23COP529 - DATA MINING

Table of Contents

23COP529 - DATA MINING

Student ID: F315284

COURSEWORK: Indian Liver Patient Dataset (ILPD) Analytics

Part 1: Data Pre-processing

Eyeballing the data set. I observed that there were missing values in the following attributes **TB** (Total Bilirubin), **DB** (Direct Bilirubin), Alkphos (Alkaline Phosphatase), **Sgot** (Aspartate Aminotransferase), TP (Total Proteins), and A/G (Ratio Albumin and Globulin Ratio). Also, observed that the data has duplicates records.

Methodology and Data/Tools used and justification:

The data preprocessing methodology involved several steps to ensure the integrity and reliability of the dataset for subsequent analysis.

1. Removing missing values

Imputing missing values in medical data carries potential risks, as inaccurate imputation could lead to incorrect conclusions or decisions regarding patient care. To prevent these potential risks, instances with missing values were removed from the dataset using Weka MultiFilter and the below Remove withValues filter for each attibutes:

"weka.filters.unsupervised.instance.RemoveWithValues -S 0.0 -C last -L first-last -M"

2. Removing duplicates

Duplicate records can introduce bias, negatively impacting the performance of predictive models. Furthermore, maintaining consistency and standardization within the dataset, duplicates records were removed using:

"weka.filters.unsupervised.instance.RemoveDuplicates"

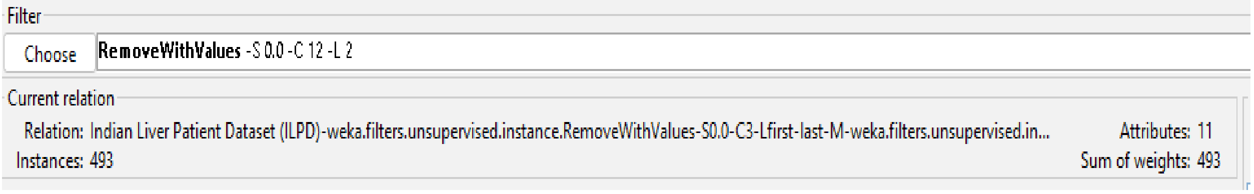
3. Removing Outliers

It's crucial to remove outliers from the dataset as they deviate significantly from the majority of data points, potentially distorting the representation of underlying phenomena and introducing noise into analyses. This noise can compromise the accuracy and reliability of models, impairing their ability to generalize effectively to unseen data. Interquartile range technique is used to identify and remove observations that deviate significantly from the norm.

***Filter**: "weka.filters.unsupervised.attribute.InterquartileRange -R first-last -O 3.0 -E 6.0. "*

Experimental Results:

The data preprocessing steps effectively addressed various data quality issues, resulting in a cleaner and more balanced dataset. Specifically, **10** instances with missing values, **13** duplicate records, and **67** outlier instances were successfully removed.



```
RemoveWithValues -S 0.0 -C 12 -L 2
```

Current relation

Relation: Indian Liver Patient Dataset (ILPD)-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C3-Lfirst-last-M-weka.filters.unsupervised.in...   Attributes: 11

Instances: 493   Sum of weights: 493

Fig 1: Output after removing missing values, duplicate values and outliers

Interpretation of these results:

The successful removal of missing values, duplicates, outliers, and class imbalance in the dataset has significant implications for subsequent analysis and interpretation. A cleaner dataset reduces the risk of bias and inaccuracies in analysis results, leading to more reliable conclusions. By addressing these data quality issues upfront, we can have greater confidence in the findings and insights derived from the dataset.

Part 2: Feature Selection

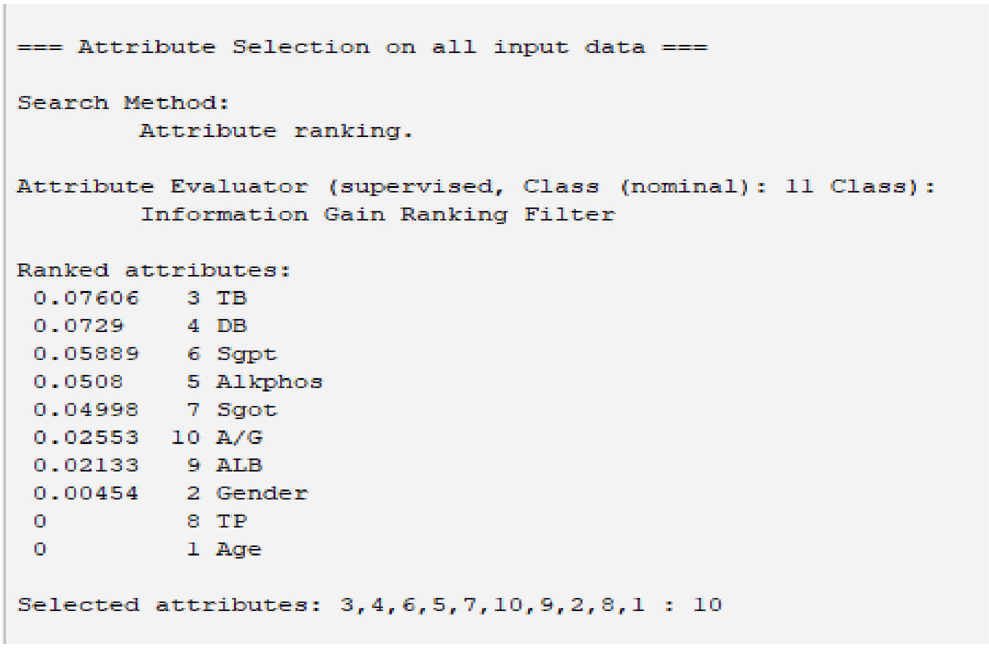
Methodology and Data/Tools used and justification:

Feature selection plays a crucial role in data mining, as irrelevant attributes can significantly impact model performance by introducing noise and complexity. To assess the predictive ability of each feature (attribute) in relation to the class variable, three widely used WEKA methods were employed:

1. **Information Gain:** Evaluates the worth of an attribute by measuring the information gain with respect to the class.
2. **Gain Ratio:** Evaluates the worth of an attribute by measuring the gain ratio with respect to the class.
3. **Correlation**: Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class.

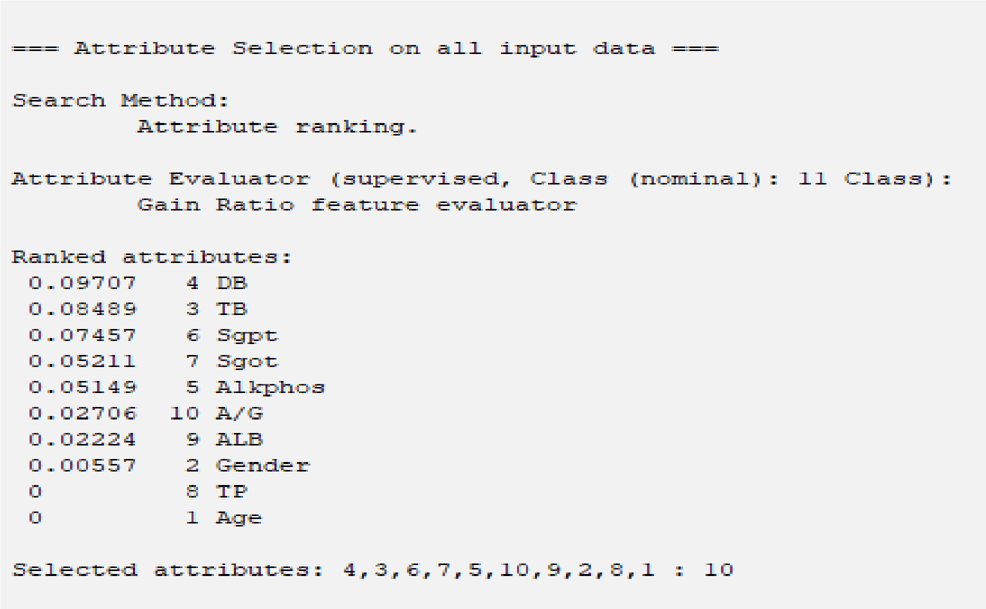
These methods were selected for their effectiveness in evaluating feature importance and their ability to provide insights into the relevance of attributes for predictive modeling. By ranking features using these methods, we gain valuable insights into which attributes contribute most significantly to predicting the target variable, thereby aiding in the development of more accurate and interpretable models.

Experimental Results:



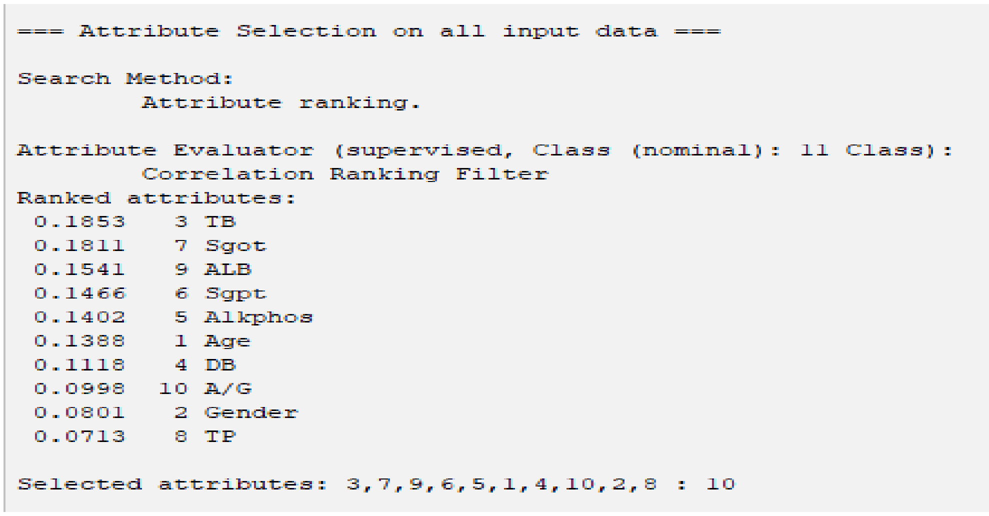
```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 11 Class):
        Information Gain Ranking Filter

Ranked attributes:
 0.07606   3 TB
 0.0729    4 DB
 0.05889   6 Sgpt
 0.0508    5 Alkphos
 0.04998   7 Sgot
 0.02553  10 A/G
 0.02133   9 ALB
 0.00454   2 Gender
 0         8 TP
 0         1 Age

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

Fig 2: Information Gain Ranking



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 11 Class):
        Gain Ratio feature evaluator

Ranked attributes:
 0.09707    4 DB
 0.08489    3 TB
 0.07457    6 Sgpt
 0.05211    7 Sgot
 0.05149    5 Alkphos
 0.02706   10 A/G
 0.02224    9 ALB
 0.00557    2 Gender
 0          8 TP
 0          1 Age

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

Fig 3: Gain Ratio Ranking



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 11 Class):
        Correlation Ranking Filter
Ranked attributes:
 0.1853    3 TB
 0.1811    7 Sgot
 0.1541    9 ALB
 0.1466    6 Sgpt
 0.1402    5 Alkphos
 0.1388    1 Age
 0.1118    4 DB
 0.0998   10 A/G
 0.0801    2 Gender
 0.0713    8 TP

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

Fig 4: Correlation Ranking

<u>Interpretation of these results:</u>

In the results obtained, Information Gain assigns scores to features, indicating their level of informativeness or relevance for predicting the class variable. TB ranks highest with a score of **0.07606**, signifying its highest relevance among features. Similarly, Gain Ratio scores assess the predictive utility of features, with DB ranking highest at **0.09707**, indicating its utmost relevance for prediction. Additionally, Correlation evaluates the linear relationship between each feature and the class variable, where TB has the highest absolute correlation of **0.1853**, suggesting its strongest association with the class variable.

Feature Name	Information Gain Rank	Gain Ratio Rank	Correlation Rank
TB	1	2	1
DB	2	1	7
Sgpt	3	3	4
Alkphos	4	5	5
Sgot	5	4	2
A/G	6	6	8
ALB	7	7	3
Gender	8	8	9
TP	9	9	10
Age	10	10	6

In tabulating the ranks, TB, DB, Sgpt, Alphos, and Sgot consistently rank among the top five in both the Information Gain and Gain Ratio methods. Similarly, in the Correlation method, TB, Sgot, Sgpt, and Alkphos rank within the top four. Consequently, TB, DB, Sgpt, Alphos, and Sgot are deemed to have relatively higher importance in predicting the class attribute. Hence, I will rank the attributes in this order TB, DB, Sgpt, Alphos, Sgot, A/G, ALB, Gender, TP and Age.

Part 3: Classification

Methodology and Data/Tools used and justification:

1. Balancing the class attributes:

Upon observation, it was noted that the class attribute (Class) is imbalanced. This imbalance poses a risk of bias in our analysis, particularly evident when fitting a ZeroR model on our dataset, which yielded an accuracy of **67.7%** by simply predicting the majority class (liver disease). To address this issue, reweighting the instances in the data was undertaken, ensuring that each class carries equal total weight.

Filter: *"weka.filters.supervised.instance.ClassBalancer -num-intervals 10 "*

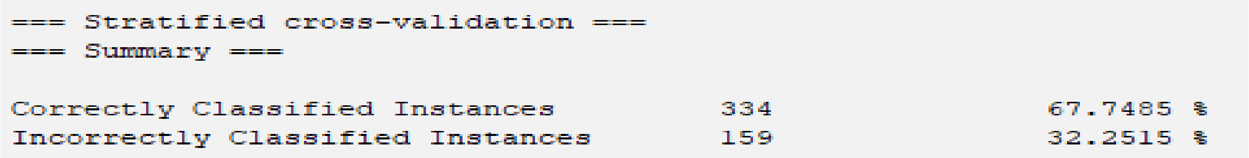
2. Applying Classifiers with 10-fold Cross-validation on datasets:

Two datasets will be utilized for evaluation: the original dataset containing all attributes and a modified dataset comprising the top-ranking attributes (TB, DB, Sgpt, Alphos, and Sgot) identified through feature selection. Using the WEKA Experimenter, three classifiers—**JRip, NaiveBayes**, and **Logistic** models—will be applied individually to both datasets, employing 10-fold cross-validation. This methodology allows for the comparison of classifier performance between the original and modified datasets, facilitating subsequent statistical t-testing.

3. Performing T-tests:

Performed paired t-tests using WEKA Experimenter to determine whether the differences in performance (correct classification) of the classifiers on the original dataset versus the modified dataset are statistically significant. This will help determine if the modifications made to the dataset based on feature selection significantly impacted the classifier performance.

Experimental Results:



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

Correctly Classified Instances         334               67.7485 %
Incorrectly Classified Instances       159               32.2515 %
```

Fig 5: Summary of ZeroR model

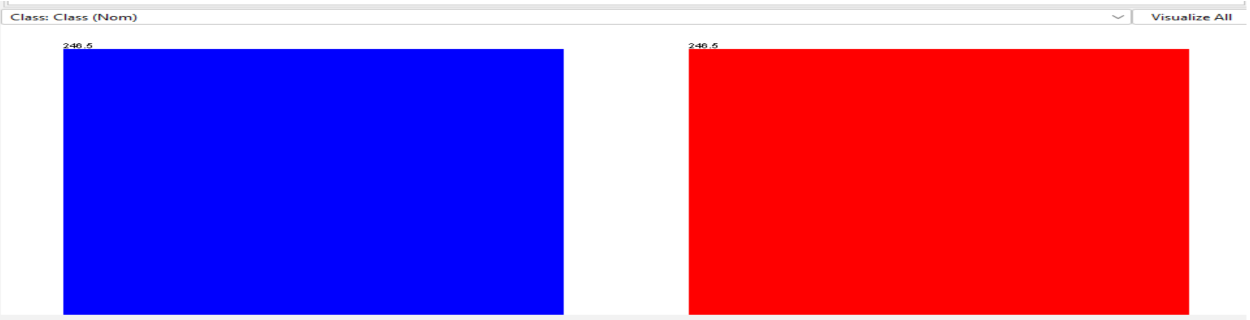
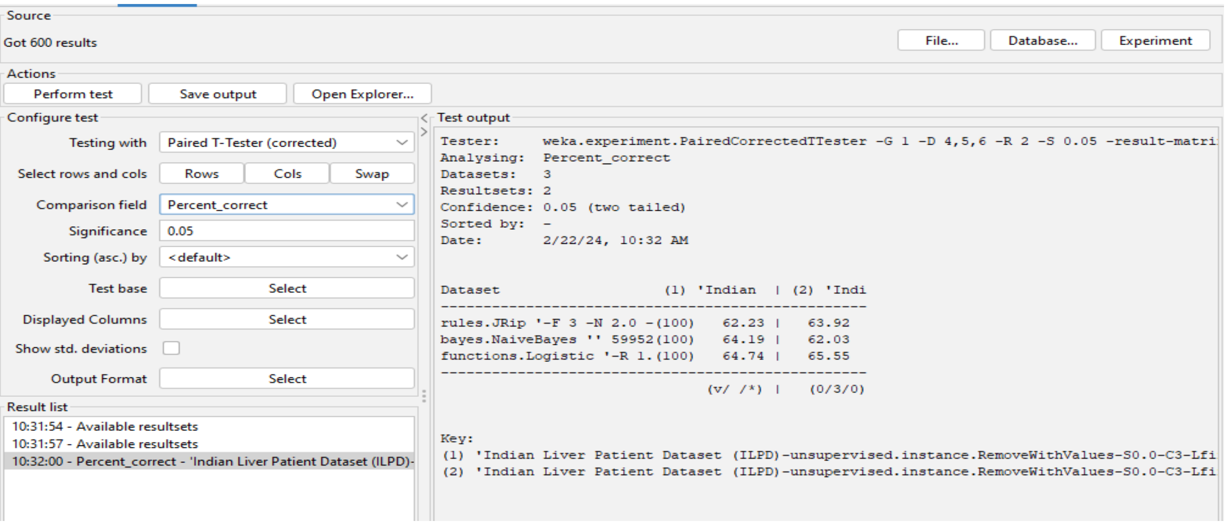


Fig 6: Output after class balancing



Got 600 results

Testing with: Paired T-Tester (corrected)

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Show std. deviations: □

10:31:54 - Available resultsets
10:31:57 - Available resultsets
10:32:00 - Percent_correct - 'Indian Liver Patient Dataset (ILPD)-

```
Tester:     weka.experiment.PairedCorrectedTTester -G 1 -D 4,5,6 -R 2 -S 0.05 -result-matri
Analysing:  Percent_correct
Datasets:   3
Resultsets: 2
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       2/22/24, 10:32 AM


Dataset                    (1) 'Indian  | (2) 'Indi
---------------------------------------------------
rules.JRip '-F 3 -N 2.0 -(100)   62.23 |   63.92
bayes.NaiveBayes '' 59952(100)   64.19 |   62.03
functions.Logistic '-R 1.(100)   64.74 |   65.55
---------------------------------------------------
                               (v/ /*) |   (0/3/0)


Key:
(1) 'Indian Liver Patient Dataset (ILPD)-unsupervised.instance.RemoveWithValues-S0.0-C3-Lfi
(2) 'Indian Liver Patient Dataset (ILPD)-unsupervised.instance.RemoveWithValues-S0.0-C3-Lfi
```

Fig 7: Output of T-tests

Interpretation of these results:

The class imbalance issue was addressed by reweighting the instances, ensuring each class carries an equal total weight of **246.5**, as depicted in the bar chart in Figure 6.

After conducting T-tests, it was found that the JRip and Logistic classifiers achieved accuracies of **62.23%** and **64.74%** on the original dataset, respectively. On the modified dataset, their accuracies improved to **63.92%** and **65.55%,** respectively. However, the NaiveBayes classifier exhibited a decrease in accuracy, achieving **62.02%** on the modified dataset compared to **64.19%** on the original dataset. The head-to-head tests against the baseline dataset (original dataset) resulted in a tie, indicating insufficient evidence to reject the null hypothesis that their performances are equivalent.

Based on the findings, the Logistic classifier consistently outperformed others, demonstrating an increase in accuracy from 64.74% to 65.55% when transitioning from the original to the modified dataset. Thus, the choice of using the Logistic classifier is justified as it exhibits the highest accuracy. Furthermore, Logistic regression is well-suited for binary classification tasks, making it an appropriate choice for predicting the class variable in medical datasets.

Part 4: Binning Strategy

Methodology and Data/Tools used and justification:

1. Discretizing Attributes:

Based on domain knowledge, elevated levels of TB, DB, Alkphos, ALT, and Sgot may signify liver dysfunction or damage, abnormal TP levels can indicate impaired protein synthesis due to liver disease, low ALB levels may suggest liver dysfunction or malnutrition, and abnormal A/G ratios may indicate liver disease or other conditions. Hence, these attributes are discretized into nominal attributes with 3 bins (low, mid and high) using WEKA discretize filter. Equal bin frequency is used to achieve a more uniform distribution of data across the bins and it is easy to interpret.

The below Weka filters were used for discretizing and renaming the nominal attributes respectively

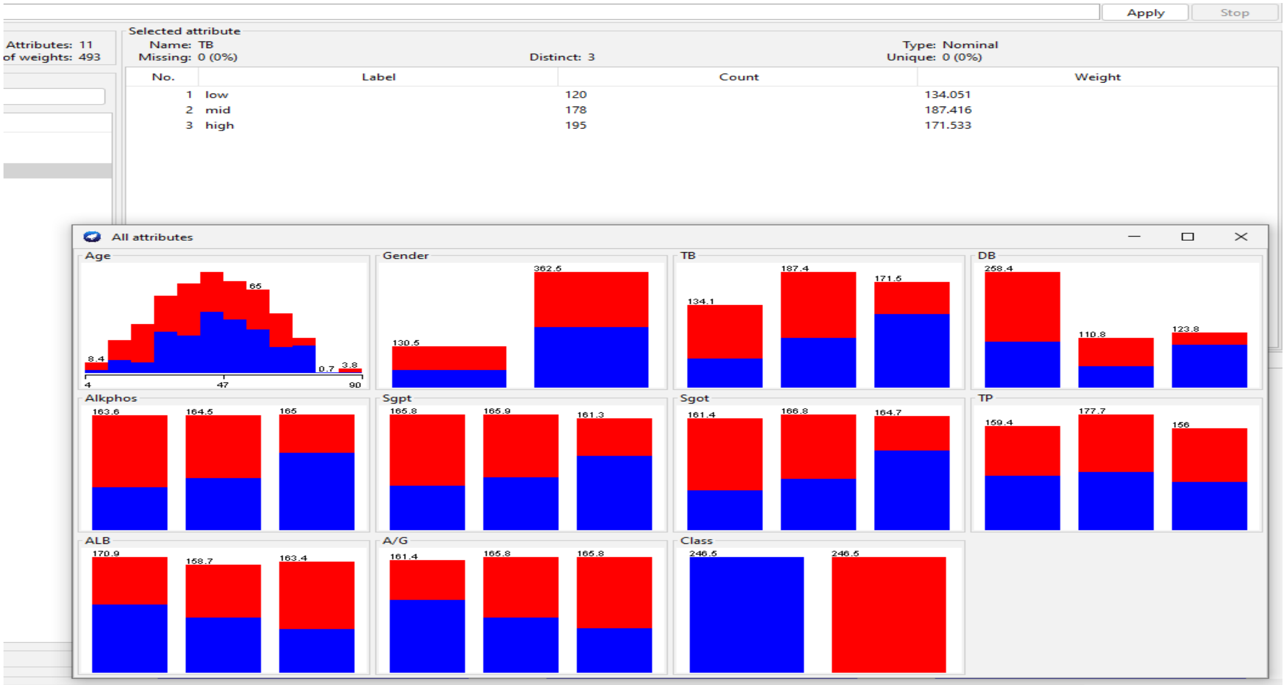
" weka.filters.unsupervised.attribute.Discretize -Y -F -B 3 -M -1.0 -R first-last -precision 6 "

" weka.filters.unsupervised.attribute.RenameNominalValues -R 3,4,5,6,7,8,9,10 -N "\'B1of3\':low,\'B2of3\':mid,\'B3of3\':high" "

2. Applying Logistic Classifier:

Logistic classifier which is the best classifier based on the findings of T-tests in Part 3 is fitted to the modified data set (discretized data set) and the original data set using WEKA logistic classifier.

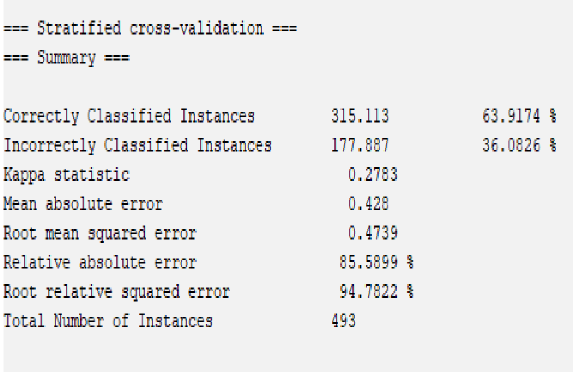
Experimental Results:



```
Attributes: 11        Selected attribute
of weights: 493         Name: TB                                        Type: Nominal
                        Missing: 0 (0%)          Distinct: 3            Unique: 0 (0%)
```

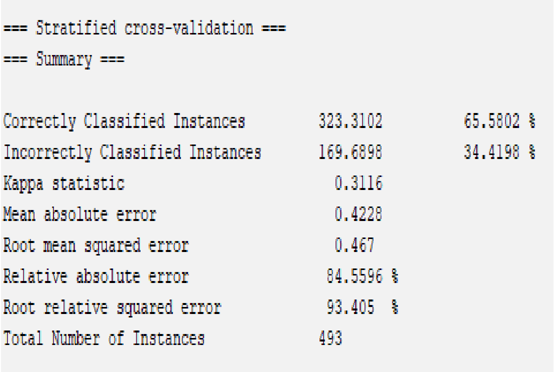
No.	Label	Count	Weight
1	low	120	134.051
2	mid	178	187.416
3	high	195	171.533

Fig 8: Visualization of attributes after discretization



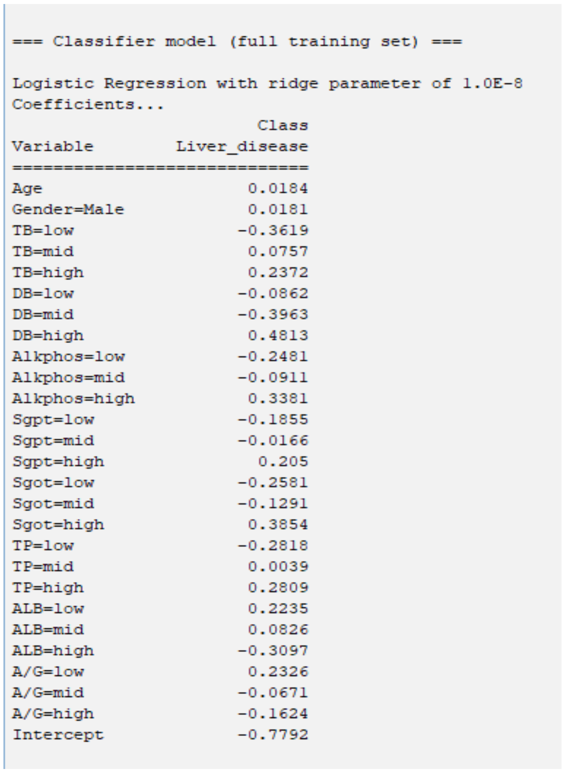
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         315.113            63.9174 %
Incorrectly Classified Instances       177.887            36.0826 %
Kappa statistic                          0.2783
Mean absolute error                      0.428
Root mean squared error                  0.4739
Relative absolute error                 85.5899 %
Root relative squared error             94.7822 %
Total Number of Instances              493
```

Fig 9: modified data model performance



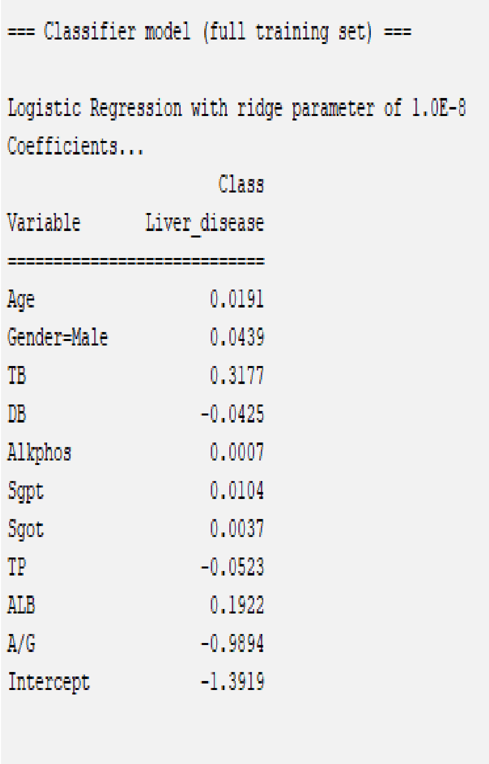
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         323.3102           65.5802 %
Incorrectly Classified Instances       169.6898           34.4198 %
Kappa statistic                          0.3116
Mean absolute error                      0.4228
Root mean squared error                  0.467
Relative absolute error                 84.5596 %
Root relative squared error             93.405  %
Total Number of Instances              493
```

Fig 10: original data model performance



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

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                               Class
Variable             Liver_disease
=============================
Age                         0.0184
Gender=Male                 0.0181
TB=low                     -0.3619
TB=mid                      0.0757
TB=high                     0.2372
DB=low                     -0.0862
DB=mid                     -0.3963
DB=high                     0.4813
Alkphos=low                -0.2481
Alkphos=mid                -0.0911
Alkphos=high                0.3381
Sgpt=low                   -0.1855
Sgpt=mid                   -0.0166
Sgpt=high                    0.205
Sgot=low                   -0.2581
Sgot=mid                   -0.1291
Sgot=high                   0.3854
TP=low                     -0.2818
TP=mid                      0.0039
TP=high                     0.2809
ALB=low                     0.2235
ALB=mid                     0.0826
ALB=high                   -0.3097
A/G=low                     0.2326
A/G=mid                    -0.0671
A/G=high                   -0.1624
Intercept                  -0.7792
```

Fig 11: coefficient on modified dataset



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

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                          Class
Variable          Liver_disease
============================
Age                      0.0191
Gender=Male              0.0439
TB                       0.3177
DB                      -0.0425
Alkphos                  0.0007
Sgpt                     0.0104
Sgot                     0.0037
TP                      -0.0523
ALB                      0.1922
A/G                     -0.9894
Intercept               -1.3919
```

Fig 12: coefficient on original dataset

Interpretation of these results:

Using the specific values of the coefficients from the logistic regression model (fig 11), we can observe the impact of different attribute levels on the prediction of liver disease. For example, for the Total Bilirubin (TB) attribute:

- The coefficient for "low" TB levels is -**0.3619**, indicating that lower TB levels are associated with a decreased likelihood of liver disease.

- The coefficient for "mid" TB levels is **0.0757**, suggesting a slightly positive association with liver disease compared to low levels.
- The coefficient for "high" TB levels is **0.2372**, indicating a stronger positive association with liver disease compared to both low and mid-levels.

Similarly, we can interpret the coefficients for other attributes such as Direct Bilirubin (DB), Alkaline Phosphatase (Alkphos), Alamine Aminotransferase (Sgpt or ALT), Aspartate Aminotransferase (Sgot), Total Proteins (TP), Albumin (ALB), and Albumin/Globulin Ratio (A/G).

While the binning strategy may result in a slightly lower accuracy of **63.91%** on the modified dataset compared to the accuracy of **65.58%** on the original dataset without discretization, the interpretation of the coefficients provides valuable insights into the relative importance of different attribute levels in predicting liver disease. By aligning with domain knowledge about liver disease indicators, the binning strategy enhances the interpretability of the dataset, allowing for a more intuitive understanding of the relationships between attribute levels and the presence of liver disease. This trade-off between accuracy and interpretability should be considered based on the specific requirements of the analysis and the importance of understanding the underlying factors contributing to liver disease prediction.

Part 5: Clustering

Methodology and Data/Tools used and justification:

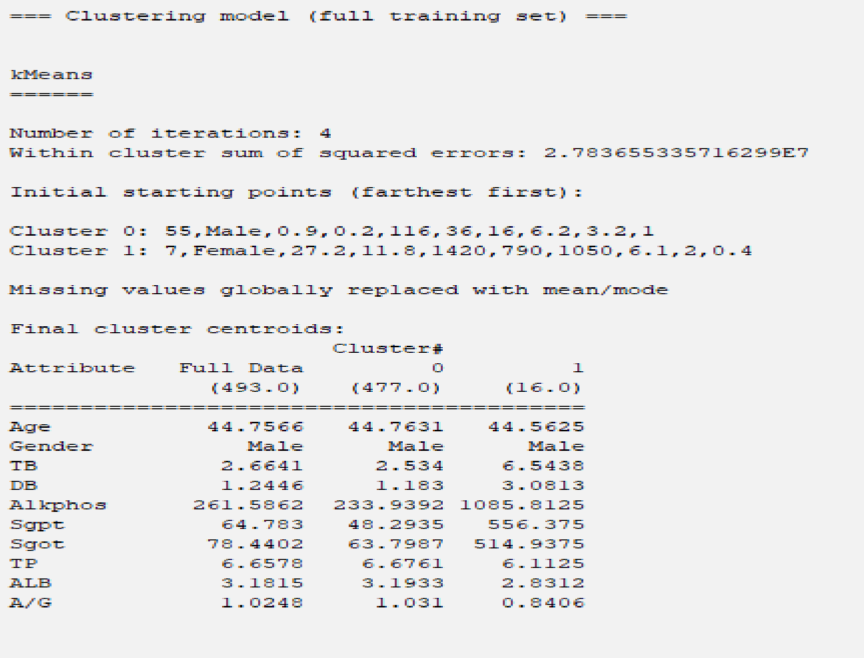
1. K-means clustering:

The Weka K-means clustering algorithm is applied to the original dataset to gain deeper insights into the data structure, utilizing the Euclidean distance metric for similarity calculation and mean-based centroid computation. The initialization parameter for k-means is adjusted to "farthest first" to enhance clustering accuracy by ensuring more optimal starting centroids. Furthermore, the number of clusters is set to two, aligning with the number of classes in the target attribute (Liver disease and No Liver disease). This approach allows for a comprehensive exploration of the dataset's inherent patterns and relationships, facilitating better understanding and potentially uncovering hidden insights within the data.

2. Evaluating the performance of K-means algorithm:

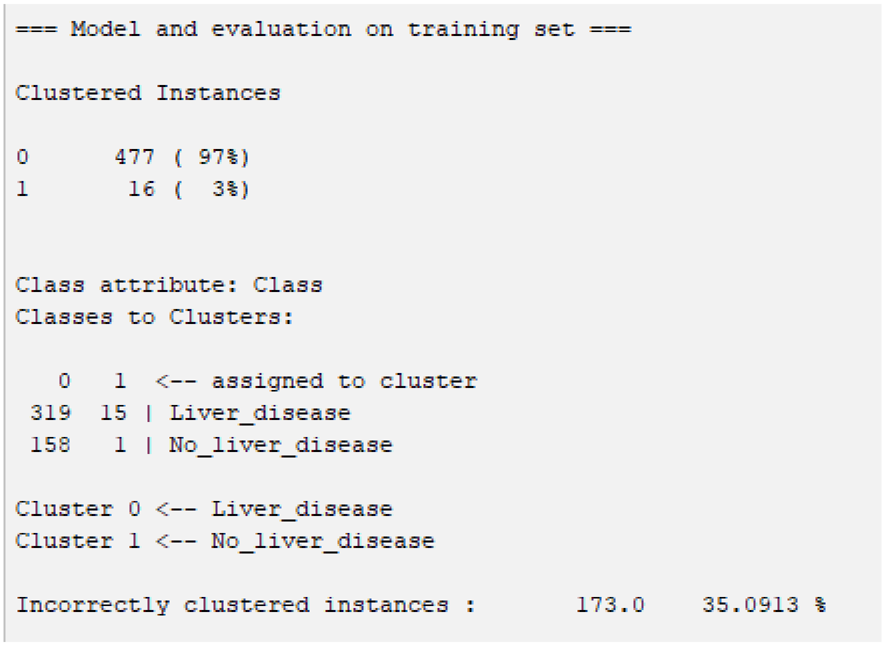
To assess the effectiveness of the K-means algorithm, I compared its clustering results against the ground-truth labels provided in the class attribute. Utilizing the "Classes to clusters evaluation" method in Weka, the algorithm disregards the class attribute during clustering generation. Subsequently, during the testing phase, it assigns classes to the clusters based on the majority value of the class attribute within each cluster. This process allows for the computation of classification error, facilitating the generation of a corresponding confusion matrix to further analyze the performance of the algorithm.

Experimental Results:



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


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 2.783655335716299E7

Initial starting points (farthest first):

Cluster 0: 55,Male,0.9,0.2,116,36,16,6.2,3.2,1
Cluster 1: 7,Female,27.2,11.8,1420,790,1050,6.1,2,0.4

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
               (493.0)    (477.0)     (16.0)
============================================
Age            44.7566    44.7631    44.5625
Gender            Male       Male       Male
TB              2.6641      2.534     6.5438
DB              1.2446      1.183     3.0813
Alkphos       261.5862   233.9392  1085.8125
Sgpt            64.783    48.2935    556.375
Sgot           78.4402    63.7987   514.9375
TP              6.6578     6.6761     6.1125
ALB             3.1815     3.1933     2.8312
A/G             1.0248      1.031     0.8406
```

Fig 13: Output of K-means clustering



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

Clustered Instances

0      477 ( 97%)
1       16 (  3%)


Class attribute: Class
Classes to Clusters:

   0   1  <-- assigned to cluster
 319  15 | Liver_disease
 158   1 | No_liver_disease

Cluster 0 <-- Liver_disease
Cluster 1 <-- No_liver_disease

Incorrectly clustered instances :       173.0    35.0913 %
```

Fig 14: confusion matrix and classification error

<u>Interpretation of these results:</u>

Upon examining the k-means result output in Fig13, it's notable that for cluster 0 (representing Liver disease), the means of the numeric attributes within the cluster are relatively closer to the overall dataset mean compared to cluster 1 (representing No liver disease). Given the original dataset's skew towards instances of the liver disease class (334 instances versus 159 instances for No liver disease), the algorithm performed reasonably well in clustering.

Additionally, evaluating the algorithm's performance reveals that it misclassified 173 instances, resulting in a classification error rate of 35.09%. Therefore, we can infer that the model achieves a classification accuracy of 64.91% when classifying new data with a similar distribution.

Further analysis of the confusion matrix yields the following performance metrics:

- Precision: 66.88%
- Recall: 95.5%
- F-measure: 78.67%

These metrics demonstrate relatively good performance, particularly in terms of identifying instances of Liver disease, which is of primary concern in this context.