Data Mining - COP259

Coursework assignment

Part 1– Pre-Processing

This part mainly comprises Missing values removal/handling, outliers' removal, Normalisation/ Standardisation.
This part is crucial for overall data mining and Analysis process as discrepancy in this can result in errors in rest of the process.

Handling Missing Values and Duplicate Removal: The Data set Indian Liver Patient Dataset (ILPD) consists of 416 liver patient records and 167 non liver patient records. There are 11 attributes in total Out of which 9 are numeric and 2 attributes class and gender are Nominal.

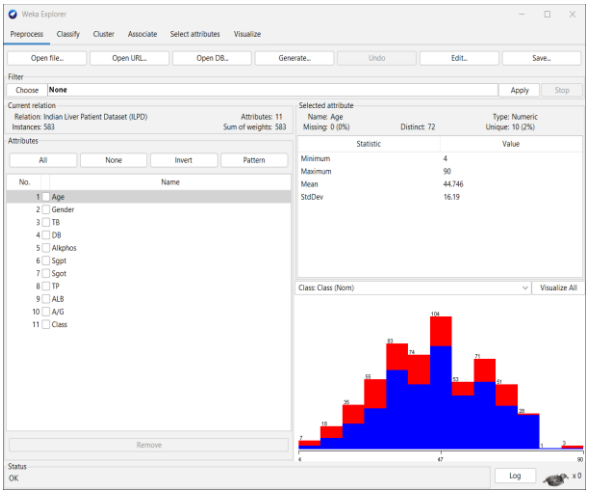

figure 1

The initial step I took is to analyse the missing values from the edit bar.
As there were missing values in A/G, DB, TB, TP , Sgot , Alkphos. Therefore, I applied filter Remove With Values "weka.filters.unsupervised.instance.RemoveWithValues -S 0.0 -C 3 -L first-last -M". We can change values by clicking the properties. Which removes all missing values.
Furthermore, I deleted all the duplicate values to get all unique and distinct values so that it will not create any confusion during the Data mining process. For this purpose, I applied the filter: " weka.filters.unsupervised.instance.RemoveDuplicates".

Outliers Removal:
Next step is to analyse the outliers. It can be done by applying filter to remove the outliers, for this we will apply filter: "weka.filters.unsupervised.attribute.Interquartile Range -R first-last -O 3.0 -E 6.0" This forms two new attributes named "Outlier" and "Extreme" which can be used to eliminate instances that weka considers to fit to these two categories.
Next step is to remove all the yes labels from outliers and extreme values. This can be done by applying filter: Values "weka.filters.unsupervised.instance.RemoveWithValues -S 0.0 -C 3 -L first-last -M"

Class Balance:
As the class is not balanced. Therefore, I have applied : "weka.filters.supervised.instance.SMOTE -C 0 -K 5 -P 100.0 -S 1" to balance both the classes in order to deal with the minority lass and remove bias in the data.

After this we will remove outliers and extreme value field inorder to avoid confusion in the Data Minning.

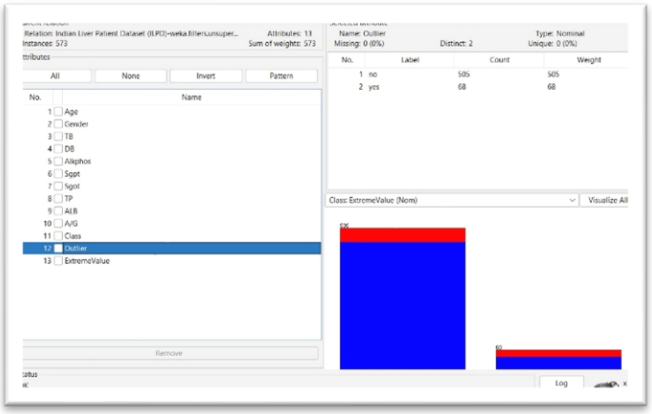

figure 2

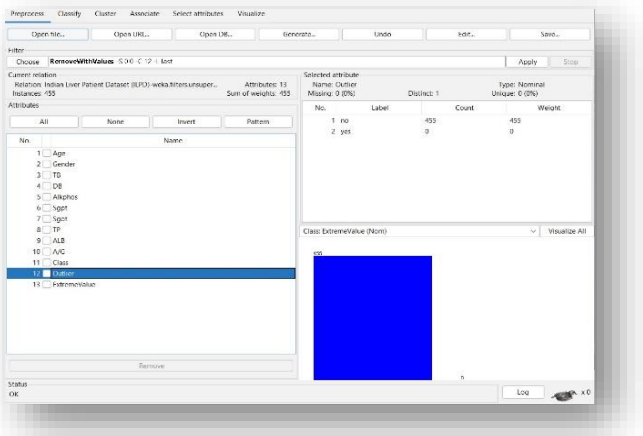

figure 3

Normalization:
After this we will Normalise the numeric attributes in order to improve performance, stability, and interpretability of the model. Normalization can be done y applying the filter : "weka.filters.unsupervised.attribute.Normalize -S 1.0 -T 0.0". Normalization rescales the Numeric attributes in the data to a common range, typically between 0 and 1 or between -1 and 1.

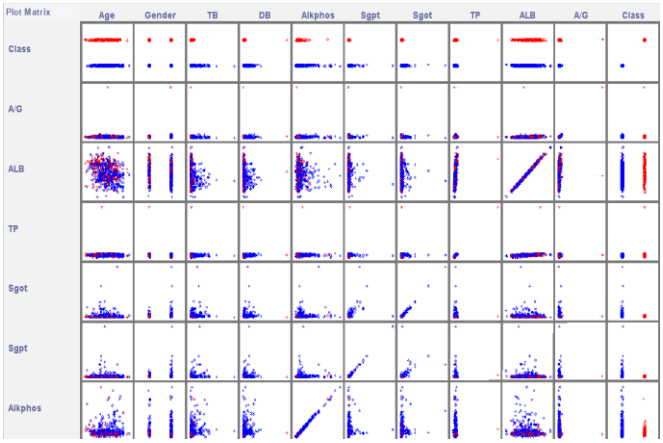

figure 4

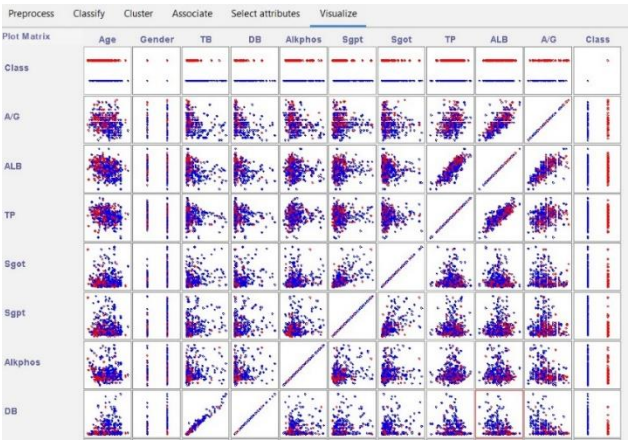

figure 5

As we can analyse that **Fig 4** seems less corelated and data seems to be overfit and noisy. However, in **Fig 5** the data seems to be correlated up to an extent and less noisy and overfitted.
It can be observed n the plots that few attributes such as Age, ALB, A/G , TP does not fit well in the data and seems to make the Data Mining process noisy. In order to get a robust model, it is necessary to get rid of all the attributes that are creating nuisance in the data. Therefore, we will test the attributes in next part to verify if these attributes are contributing towards the data or are they unnecessary.

Part 2 – Ranking

It is a method to Rank the attributes based on their individual evaluation scores.We do Ranking to reduce the size / dimension of the data for facilitating the data mining and analysis. As, the larger the data more chances of redundancy. However, it is crucial to reduce the dimension in such manner that will prevent loss of important information. In this part I have used 3 methods for Ranking.

PCA:

Principal Component Analysis is a method of Ranking that aims to reduce the size of dimension while preserving the information. I have used this method by applying attribute selection from the pane and choose “weka.attributeSelection.PrincipalComponents -R 0.95 -A 5” from attribute evaluator and “weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1” as the Ranker method. The PCA suggests strong correlation between TB, DB towards the target variable. As seen in **figure 6** equations Also, the high correlation suggests that a significant portion of the data contributes towards the target variable. PC1 describes the greatest variance (29.3%), resulted by PC2 (18.09%), and so on. Alkphos, Gender=Male: Impact substantially to PC3, implying some significance but smaller than the preliminary attributes.

```
Ranked attributes:
 0.7045    1 0.437TB+0.436DB-0.371ALB-0.334A/G+0.307Sgot...
 0.5235    2 -0.494ALB-0.481TP-0.395Sgpt-0.297A/G-0.292Sgot...
 0.4176    3 -0.438Sgot-0.434Sgpt+0.43 DB+0.414TB+0.303TP...
 0.3222    4 0.571Alkphos-0.506Gender=Male-0.417A/G-0.411Age+0.156TP...
 0.2336    5 0.757Class=No_liver_disease-0.492Age+0.262TB+0.243DB-0.146TP...
 0.155     6 -0.746Gender=Male-0.478Alkphos+0.329Sgot+0.245Sgpt-0.136TP...
 0.089     7 0.668Age+0.596Class=No_liver_disease+0.254Alkphos+0.218TP+0.142Sgpt...
 0.0363    8 -0.643A/G-0.503Alkphos+0.48 TP+0.205Sgot+0.197Gender=Male...

Selected attributes: 1,2,3,4,5,6,7,8 : 8
```

figure 6

InfoGainAttributeEval:

I have next applied this method for ranking the attributes for classification. This was performed on the original data. Class is nominal here.

Similar to the PCA this method also indicates TB, DB as the most significant attributes contributing towards the most information gain. However, Alkphos, Sgot, Sgpt to be of moderate importance. A/G, ALB, TP, Age are least contributing towards predicting Liver Disease, No Liver Disease (class). However, it can be observed that Gender can have a negative impact on the data if not handled carefully. Nevertheless, it might be worth to keep this attribute with prior domain knowledge.

InfoGain also provides almost the same ranking as he PCA.

It can be observed from the **figure 7** below that DB, TB has the highest corelation. However, A/G, ALB, TP and Age can be neglected as they have zero values which indicates they have no contribution towards predicting the class.

```
Ranked attributes:
 0.0577     4 DB
 0.05215    3 TB
 0.03988    5 Alkphos
 0.02963    7 Sgot
 0.02867    6 Sgpt
 0.00272    2 Gender
 0         10 A/G
 0          9 ALB
 0          8 TP
 0          1 Age

Selected attributes: 4,3,5,7,6,2,10,9,8,1 : 10
```

figure 7

ReliefAttributeEval:
This method also uses Nominal class. It is interesting to know that this also signifies TB, DB as the promising attribute leading towards major prediction of the target variable. Whereas Alkphos, Sgot, Sgpt to be of moderate importance as mentioned by InfoGain method. Furthermore, gender needs to handle cautiously as it can have a negative trend in the model as observed in the **figure 8** below.
It can be observed below that Relief Attribute is similar to the above 2. However, Age is on 6th Rank here and in the Info Gain it is last.

```
Ranked attributes:
 0.02257    3 TB
 0.02148    4 DB
 0.01041    5 Alkphos
 0.0087     7 Sgot
 0.00595    6 Sgpt
 0.00578    1 Age
 0.00492   10 A/G
 0.00238    8 TP
 0.00195    9 ALB
-0.00265    2 Gender

Selected attributes: 3,4,5,7,6,1,10,8,9,2 : 10
```

figure 8

From the above Ranking methods it can be inferred that the attributes which are contributing towards predicting the class are: TB, DB, Alkphos, Sgot, Sgpt, Gender and class is our target variable/attribute. Although Gender is negatively classified in the Relief Attribute selection, but PCA and Info Gain signifies it as a low contributor, and it can be taken with the prior domain knowledge. Therefore, I am including this attribute.

Part 3 – Classification

In this part I will compare the models on the original dataset and the modified dataset as per part 2. The modified dataset consists of attributes out of which one is the target attribute.

a. Original Dataset:

Random Forest:
First, I will apply Random Forest classifier on original dataset which is a type of a decision tree.
The model identifies 73.65% correct instances which seems to be a moderate accuracy. Kappa statistic is 0.4693 which indicates fair agreement between predictions and true labels. MAE and RMSE indicates that the model is not affected by larger errors.
The model identifies 193 instances of liver disease accurately. However, 100 times the model identify liver diseases incorrectly. Contributing towards the low precision and kappa statistic for the Liver_Disesase class. Which can be observed in the figure 9 below.

```
Correctly Classified Instances         450               73.6498 %
Incorrectly Classified Instances       161               26.3502 %
Kappa statistic                          0.4693
Mean absolute error                      0.3668
Root mean squared error                  0.4174
Relative absolute error                 73.4894 %
Root relative squared error             83.5506 %
Total Number of Instances              611

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.659    0.192    0.760      0.659    0.706      0.473    0.816     0.821     Liver_disease
                 0.808    0.341    0.720      0.808    0.761      0.473    0.816     0.815     No_liver_disease
Weighted Avg.    0.736    0.270    0.739      0.736    0.735      0.473    0.816     0.818

=== Confusion Matrix ===

   a   b   <-- classified as
 193 100 |   a = Liver_disease
  61 257 |   b = No_liver_disease
```

figure 9

Logistic Regression:

This model shows lower accuracy 64.8118%. However, it has similar trends like the previous model. Misclassifies more Liver_disease cases, which could be critical.

```
Correctly Classified Instances         396               64.8118 %
Incorrectly Classified Instances       215               35.1882 %
Kappa statistic                          0.2883
Mean absolute error                      0.4258
Root mean squared error                  0.4661
Relative absolute error                 85.2986 %
Root relative squared error             93.3018 %
Total Number of Instances              611

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.512    0.226    0.676      0.512    0.583      0.297    0.686     0.709     Liver_disease
                 0.774    0.488    0.632      0.774    0.696      0.297    0.686     0.666     No_liver_disease
Weighted Avg.    0.648    0.363    0.653      0.648    0.642      0.297    0.686     0.687

=== Confusion Matrix ===

   a   b   <-- classified as
 150 143 |   a = Liver_disease
  72 246 |   b = No_liver_disease
```

figure 10

JRip:

This is same as the Logistic Regression. It is important to note that the low values can be also due to overfitting. Therefore, we can try reducing the sample size.

Rules:

(TB >= 0.157895) => Class=Liver_disease (138.0/34.0)

(Alkphos >= 0.24792) and (Age >= 0.151163) and (ALB >= 0.326087) => Class=Liver_disease (90.0/25.0)

=> Class=No_liver_disease (383.0/124.0)

```
Correctly Classified Instances         396               64.8118 %
Incorrectly Classified Instances       215               35.1882 %
Kappa statistic                          0.2892
Mean absolute error                      0.4143
Root mean squared error                  0.4735
Relative absolute error                 82.9926 %
Root relative squared error             94.7762 %
Total Number of Instances              611

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.529    0.242    0.668      0.529    0.590      0.295    0.683     0.678     Liver_disease
                 0.758    0.471    0.636      0.758    0.692      0.295    0.683     0.651     No_liver_disease
Weighted Avg.    0.648    0.361    0.651      0.648    0.643      0.295    0.683     0.664

=== Confusion Matrix ===

   a   b   <-- classified as
 155 138 |   a = Liver_disease
  77 241 |   b = No liver disease
```

figure 11

b. Modified Dataset:

Random Forest:

This model has 75.49% accuracy which is quite similar to the Random Forest model without attribute selection. Overall, the model performs better than the original dataset. However, it identifies No Liver Disease more accurately.

```
Correctly Classified Instances         154               75.4902 %
Incorrectly Classified Instances        50               24.5098 %
Kappa statistic                          0.5108
Mean absolute error                      0.3587
Root mean squared error                  0.4248
Relative absolute error                 71.8096 %
Root relative squared error             85.0322 %
Total Number of Instances              204

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.789    0.275    0.714      0.789    0.750      0.513    0.810     0.778     Liver_disease
                 0.725    0.211    0.798      0.725    0.760      0.513    0.810     0.820     No_liver_disease
Weighted Avg.    0.755    0.241    0.759      0.755    0.755      0.513    0.810     0.801

=== Confusion Matrix ===

  a  b   <-- classified as
 75 20 |  a = Liver_disease
 30 79 |  b = No_liver_disease
```

figure 12

Logistic Regression:

The model has 67.6471% accuracy slightly greater than the one with the attribute selection. However, the correctly identified instances are just 138.

```
Correctly Classified Instances         138               67.6471 %
Incorrectly Classified Instances        66               32.3529 %
Kappa statistic                          0.3346
Mean absolute error                      0.439
Root mean squared error                  0.4622
Relative absolute error                 87.8962 %
Root relative squared error             92.5284 %
Total Number of Instances              204

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.474    0.147    0.738      0.474    0.577      0.356    0.686     0.711     Liver_disease
                 0.853    0.526    0.650      0.853    0.738      0.356    0.686     0.690     No_liver_disease
Weighted Avg.    0.676    0.350    0.691      0.676    0.663      0.356    0.686     0.700

=== Confusion Matrix ===

  a  b   <-- classified as
 45 50 |  a = Liver_disease
 16 93 |  b = No_liver_disease
```

figure 13

JRip:

This model shows 57.3529% accuracy which is the lowest than the previous ones. The low Kappa statistics indicates the poor performance and misclassification of most of the instances.

```
Correctly Classified Instances         117               57.3529 %
Incorrectly Classified Instances        87               42.6471 %
Kappa statistic                          0.1389
Mean absolute error                      0.4516
Root mean squared error                  0.4953
Relative absolute error                 90.4169 %
Root relative squared error             99.1534 %
Total Number of Instances              204

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.505    0.367    0.545      0.505    0.525      0.139    0.605     0.593     Liver_disease
                 0.633    0.495    0.595      0.633    0.613      0.139    0.605     0.592     No_liver_disease
Weighted Avg.    0.574    0.435    0.572      0.574    0.572      0.139    0.605     0.593

=== Confusion Matrix ===

  a  b   <-- classified as
 48 47 |  a = Liver_disease
 40 69 |  b = No_liver_disease
```

figure 14

T-test:

```
Dataset                   (1) functions.Logi | (2) trees.Rando (3) rules.JRip
--------------------------------------------------------------------------------
'Indian Liver Patient Dat(100)   65.49(5.63) |   61.48(6.09)     61.48(5.75)
'Indian Liver Patient Dat(100)   63.15(3.89) |   65.20(5.89)     61.91(5.85)
--------------------------------------------------------------------------------
                                     (v/ /*) |         (0/2/0)         (0/2/0)
```

figure 15

The T-test in the **figure 15** indicates that Logistic Regression performed well without attribute selection. However, Random Forest performed well with attribute selection. While JRip continued to be a worse model selection. Therefore, Random Forest with attribute selection should be a good choice while proceeding further in the Data Mining.

Insights:

It is important to have some domain knowledge while opting a model. We can conclude following points from the above classification:
Adverse factors for Age, TB, DB: It depicts that these factors reduce the odds of being having liver disease.
Positive factor for ALB: Higher albumin levels raise the odds of having liver disease.
Odds ratios: High odds ratio for ALB might be misleading. Therefore, it needs to be handled cautiously.

Part 4 – Discretisation

It is an approach to achieve equal width binning data by discretising the numeric attributes. This is an unsupervised approach. In this part I have applied "weka.filters.unsupervised.attribute.Discretize -B 40 -M -1.0 -R first-last -precision 6" filter to the modified dataset with selected attributes to produce a modified version based on the binning technique.
I have used "Equal Width Binning" since it was giving the better accuracy after trying some other unsupervised methods. The number of bins were set to be 10 after some trial and error with 2,5,40. 10 was giving the optimised result.
The accuracy of the model is 70.098% which is almost the same as the one done without binning.

```
Correctly Classified Instances         143               70.098  %
Incorrectly Classified Instances        61               29.902  %
Kappa statistic                          0.4036
Mean absolute error                      0.3801
Root mean squared error                  0.4424
Relative absolute error                 76.0893 %
Root relative squared error             88.5664 %
Total Number of Instances              204

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.737     0.330     0.660      0.737     0.697      0.406    0.773     0.708     Liver_disease
               0.670     0.263     0.745      0.670     0.705      0.406    0.773     0.820     No_liver_disease
Weighted Avg.  0.701     0.294     0.706      0.701     0.701      0.406    0.773     0.768

=== Confusion Matrix ===

  a  b   <-- classified as
 70 25 |  a = Liver_disease
 36 73 |  b = No_liver_disease
```

figure 16

Although, the results are almost similar as seen above in the figure 16. However, we can observe that the model performance on the Liver Disease is better than before and there seems to be a better balance between the two classes. However, No_Liver disease is the better performer here.
Binning reduces the number of discrete values for a numeric attribute, which can improve the efficiency of many machine learning algorithms by limiting the search space and speeding up calculations. This technique works better with larger dataset.
Binning can help to avoid overfitting, which arises when a model understands the training data too well and fails to simplify to new data. This is because applying modified data requires the model to acquire limited parameters.

It can enhance the interpretability of model outcome by clustering comparable values together. This can help us comprehend how the model makes its predictions.

Equal-width binning can handle missing values by assigning them to the next bin. This can be useful for datasets that contain missing data. That is why I have used in this case.
Although the model accuracy is slightly better without binning. However, we will still prefer the one with binning. As we can observe in the graph the variance in the one with discretization seems slightly higher than the one without it. Which indicates there is no overfitting or underfitting and data is spread uniformly when we apply discretization.

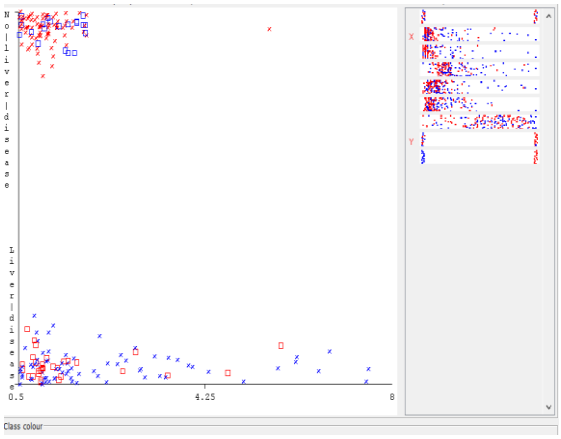

figure 17

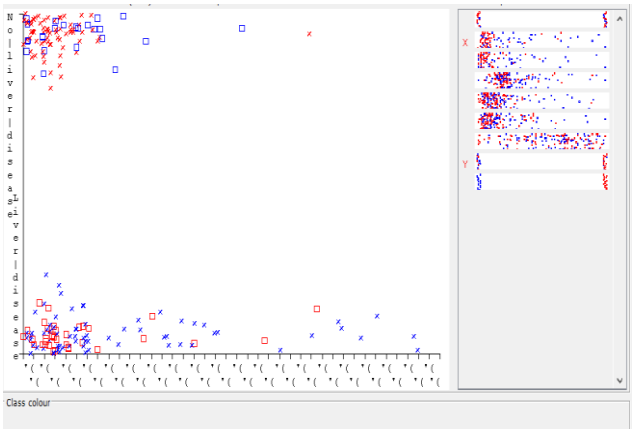

figure 18

Part 5 – Clustering

It is used to gain insights from the data and analyse the patterns of the data. In this method grouping is done on the nearest kth cluster around the centroid.

In this part we will apply the clustering to the original dataset. The dataset contains two classes Liver_Disease and No_Liver_disease. This is done by applying “weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 6 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10” under the cluster tab. Furthermore, I have also added the cluster filter by applying: “weka.filters.unsupervised.attribute.AddCluster -W "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". This will create an additional nominal attribute which can be treated as a class. As shown in the **figure 19**.

No.	Name
1	Age
2	Gender
3	TB
4	DB
5	Alkphos
6	Sgpt
7	Sgot
8	TP
9	ALB
10	A/G
11	Class
12	cluster

figure 19

We will than proceed with applying the cluster and will choose classes to cluster evaluation.

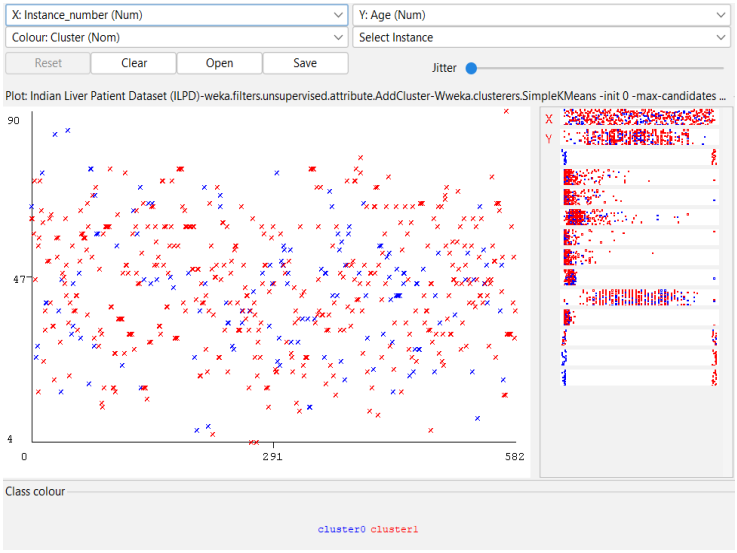

figure 20

The graph in the **figure 20** indicates two clusters and they have achieved accurate alignment with the class attribute in identifying Liver_disease and No_Liver_disease.
Furthermore, after trying clustering on 3,4 and 6 number of clusters 2 seems to be a better choice since there is no over_fitting and under_fitting and the grouping is done with a visible separation without any overlapping. As the clustering algorithm took the "Class" attribute into account, it was able to directly optimize for cluster alignment with known class boundaries. This also indicates that the attributes (TB, Age , Gender) might be a good attribute for the predictor.
A 0% error rate is desirable, representing that the k-means algorithm positively separated the data points into two distinct clusters based on the "Class" attribute used to generate the "cluster" attribute.
While this appears to be an ideal result, but it should be kept in mind that that k-means is an unsupervised algorithm that does not guarantee perfect clustering.

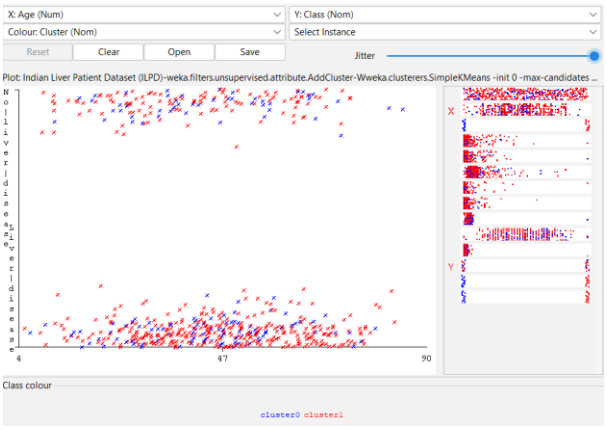

figure 21

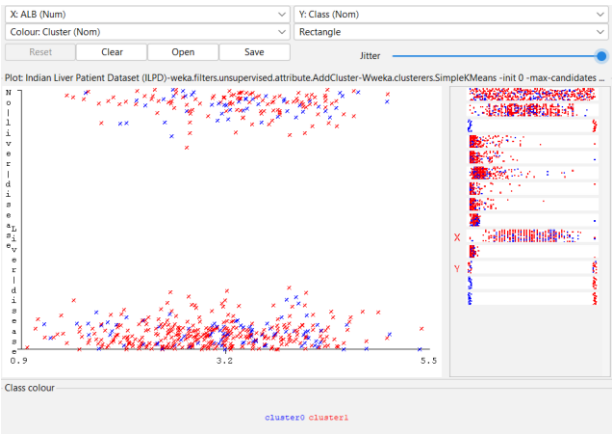

figure 22

Figure 21 is the comparison between class and age and there is poor clustering in this. Which can be further verified from the attribute selection done in the previous parts where age is a negative attribute for the class. Same pattern is identified in **Figure 22** which shows ALB comparison with the class.

figure 23

Figure 23 is the plot of DB vs Class. Although there is a possible association between these two. However, due to data being noisy and presence of outliers there seems to be fitting issues. The overlap between the two clusters draws significant attention.

figure 24

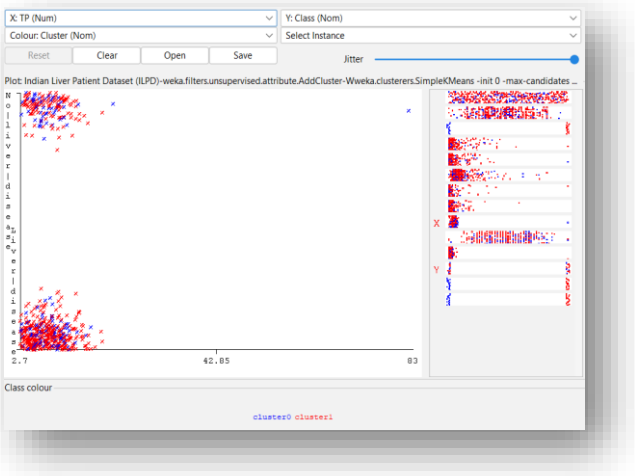

figure 25

Just like DB, TP , TB , A/G shows similar pattern in the **figure 24 and 25** respectively. Which indicates that data needs to be handle cautiously and should be pre-processed before proceeding.