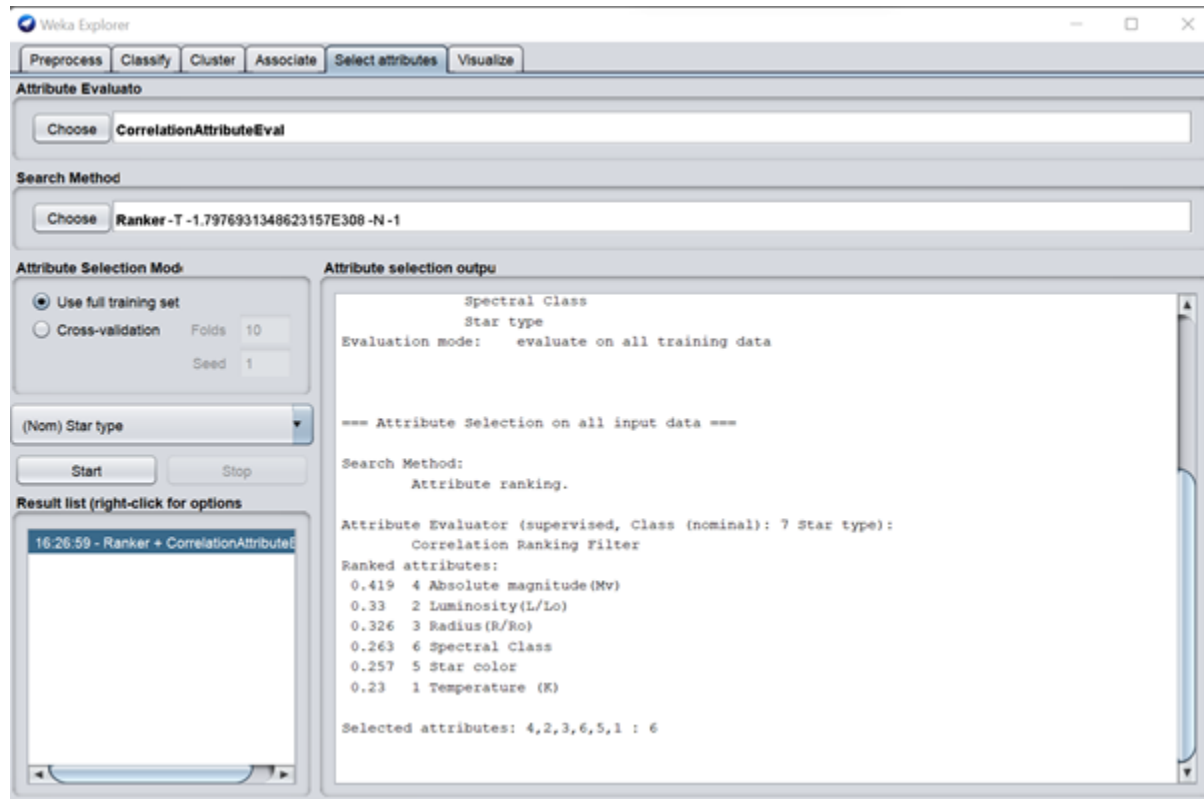
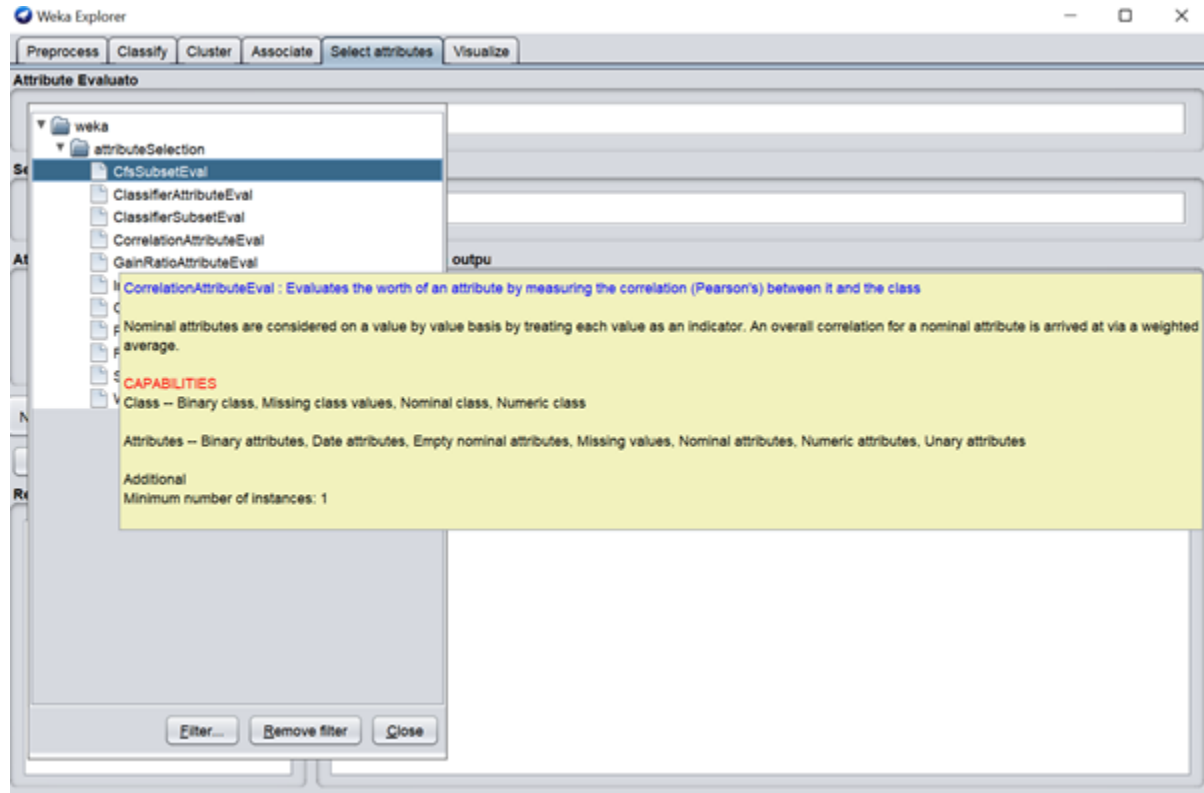
```

a b c d e f <-- classified as
40 0 0 0 0 0 | a = A
0 40 0 0 0 0 | b = B
4 1 35 0 0 0 | c = F
1 2 0 37 0 0 | d = C
0 0 0 0 40 0 | e = D
2 1 0 0 0 37 | f = E

```



Calculate correlation matrix for numerical attributes.



9) Association rule mining

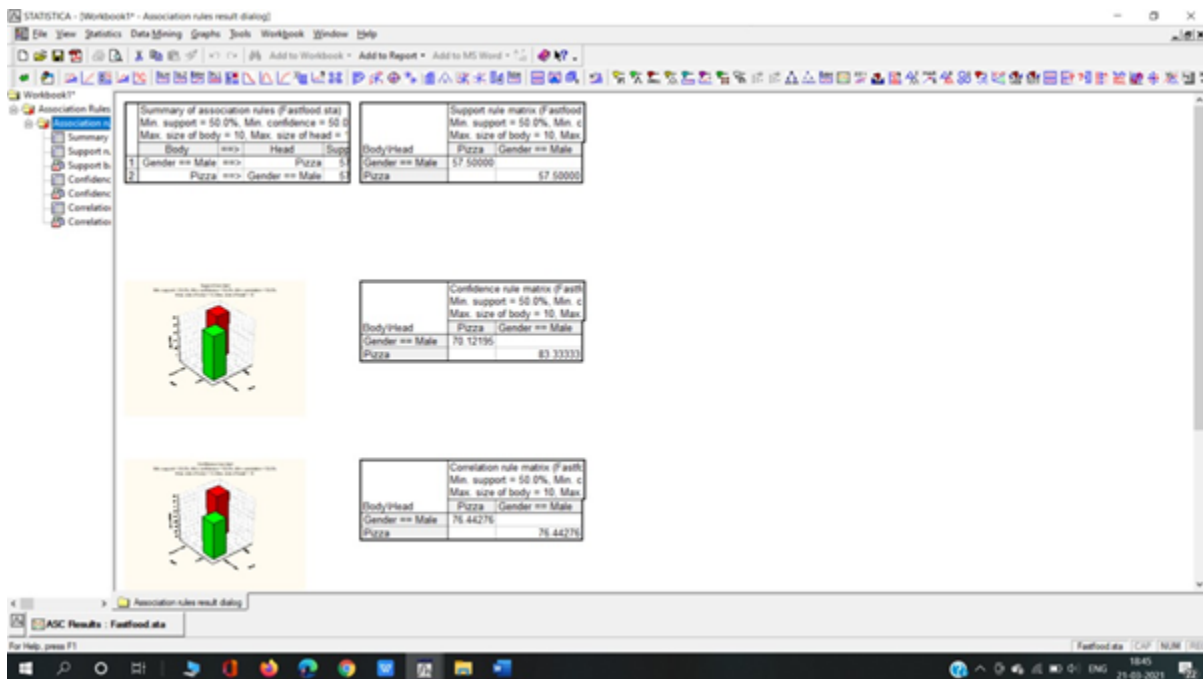
Association rule learning has become a popular approach for uncovering hidden associations between variables in huge data sets. Association rules can be used to express the discovered relationships. Association rules are rules that show the connection or correlation between two or more objects. A correlation matrix is a table that displays the coefficients of correlation between variables. The correlation between two variables is shown in each cell of the table. A correlation matrix can be used to summarise data, as an input to a more sophisticated study, or as a diagnostic tool for advanced analyses.

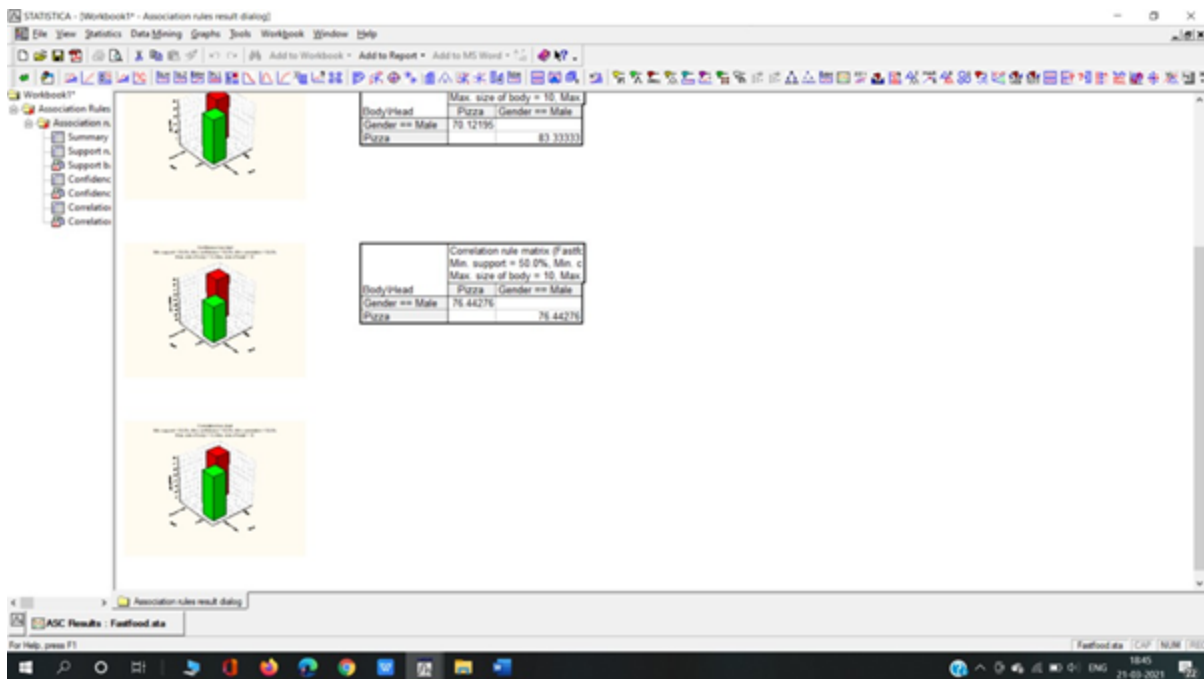
- 1) Summary of Association Rule
- 2) Support Rule Matrix
- 3) Support Bar Chart
- 4) Correlation Rule Matrix
- 5) Correlation Bar Chart

Done in three ways

- 1) On statistical tool
- 2) On WEKA with WEKA explorer window
- 3) On WEKA with knowledge window

Implementing on Statistical tool (OUTPUT):





Summary of Association Rules

STATISTICA - [Workbook1* - Summary of association rules (Fastfood.sta)]

File Edit View Insert Format Statistics Data Mining Graphs Tools Data Workbook Window Help

Summary of association rules (Fastfood.sta)
Min. support = 50.0%, Min. confidence = 50.0%, Min. correlation = 50.0%
Max. size of body = 10, Max. size of head = 10

	Body	==>	Head	Support(%)	Confidence(%)	Correlation(%)
1	Gender == Male	==>	Pizza	57.50000	70.12195	76.44276
2	Pizza	==>	Gender == Male	57.50000	83.33333	76.44276

Support Rule Matrix

STATISTICA - [Workbook1* - Support rule matrix (Fastfood.sta)]

File Edit View Insert Format Statistics Data Mining Graphs Tools Data Workbook Window Help

Support rule matrix (Fastfood.sta)
Min. support = 50.0%, Min. confidence = 50.0%, Min. correlation = 50.0%
Max. size of body = 10, Max. size of head = 10

Body/Head	Pizza	Gender == Male
Gender == Male	57.50000	
Pizza		57.50000

Confidence Rule Matrix

STATISTICA - [Workbook1* - Confidence rule matrix (Fastfood.sta)]

File Edit View Insert Format Statistics Data Mining Graphs Tools Data Workbook Window Help

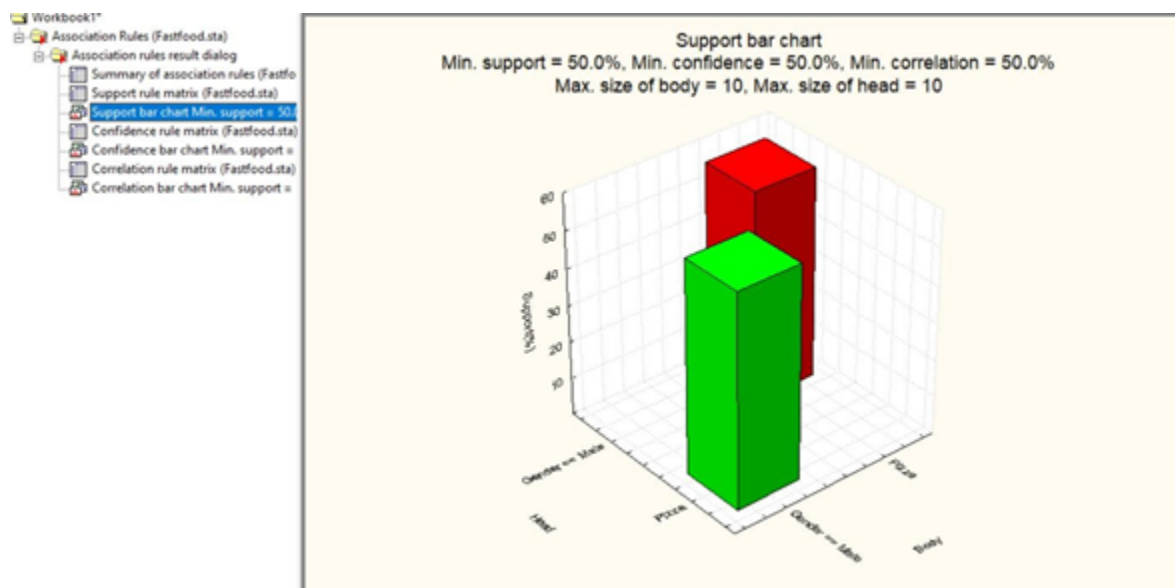
Arial 10 B I U

Workbook1*

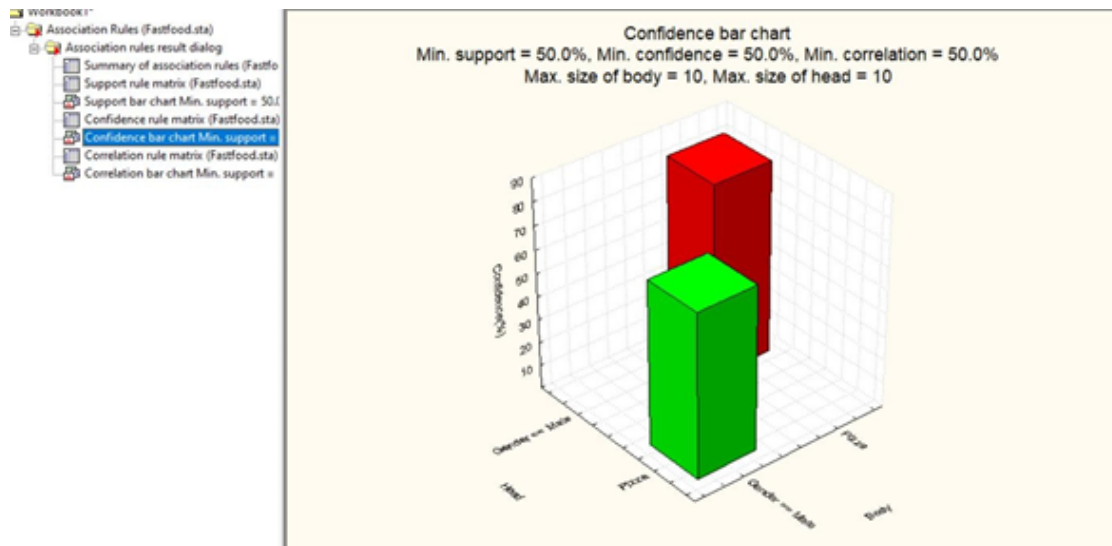
- Association Rules (Fastfood.sta)
 - Association rules result dialog
 - Summary of association rules (Fastfo
 - Support rule matrix (Fastfood.sta)
 - Support bar chart Min. support = 50.0
 - Confidence rule matrix (Fastfood.sta)**
 - Confidence bar chart Min. support =
 - Correlation rule matrix (Fastfood.sta)

Min. support = 50.0%, Min. confidence = 50.0%, Min. correlation = 50.0%		Max. size of body = 10, Max. size of head = 10	
Body\Head	Pizza	Gender == Male	
Gender == Male	70.12195		
Pizza		83.33333	

Support Bar Chart



Confidence Bar Chart



Correlation Matrix

STATISTICA - [Workbook1* - Correlation rule matrix (Fastfood.sta)]

File Edit View Insert Format Statistics Data Mining Graphs Tools Data Workbook Window Help

Arial 10 B I U

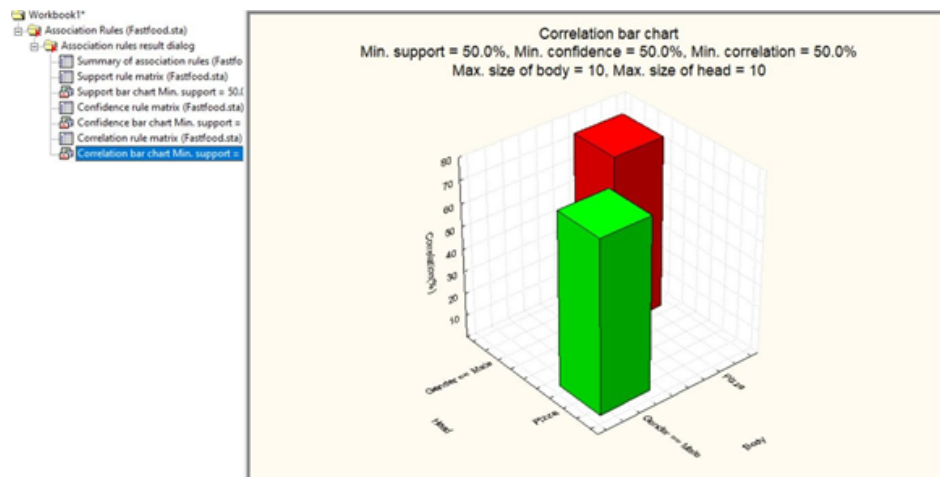
Workbook1*

- Association Rules (Fastfood.sta)
 - Association rules result dialog
 - Summary of association rules (Fastfo
 - Support rule matrix (Fastfood.sta)
 - Support bar chart Min. support = 50.0
 - Confidence rule matrix (Fastfood.sta)
 - Confidence bar chart Min. support =
 - Correlation rule matrix (Fastfood.sta)**
 - Correlation bar chart Min. support =

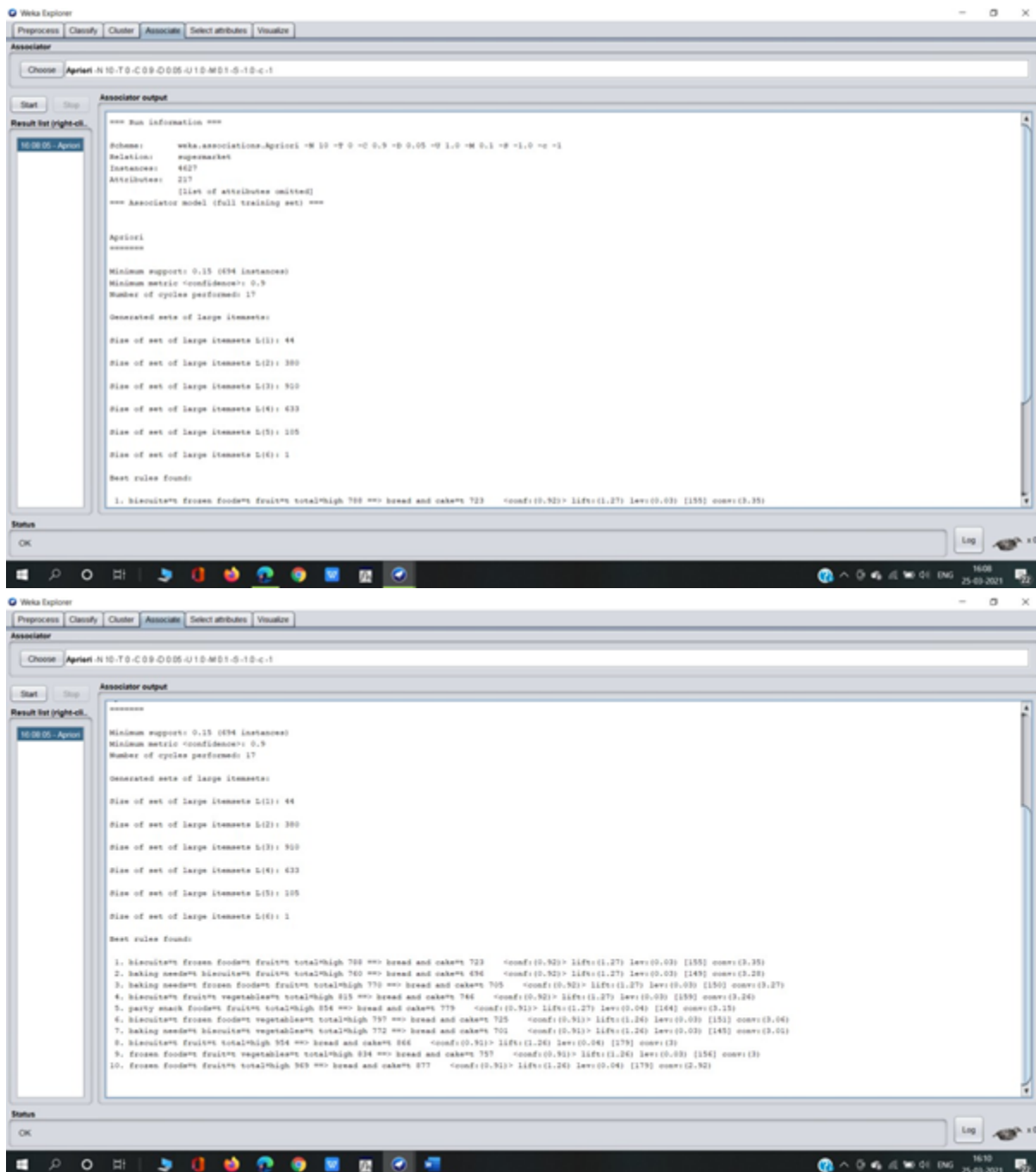
Correlation rule matrix (Fastfood.sta)
 Min. support = 50.0%, Min. confidence = 50.0%, Min. correlation = 50.0%
 Max. size of body = 10, Max. size of head = 10

Body/Head	Pizza	Gender == Male
Gender == Male	76.44276	
Pizza		76.44276

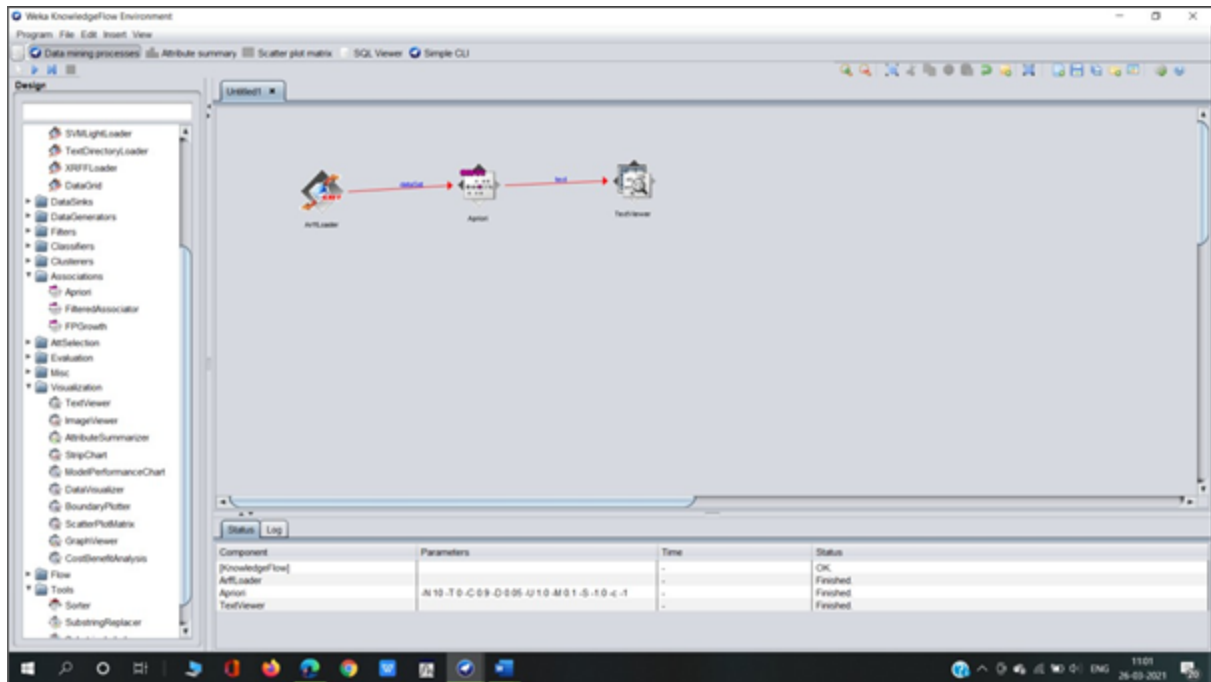
Correlation Barchart



Implementation on WEKA(Output)



On Knowledge Flow working with apriori after selecting a data set of supermarkets. Using text viewer for result finding full information after using algorithm.

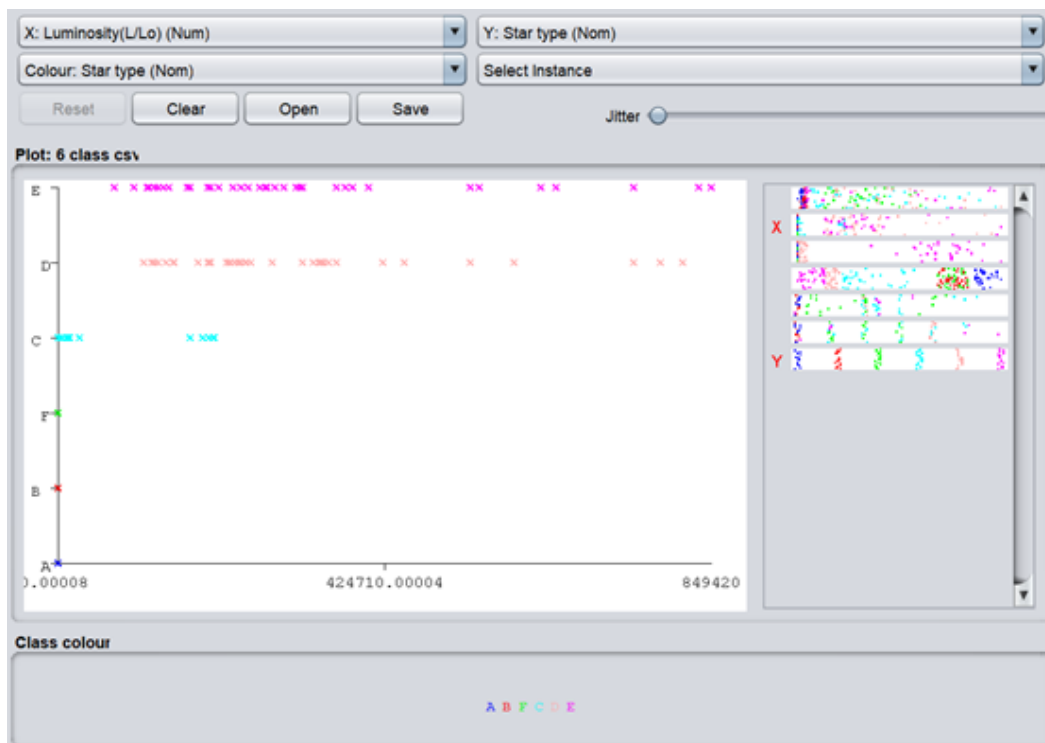
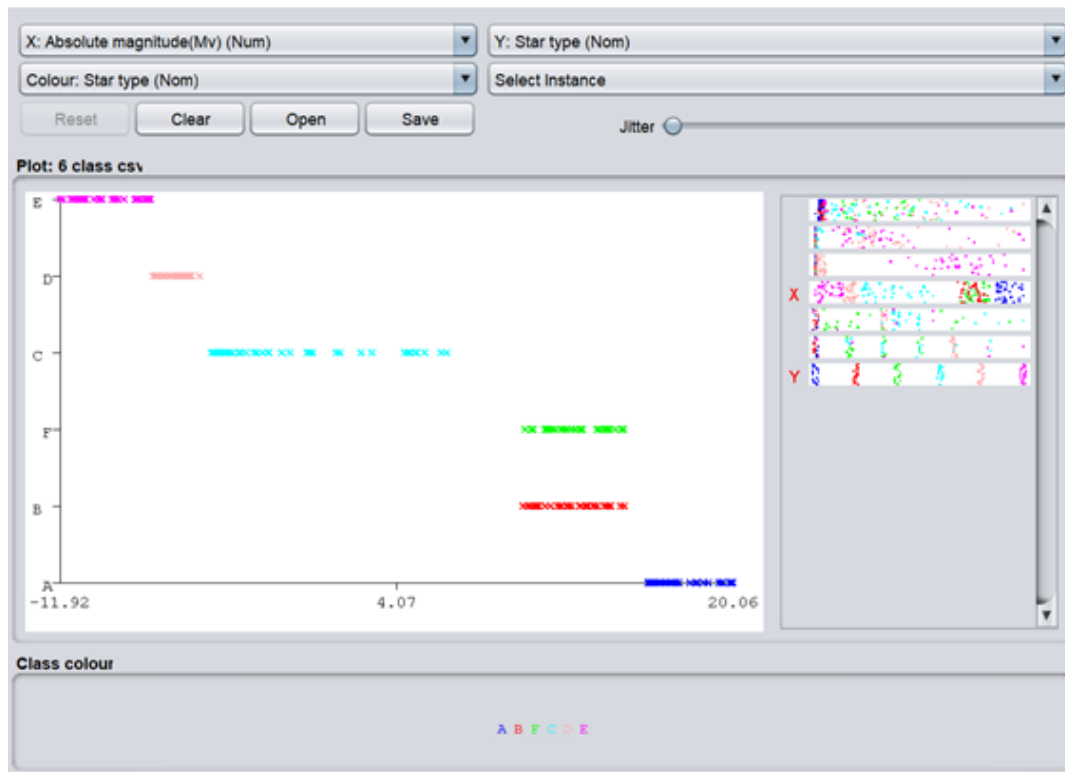


TEXTVIEWER RESULT

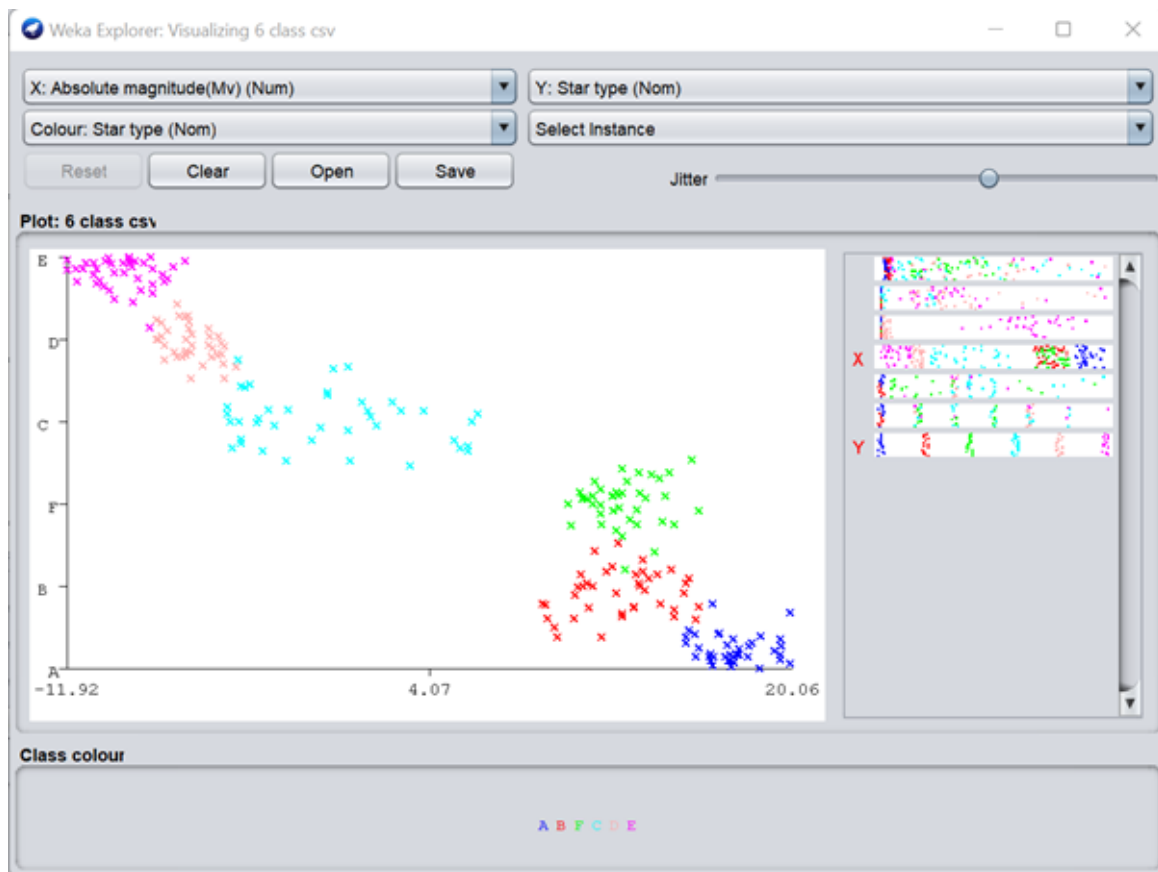
The screenshot shows the 'Text Viewer' window with a 'Result list' on the left and a 'Text' area on the right. The 'Result list' shows '10:15:00.404 - Model: Apriori'. The 'Text' area contains the following output:

```
*** Association model ***  
Scheme: Apriori  
Relation: supermarket  
  
Apriori  
=====  
Minimum support: 0.15 (636 instances)  
Minimum metric <confidence>: 0.9  
Number of cycles performed: 17  
  
Generated sets of large itemsets:  
Size of set of large itemsets L(1): 48  
Size of set of large itemsets L(2): 300  
Size of set of large itemsets L(3): 910  
Size of set of large itemsets L(4): 433  
Size of set of large itemsets L(5): 105  
Size of set of large itemsets L(6): 1  
  
Best rules found:  
1. biscuits% frozen food% fruit% total%high 708 ==> bread and cake% 723 <conf:(0.92)> lift:(1.27) lev:(0.03) [155] covr:(3.35)  
2. baking needs% biscuits% fruit% total%high 760 ==> bread and cake% 696 <conf:(0.92)> lift:(1.27) lev:(0.03) [149] covr:(3.28)  
3. baking needs% frozen food% fruit% total%high 770 ==> bread and cake% 708 <conf:(0.92)> lift:(1.27) lev:(0.03) [150] covr:(3.27)  
4. biscuits% fruit% vegetable% total%high 815 ==> bread and cake% 746 <conf:(0.92)> lift:(1.27) lev:(0.03) [155] covr:(3.24)  
5. party snack food% fruit% total%high 854 ==> bread and cake% 779 <conf:(0.92)> lift:(1.27) lev:(0.04) [144] covr:(3.15)  
6. biscuits% frozen food% vegetable% total%high 797 ==> bread and cake% 725 <conf:(0.91)> lift:(1.24) lev:(0.03) [151] covr:(3.04)  
7. baking needs% biscuits% vegetable% total%high 772 ==> bread and cake% 701 <conf:(0.92)> lift:(1.24) lev:(0.03) [145] covr:(3.01)  
8. biscuits% fruit% total%high 958 ==> bread and cake% 846 <conf:(0.91)> lift:(1.24) lev:(0.04) [179] covr:(3)  
9. frozen food% fruit% vegetable% total%high 834 ==> bread and cake% 757 <conf:(0.91)> lift:(1.24) lev:(0.03) [154] covr:(3)  
10. frozen food% fruit% total%high 949 ==> bread and cake% 877 <conf:(0.92)> lift:(1.24) lev:(0.04) [179] covr:(2.92)
```

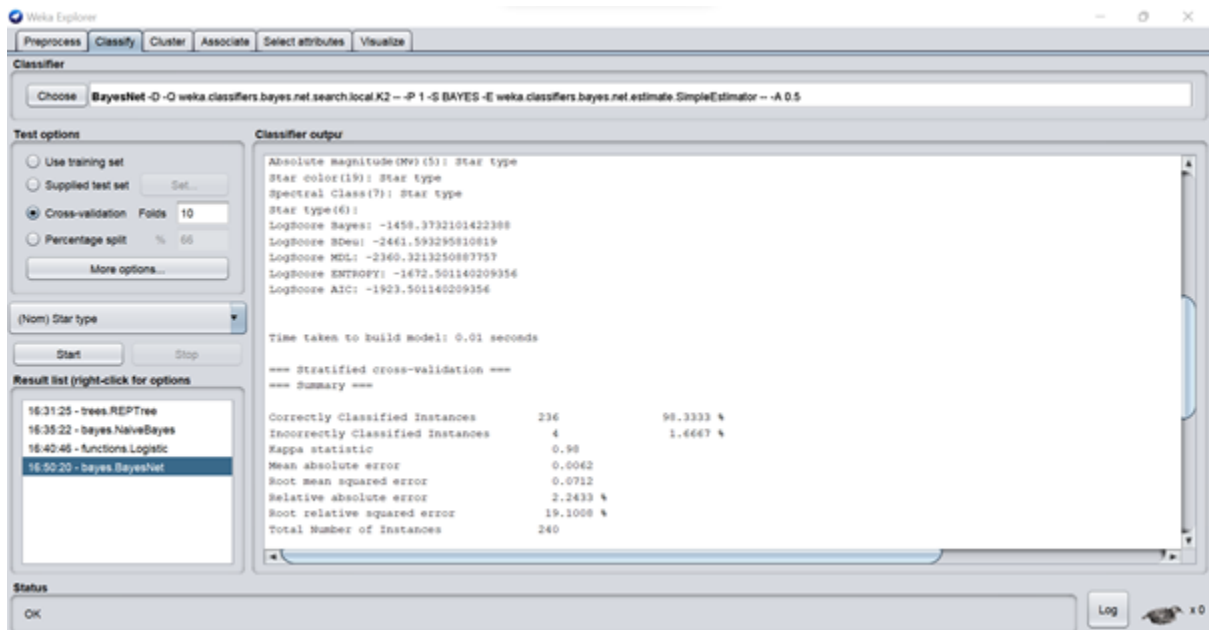
10) Visualization Techniques



Using Jitter - Jitter is used to introduce random noise to the dataset so as to classify data points (separate them)



11) Use Bayesian Learning for classification



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Test options

- ☐ Use training set
- ☐ Supplied test set
- ☒ Cross-validation Folds 10
- ☐ Percentage split % 66

More options...

(Nom) Star type

Start Stop

Result list (right-click for options)

- 16:31:25 - trees.REPTree
- 16:35:22 - bayes.NaiveBayes
- 16:40:46 - functions.Logistic
- 16:50:20 - bayes.BayesNet

Classifier output

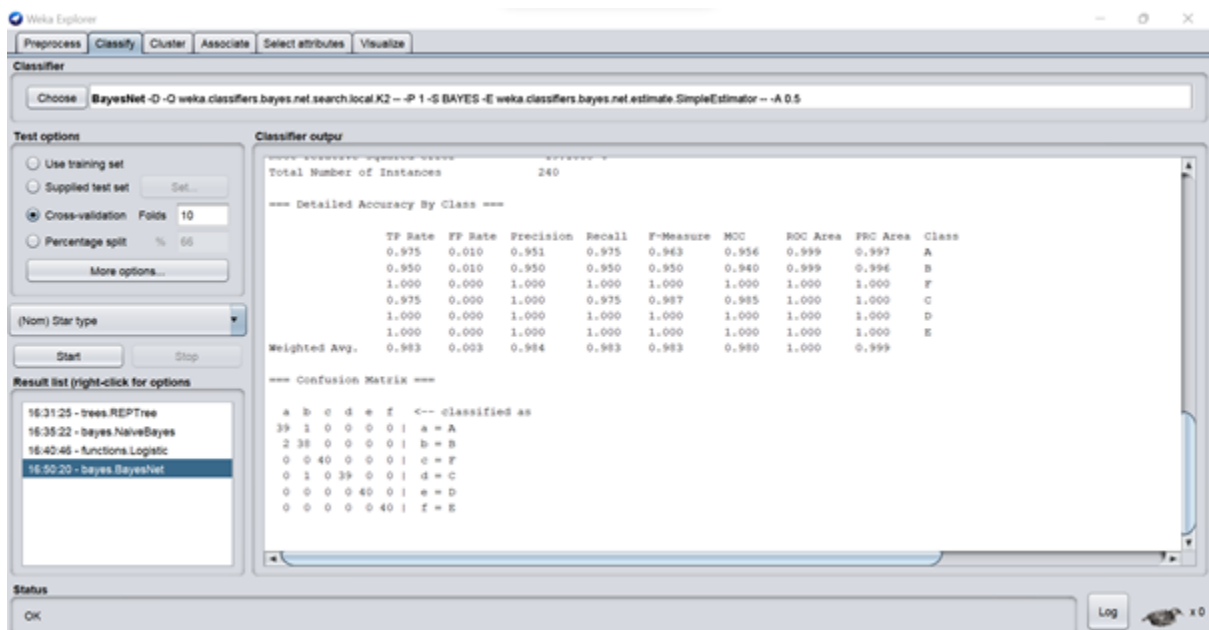
```
Absolute magnitude(MV): Star type
Star color(19): Star type
Spectral Class(7): Star type
Star type(6):
LogScore Bayes: -1458.3732101422388
LogScore BDeu: -2461.593295810819
LogScore MDL: -2340.3213250887757
LogScore ENTROPY: -1672.501140209356
LogScore AIC: -1923.501140209356

Time taken to build model: 0.01 seconds

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

Correctly Classified Instances      236      98.3333 %
Incorrectly Classified Instances    4        1.6667 %
Kappa statistic                    0.98
Mean absolute error                0.0042
Root mean squared error            0.0712
Relative absolute error            2.2433 %
Root relative squared error        19.1008 %
Total Number of Instances         240
```

Status: OK Log x0



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Test options

- ☐ Use training set
- ☐ Supplied test set
- ☒ Cross-validation Folds 10
- ☐ Percentage split % 66

More options...

(Nom) Star type

Start Stop

Result list (right-click for options)

- 16:31:25 - trees.REPTree
- 16:35:22 - bayes.NaiveBayes
- 16:40:46 - functions.Logistic
- 16:50:20 - bayes.BayesNet

Classifier output

```
Total Number of Instances      240

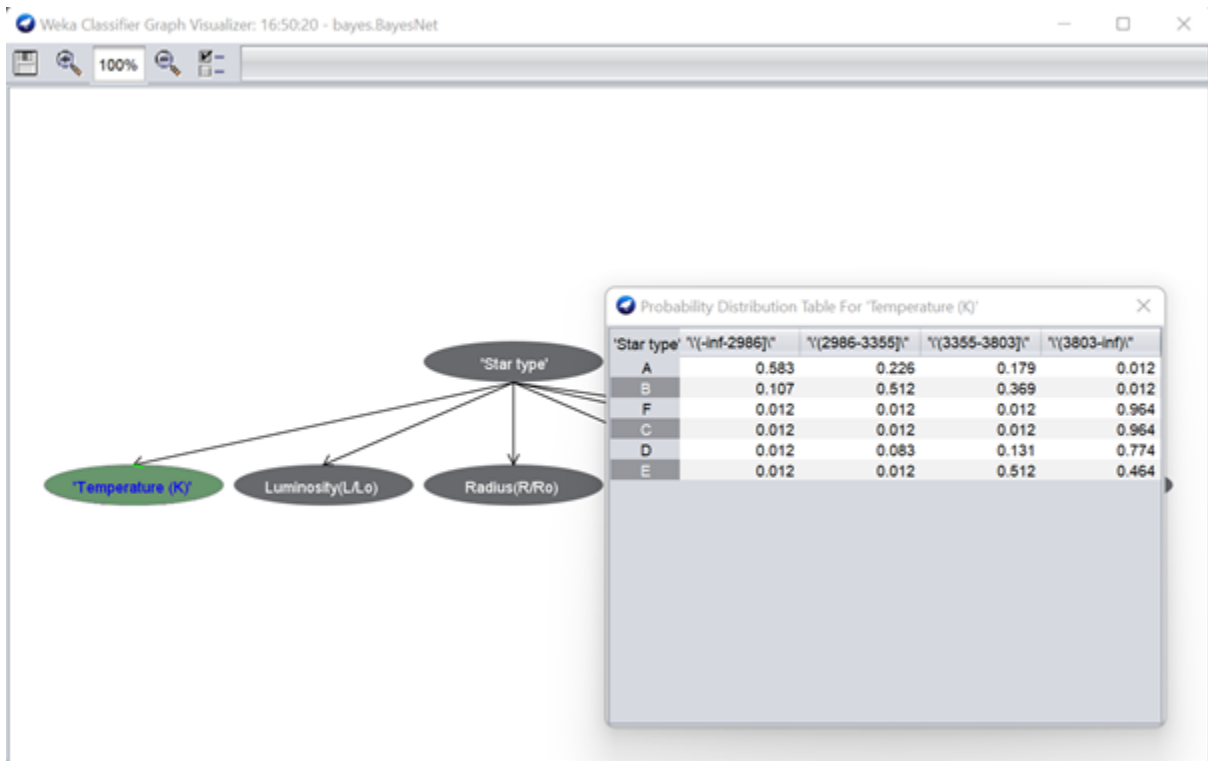
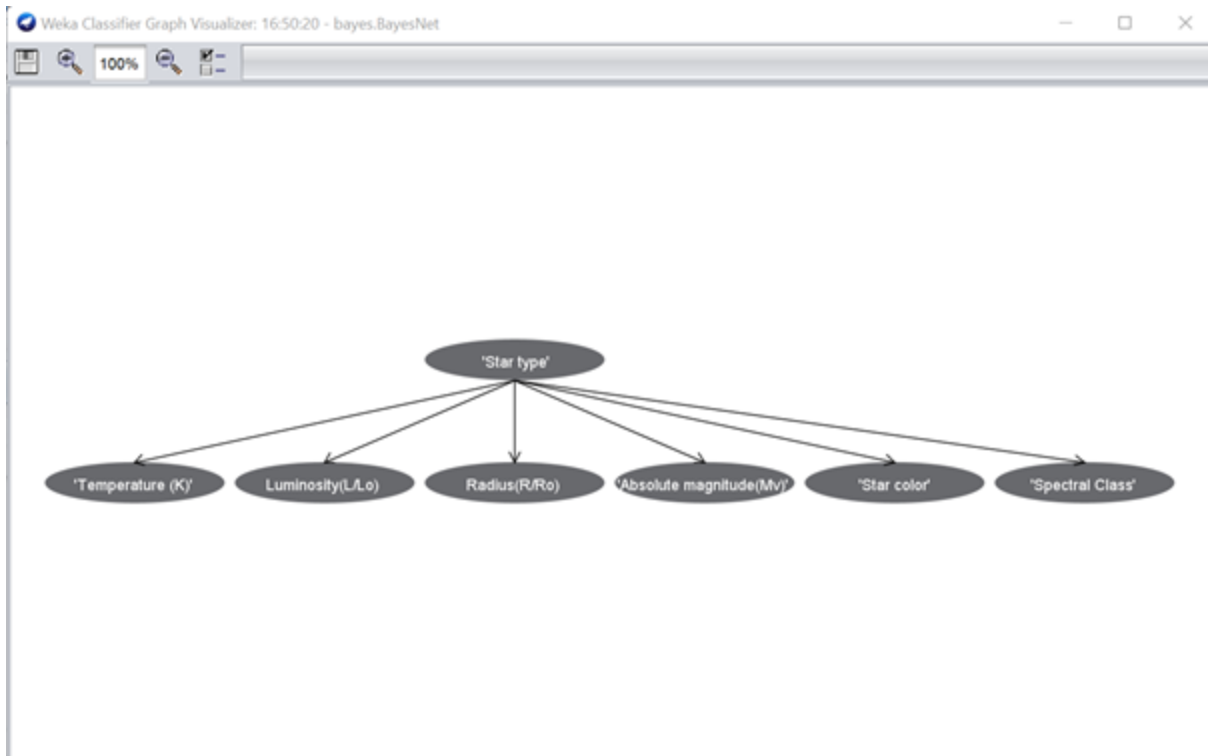
=== Detailed Accuracy By Class ===

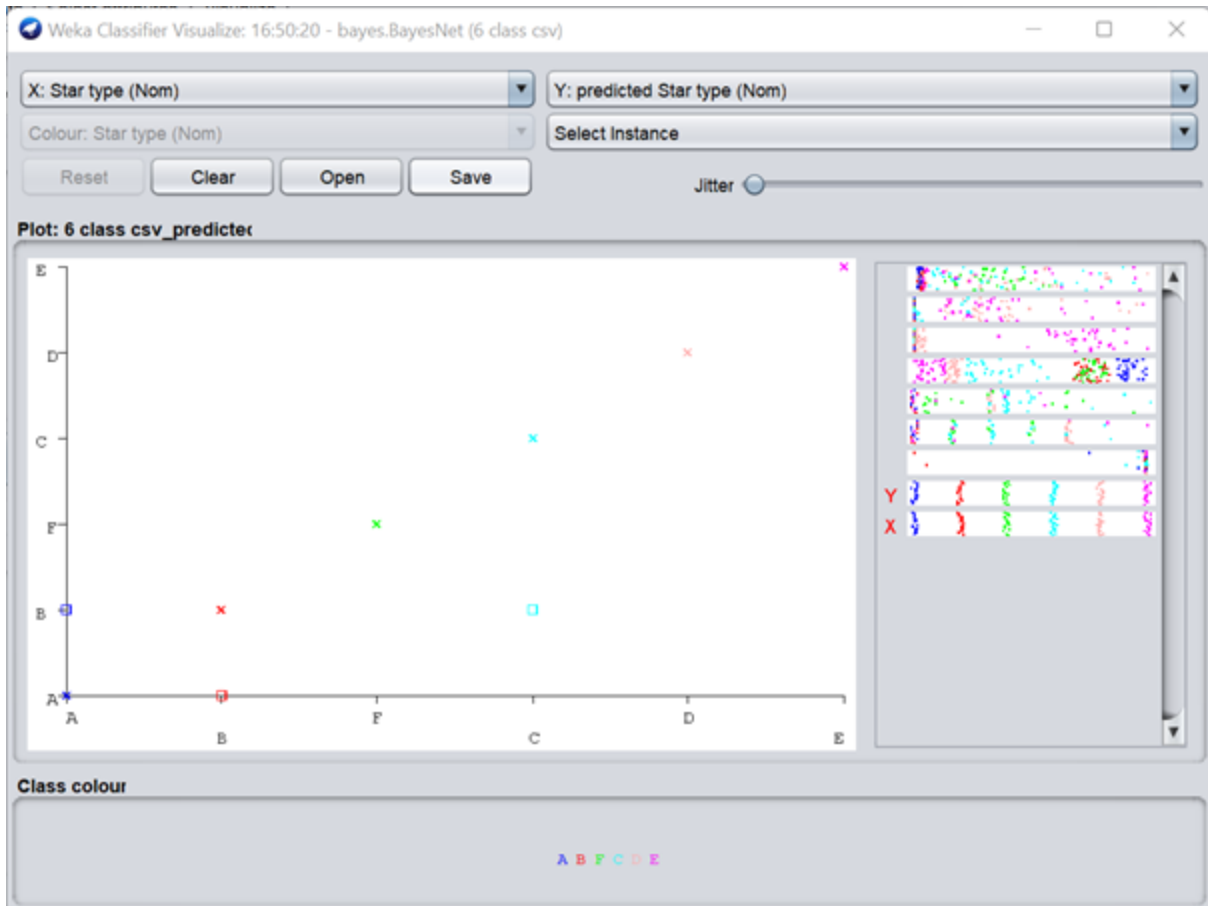
      TP Rate  FP Rate  Precision  Recall   F-Measure  MDC     ROC Area  PRC Area  Class
      0.975   0.010   0.951    0.975   0.963    0.956   0.999    0.997    A
      0.950   0.010   0.950    0.950   0.950    0.940   0.999    0.996    B
      1.000   0.000   1.000    1.000   1.000    1.000   1.000    1.000    F
      0.975   0.000   1.000    0.975   0.987    0.985   1.000    1.000    C
      1.000   0.000   1.000    1.000   1.000    1.000   1.000    1.000    D
      1.000   0.000   1.000    1.000   1.000    1.000   1.000    1.000    E
Weighted Avg.  0.983   0.003   0.984    0.983   0.983    0.980   1.000    0.999

=== Confusion Matrix ===

 a b c d e f <-- classified as
39 1 0 0 0 0 | a = A
 2 38 0 0 0 0 | b = B
 0 0 40 0 0 0 | c = F
 0 1 0 39 0 0 | d = C
 0 0 0 0 40 0 | e = D
 0 0 0 0 0 40 | f = E
```

Status: OK Log x0





12) Implement clustering algorithm

K-means Clustering is a popular exploratory data analysis tool for gaining an understanding of the data's structure. It is the task of identifying subgroups in data so that data points within the same subgroup (cluster) are extremely similar while data points within different clusters are very dissimilar. To put it another way, we strive to discover homogeneous subgroups within the data so that data points in each cluster are as comparable as feasible based on a similarity measure like euclidean-based distance or correlation-based distance. The choice of the similarity measure to utilise depends on the application.

K Means algorithm is an iterative procedure that attempts to split a dataset into K unique non-overlapping subgroups (clusters), each of which contains only one data point. It attempts to make intra-cluster data points as comparable as possible while maintaining clusters as distinct (far) as possible. It distributes data points to clusters in such a way that the sum of the squared distances between them and the cluster's centroid (arithmetic mean of all the data points in that cluster) is as small as possible. Within clusters, the less variance there is, the more homogenous (similar) the data points are.

The screenshot displays the Weka Clusterer window. The 'Clusterer' dropdown is set to 'SimpleKMeans'. The 'Cluster mode' section has 'Use training set' selected. The 'Cluster output' pane shows the following text:

```
=== Clustering model (full training set) ===
KMeans
-----
Number of iterations: 18
Within cluster sum of squared errors: 64.53766702389433

Initial starting points (random):
Cluster 0: 13.3,1.72,2.14,17.94,2.4,2.19,0.27,1.35,3.95,1.02,2.77,1285
Cluster 1: 12.22,1.29,1.94,19.92,2.34,2.04,0.39,2.08,2.7,0.84,3.02,312

Missing values globally replaced with mean/mode

Final cluster centroids:
Attribute      Full Data      Cluster#
              (178.0)      (108.0)      (70.0)
-----
Alcohol         13.0006      13.0809      12.8767
Malic_Acid       2.3363       1.9249       2.9711
Ash              0.3448       0.3443       0.4006
```

The 'Result list' on the left shows '16.58.26 - SimpleKMeans'.

Final cluster centroids:

Attribute	Full Data (178.0)	Cluster#	
		0 (108.0)	1 (70.0)
Alcohol	13.0006	13.0809	12.8767
Malic_Acid	2.3363	1.9249	2.9711
Ash	2.3665	2.3443	2.4009
Ash_Alcanity	19.4949	18.5296	20.9843
Magnesium	99.7416	100.9352	97.9
Total_Phenols	2.2951	2.6738	1.7109
Flavanoids	2.0293	2.6898	1.0101
Nonflavanoid_Phenols	0.3619	0.3008	0.456
Proanthocyanins	1.5909	1.8576	1.1794
Color_Intensity	5.0581	4.4097	6.0584
Hue	0.9574	1.0671	0.7882
OD280	2.6117	3.0891	1.8751
Proline	746.8933	844.6852	596.0143

Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0 108 (61%)
1 70 (39%)

