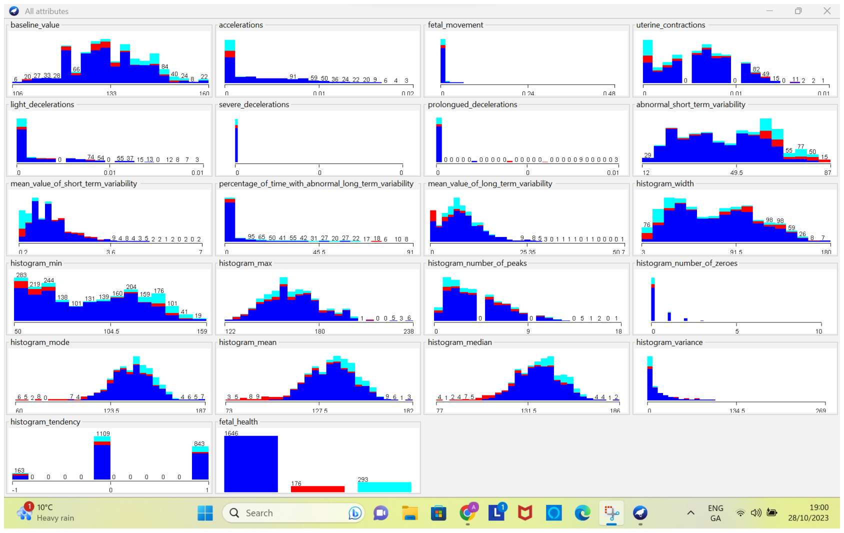
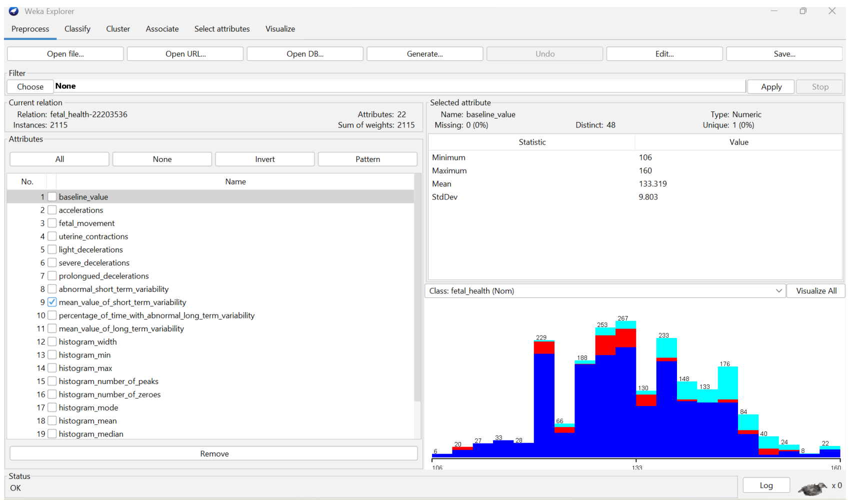
Assignment 1

Student number 22203536





Filter
Choose None

Current relation
Relation: fetal_health-22203536
Instances: 2115
Attributes: 22
Sum of weights: 2115

Selected attribute
Name: baseline_value
Missing: 0 (0%)
Distinct: 48
Type: Numeric
Unique: 1 (0%)

Statistic	Value
Minimum	106
Maximum	160
Mean	133.319
StdDev	9.803

Class: fetal_health (Nom)

Attributes

No.	Name
1	baseline_value
2	accelerations
3	fetal_movement
4	uterine_contractions
5	light_decelerations
6	severe_decelerations
7	prolongued_decelerations
8	abnormal_short_term_variability
9	✓ mean_value_of_short_term_variability
10	percentage_of_time_with_abnormal_long_term_variability
11	mean_value_of_long_term_variability
12	histogram_width
13	histogram_min
14	histogram_max
15	histogram_number_of_peaks
16	histogram_number_of_zeroes
17	histogram_mode
18	histogram_mean
19	histogram_median

Status
OK

1.1

Upon first inspection the personalized fetal health dataset I was assigned had 22 features and 2115 instances. I first checked for missing data. No missing data was found. I then removed duplicate rows as these could be double entries. There were only 13 duplicate rows and after removal this left 2102 instances. I then removed outliers by interquartile range method and using Weka's remove with values feature after calculating outliers. After removal of outliers there were 1800 instances in the data. The purpose of removing outliers was for KNN since it relies on distance, outliers can cause incorrect classifications and for outliers can affect the quality of splits in the decision tree classifier. I then binned several of the features. Binning can simplify decision trees and make them more interpretable and can reduce the influence of small variations in continuous data for KNN classifiers. I normalized all the numerical features between 0 and 1. For KNN since it relies on distance features on larger scales can affect the outcome. Since Naive Bayes deals in probabilities normalization between 0 and 1 can also help here and also helps make it easier to interpret. I removed several features with low variance such as fetal_movement, severe_decelerations and prolonged_decelerations as I did not expect them to contribute meaningfully to predictions. These features also had a high number of 0s or a high percentage of data skewed toward 0. Data that has little variation is often not helpful for making predictions.

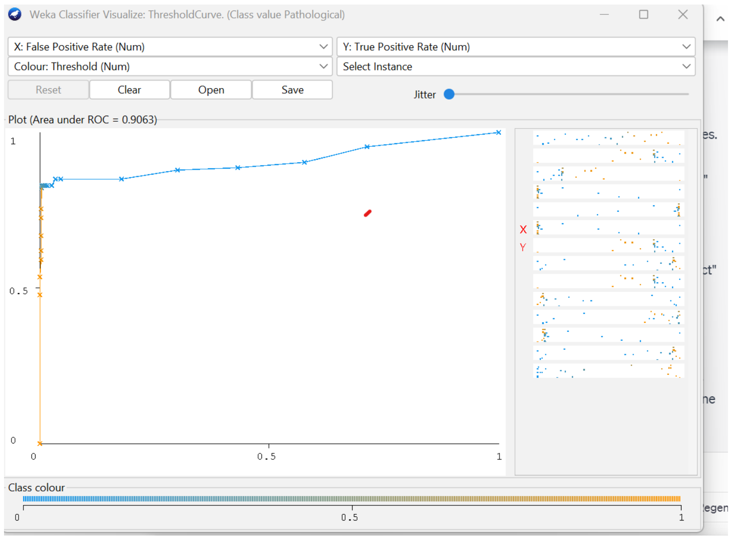
1.2
After thoroughly examining the 3 classifiers. The decision tree classifier was shown to have the highest performance. At k= 5 it accurately classified 92.111% of the target. The KNN classifier came in second, classifying 88.7778% of the training examples and the worst classifier was naive bayes. After trial and error the optimal value for k found for KNN was 50. Above and below this the performance was slightly worse. For all 3 classifiers the performance at k=3 and k=5 was virtually identical. Using 1/d weighting did not in fact change the accuracy. The metrics that I selected were accuracy, precision, recall , f1 score and kappa statistic. This was due to the nature of the problem being a multi-class classification problem. Accuracy gives a general idea of how often the classifier is correct across classes. Precision and recall give you an idea of the sensitivity/ specificity of the classifier. F1 gives you a balanced measure of both and kappa measures agreement between predicted and actual classifications taking chance agreement into account.

Further benchmarks for comparison are available in the table below for k=5. These benchmarks are the weighted averages for all classes including Pathological, Suspect, and Normal which will indicate general performance when dealing with fetal health.

Classifier	Accuracy	Recall	Precision	Kappa	F1 score
KNN	88.7778%	0.888	0.885	0.6003	0.886
Naive bayes	78.9444%	0.789	0.869	0.4616	0.815
Decision tree	92.111%	0.921	0.917	0.7095	0.918

As you can see from the table the Decision tree classifier performed best. Screenshots of the classifiers and their performances are available in a folder submitted with this pdf including the relevant .png (some but not all of which are included in this pdf for brevity).

1.3
First we will look at the **Decision tree classifier.**



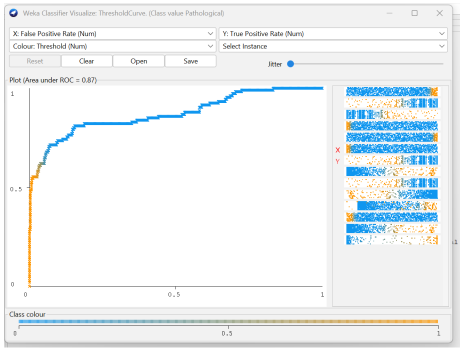
Weka Classifier Visualize: ThresholdCurve. (Class value Pathological)

X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance

Plot (Area under ROC = 0.9063)

The ROC curve for the "Pathological" class indicates that the classifier demonstrates outstanding performance in distinguishing between "Pathological" and other outcomes. The Area Under the Curve (AUC) value is 0.9063, which is near the maximum value of 1. This suggests that the model is highly effective in predicting the "Pathological" class, with the curve being close to the top-left corner, a hallmark of an excellent classifier. In practical terms, this means that the classifier does a commendable job at balancing sensitivity and specificity for this class, and can be considered reliable for predicting pathological cases.

Naive Bayes



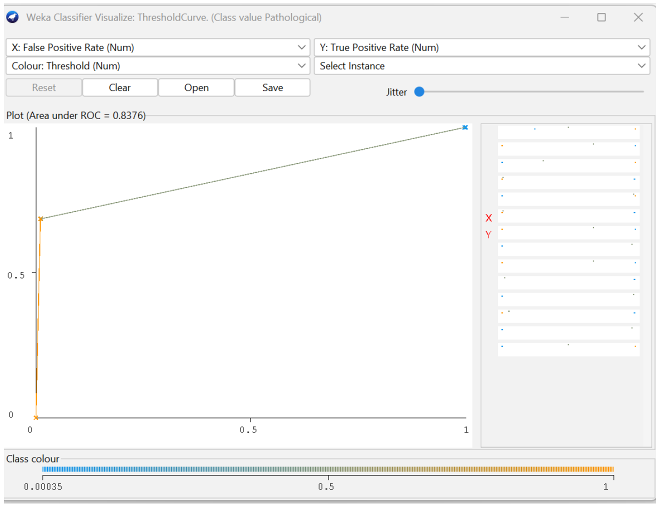
Weka Classifier Visualize: ThresholdCurve. (Class value Pathological)

X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance

Plot (Area under ROC = 0.87)

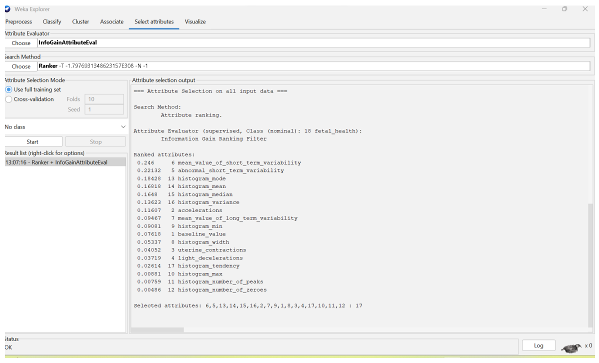
The ROC curve presented here represents the "Pathological" class performance of a Naive Bayes classifier. The Area Under the Curve (AUC) for this classifier is 0.87. Compared to the first classifier with an AUC of 0.9063, the Naive Bayes classifier shows slightly inferior performance in distinguishing between "Pathological" and other outcomes. While an AUC of 0.87 is still commendable and indicates a good classifier, it's not as close to the perfect score of 1 as the previous classifier. This suggests that, between the two, the first classifier has a slightly better capability of correctly classifying the instances, achieving both higher sensitivity and specificity for the "Pathological" class.

KNN



The ROC curve depicted here corresponds to the "Pathological" class performance of a KNN classifier. The Area Under the Curve (AUC) for this classifier is 0.8376. When compared with the decision tree classifier AUC of 0.9063 and the Naive Bayes classifier's AUC of 0.87, the KNN classifier demonstrates a lower performance in discerning between "Pathological" and other outcomes Although an AUC of 0.8376 is respectable and indicates a reasonably good classifier, it falls short in performance when juxtaposed with the other two classifiers. This suggests that the KNN classifier might not be as adept at achieving a balance between sensitivity and specificity for the "Pathological" class as its counterparts. Essentially among the three classifiers evaluated, the first one remains superior, followed by the Naive Bayes, with KNN trailing behind in terms of ROC performance for the "Pathological" class. I am satisfied with the performance of the decision tree classifier but there is always room for improvement in high stakes applications like fetal health.

1.4



Attribute Evaluator: Choose InfoGainAttributeEval

Search Method: Choose Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: [✓ Use full training set] Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options): 13:07:16 - Ranker + InfoGainAttributeEval

Attribute selection output

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

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 18 fetal_health):
        Information Gain Ranking Filter

Ranked attributes:
 0.246     6 mean_value_of_short_term_variability
 0.22132   5 abnormal_short_term_variability
 0.18428  13 histogram_mode
 0.16818  14 histogram_mean
 0.1648   15 histogram_median
 0.13623  16 histogram_variance
 0.11607   2 accelerations
 0.09467   7 mean_value_of_long_term_variability
 0.09081   9 histogram_min
 0.07618   1 baseline_value
 0.05337   8 histogram_width
 0.04052   3 uterine_contractions
 0.03719   4 light_decelerations
 0.02614  17 histogram_tendency
 0.00881  10 histogram_max
 0.00759  11 histogram_number_of_peaks
 0.00486  12 histogram_number_of_zeroes

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

Status
OK

I utilized the Information Gain attribute evaluator paired with the Ranker search method. The aim was to gauge the significance of each attribute in relation to the prediction of fetal health. Upon loading the dataset in
Rankings

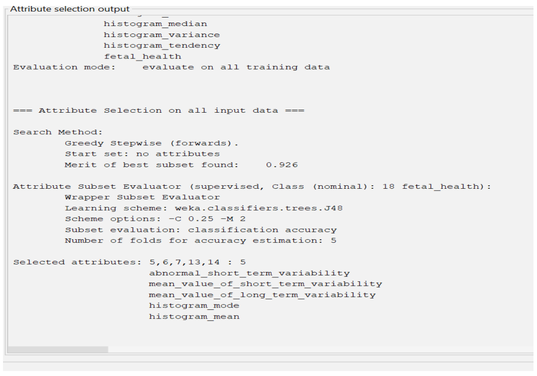
1. **mean_value_of_short_term_variability** (0.246): Taking the lead, this attribute has emerged as the most informative one in the dataset with the highest information gain.
2. . **abnormal_short_term_variability** (0.22132): A close second, this attribute too showcases high discriminative power in the context of fetal health prediction.
3. **histogram_mode** (0.18428): Following the top two, the mode of the histogram is third in rank, indicating its relevance.
4. **histogram_mean** (0.1618) and histogram_median (0.1648): These attributes, representing the mean and median of the histogram, are also among the top attributes, indicating their importance.

The subsequent attributes in the ranking provide decreasing levels of information gain. While they still hold value, their discriminative power diminishes compared to the top-ranked attributes.

Implications:
The results show the potential hierarchy of attributes when constructing a predictive model. The attributes with the highest information gain are crucial and should ideally be prioritized in the modeling phase. While the complete list of selected attributes encompasses the majority of the dataset's features, it might be more efficient to focus on the top attributes, especially if a parsimonious model is desired.
In conclusion, based on the obtained attribute rankings, the next steps in the modeling process should emphasize the top-ranked attributes for optimal performance.

1.5

Using Wrapper plus forward search to find the top 5 features.

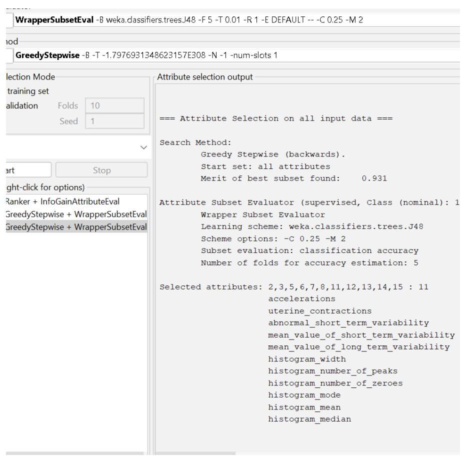


```
              histogram_median
              histogram_variance
              histogram_tendency
              fetal_health
Evaluation mode:    evaluate on all training data



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

Search Method:
        Greedy Stepwise (forwards).
        Start set: no attributes
        Merit of best subset found:    0.926

Attribute Subset Evaluator (supervised, Class (nominal): 18 fetal_health):
        Wrapper Subset Evaluator
        Learning scheme: weka.classifiers.trees.J48
        Scheme options: -C 0.25 -M 2
        Subset evaluation: classification accuracy
        Number of folds for accuracy estimation: 5

Selected attributes: 5,6,7,13,14 : 5
                     abnormal_short_term_variability
                     mean_value_of_short_term_variability
                     mean_value_of_long_term_variability
                     histogram_mode
                     histogram_mean
```

Top 5

1. abnormal_short_term_variability
2. mean_value_of_short_term_variability
3. mean_value_of_long_term_variability
4. histogram_mode
5. histogram_mean

Using Wrapper plus search backwards



WrapperSubsetEval -B weka.classifiers.trees.J48 -F 5 -T 0.01 -R 1 -E DEFAULT -- -C 0.25 -M 2

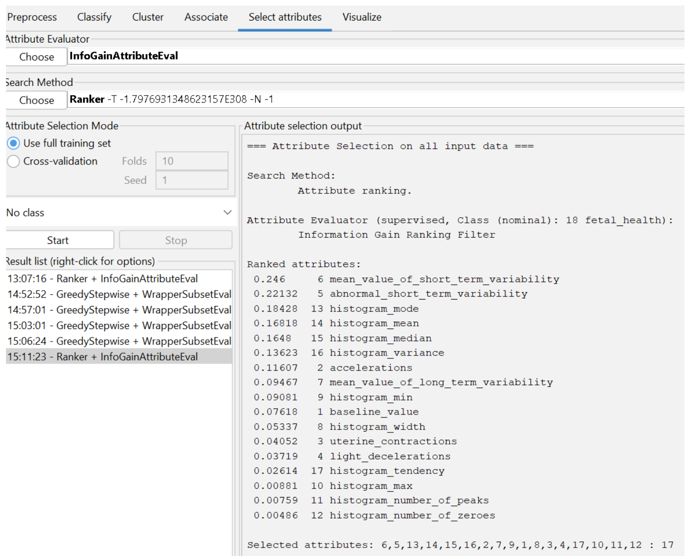
GreedyStepwise -B -T -1.7976931348623157E308 -N -1 -num-slots 1

Ranker + InfoGainAttributeEval
GreedyStepwise + WrapperSubsetEval
GreedyStepwise + WrapperSubsetEval

Attribute selection output

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

Search Method:
        Greedy Stepwise (backwards).
        Start set: all attributes
        Merit of best subset found:    0.931

Attribute Subset Evaluator (supervised, Class (nominal): 1
        Wrapper Subset Evaluator
        Learning scheme: weka.classifiers.trees.J48
        Scheme options: -C 0.25 -M 2
        Subset evaluation: classification accuracy
        Number of folds for accuracy estimation: 5

Selected attributes: 2,3,5,6,7,8,11,12,13,14,15 : 11
                     accelerations
                     uterine_contractions
                     abnormal_short_term_variability
                     mean_value_of_short_term_variability
                     mean_value_of_long_term_variability
                     histogram_width
                     histogram_number_of_peaks
                     histogram_number_of_zeroes
                     histogram_mode
                     histogram_mean
                     histogram_median
```

Using information gain



Attribute Evaluator: **InfoGainAttributeEval**

Search Method: **Ranker** -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: [✓ Use full training set] Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)
13:07:16 - Ranker + InfoGainAttributeEval
14:52:52 - GreedyStepwise + WrapperSubsetEval
14:57:01 - GreedyStepwise + WrapperSubsetEval
15:03:01 - GreedyStepwise + WrapperSubsetEval
15:06:24 - GreedyStepwise + WrapperSubsetEval
15:11:23 - Ranker + InfoGainAttributeEval

Attribute selection output

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

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 18 fetal_health):
        Information Gain Ranking Filter

Ranked attributes:
 0.246      6 mean_value_of_short_term_variability
 0.22132    5 abnormal_short_term_variability
 0.18428   13 histogram_mode
 0.16818   14 histogram_mean
 0.1648    15 histogram_median
 0.13623   16 histogram_variance
 0.11607    2 accelerations
 0.09467    7 mean_value_of_long_term_variability
 0.09081    9 histogram_min
 0.07618    1 baseline_value
 0.05337    8 histogram_width
 0.04052    3 uterine_contractions
 0.03719    4 light_decelerations
 0.02614   17 histogram_tendency
 0.00881   10 histogram_max
 0.00759   11 histogram_number_of_peaks
 0.00486   12 histogram_number_of_zeroes

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

Using information gain and ranker for the search method

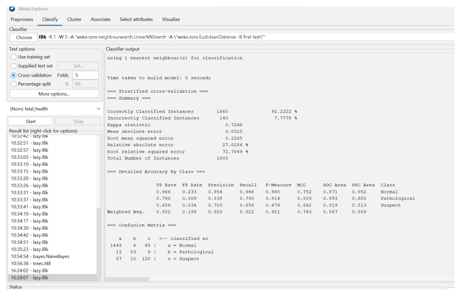
1. mean_value_of_short_term_variability

2. abnormal_short_term_variability
3. histogram_mode
4. histogram_mean
5. histogram_median

After comparing the feature selections from the three methods - Ranker with InfoGainAttributeEval, GreedyStepwise with WrapperSubsetEval (backwards search), and GreedyStepwise with WrapperSubsetEval (forwards search) - I infer several observations. The Ranker with InfoGainAttributeEval method highlights features such as "mean_value_of_short_term_variability," "abnormal_short_term_variability," and "histogram_mode" due to their high information gain scores. The forwards GreedyStepwise approach, focusing on classifier performance, identifies "abnormal_short_term_variability," "mean_value_of_short_term_variability," "mean_value_of_long_term_variability," "histogram_mode," and "histogram_mean" as its top attributes. However, the backwards GreedyStepwise method did not conclusively identify its top 5 features in the provided data. In summary, while InfoGain focuses on data entropy reduction, both GreedyStepwise methods emphasized classifier effectiveness. 4 out of 5 features were the same for 2 methods and the slight differences in selected attributes among the methods signify the nuances each method brings to the table in determining feature importance.

1.6

Top 5 features KNN



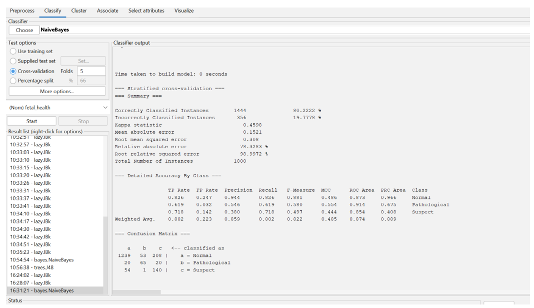
Classifier: IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options: Cross-validation, Folds 5

(Nom) fetal_health

```
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        1660               92.2222 %
Incorrectly Classified Instances       140                7.7778 %
Kappa statistic                          0.7248
Mean absolute error                      0.0525
Root mean squared error                  0.2265
Relative absolute error                 27.0286 %
Root relative squared error             72.7849 %
Total Number of Instances             1800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.966    0.233    0.954      0.966    0.960      0.752    0.871     0.952     Normal
                 0.790    0.009    0.838      0.790    0.814      0.803    0.893     0.682     Pathological
                 0.656    0.034    0.703      0.656    0.679      0.642    0.819     0.513     Suspect
Weighted Avg.    0.922    0.199    0.920      0.922    0.921      0.743    0.867     0.889

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1449    6   45 |    a = Normal
   13   83    9 |    b = Pathological
   57   10  128 |    c = Suspect
```

Top 5 features naive Bayes



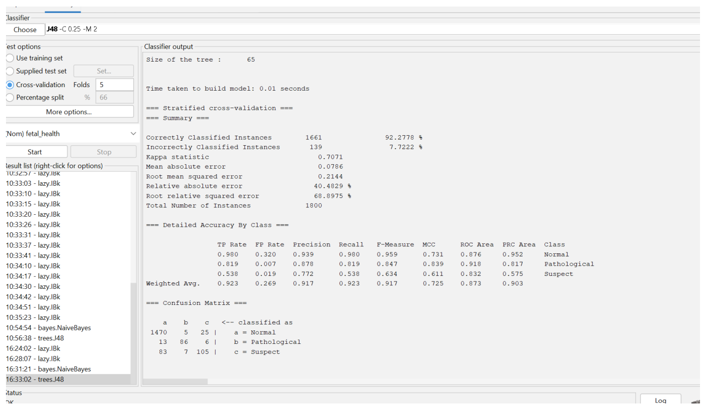
Classifier: NaiveBayes

Test options: Cross-validation, Folds 5

(Nom) fetal_health

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances        1444               80.2222 %
Incorrectly Classified Instances       356               19.7778 %
Kappa statistic                          0.4598
Mean absolute error                      0.1521
Root mean squared error                  0.308
Relative absolute error                 78.3283 %
Root relative squared error             98.9972 %
Total Number of Instances             1800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.826    0.247    0.944      0.826    0.881      0.486    0.873     0.966     Normal
                 0.619    0.032    0.546      0.619    0.580      0.554    0.914     0.675     Pathological
                 0.718    0.142    0.380      0.718    0.497      0.444    0.854     0.408     Suspect
Weighted Avg.    0.802    0.223    0.859      0.802    0.822      0.485    0.874     0.889

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1239   53  208 |    a = Normal
   20   65   20 |    b = Pathological
   54    1  140 |    c = Suspect
```

Top 5 features for decision tree classifier



J48 -C 0.25 -M 2

Cross-validation Folds 5

% 66

(Nom) fetal_health

```
10:32:57 - lazy.IBk
10:33:03 - lazy.IBk
10:33:10 - lazy.IBk
10:33:15 - lazy.IBk
10:33:20 - lazy.IBk
10:33:26 - lazy.IBk
10:33:31 - lazy.IBk
10:33:37 - lazy.IBk
10:33:41 - lazy.IBk
10:34:10 - lazy.IBk
10:34:17 - lazy.IBk
10:34:30 - lazy.IBk
10:34:42 - lazy.IBk
10:34:51 - lazy.IBk
10:35:23 - lazy.IBk
10:54:54 - bayes.NaiveBayes
10:56:38 - trees.J48
16:24:02 - lazy.IBk
16:28:07 - lazy.IBk
16:31:21 - bayes.NaiveBayes
16:33:02 - trees.J48
```

Classifier output

```
Size of the tree :      65


Time taken to build model: 0.01 seconds

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

Correctly Classified Instances        1661               92.2778 %
Incorrectly Classified Instances       139                7.7222 %
Kappa statistic                          0.7071
Mean absolute error                      0.0786
Root mean squared error                  0.2144
Relative absolute error                 40.4829 %
Root relative squared error             68.8975 %
Total Number of Instances             1800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.320    0.939      0.980    0.959      0.731    0.876     0.952     Normal
                 0.819    0.007    0.878      0.819    0.847      0.839    0.918     0.817     Pathological
                 0.538    0.019    0.772      0.538    0.634      0.611    0.832     0.575     Suspect
Weighted Avg.    0.923    0.269    0.917      0.923    0.917      0.725    0.873     0.903

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1470    5   25 |    a = Normal
   13   86    6 |    b = Pathological
   83    7  105 |    c = Suspect
```

OK

1. First we will compare the accuracy :
- Decision tree classifier: Achieved an accuracy of approximately 92.2778%.
- Naive Bayes: Achieved an accuracy of approximately 80.2222%.
- KNN: Achieved an accuracy of approximately 92.2222%.

From the accuracy standpoint, both the Decision Tree and k-NN classifiers are close, with the Decision Tree slightly edging out the k-NN. The Naive Bayes classifier, however, has a significantly lower accuracy in comparison.
2. Next we will compare the kappa statistic
- Decision Tree (J48): Achieved a Kappa statistic of 0.7071.
- Naive Bayes: Achieved a Kappa statistic of 0.4598.
- KNN: Achieved a Kappa statistic of 0.7248.

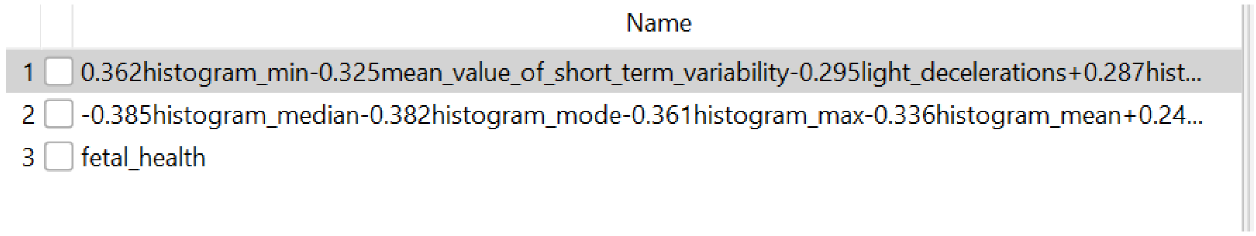
The Kappa statistic provides an indication of how well the classifier is performing in comparison to random classification. Higher values of Kappa signify better classification. Both the Decision Tree and KNN have good values, with KNN slightly outperforming the Decision Tree. Naive Bayes has a lower Kappa statistic.
3. Class-wise Performance: For 'Normal' class:
- Decision Tree classifier: TP Rate of 0.980 and Precision of 0.939.
- Naive Bayes: TP Rate of 0.826 and Precision of 0.944.
- KNN: TP Rate of 0.966 and Precision of 0.954.

For 'Pathological' class:
- Decision Tree (J48): TP Rate of 0.819 and Precision of 0.878.
- Naive Bayes: TP Rate of 0.619 and Precision of 0.546.
- k-NN: TP Rate of 0.790 and Precision of 0.838.

From the above metrics, the Decision Tree and k-NN have comparable performances for the 'Normal' class, with k-NN slightly outperforming the Decision Tree. For the 'Pathological' class, the Decision Tree has a better performance than k-NN. Naive Bayes lags behind in both classes.
4. Confusion Matrix: Based on the provided data for the confusion matrix, the Decision Tree and k-NN have fewer misclassifications for the 'Normal' and 'Pathological' classes compared to the Naive Bayes classifier. The Decision Tree particularly has fewer misclassifications for the 'Pathological' class compared to k-NN.

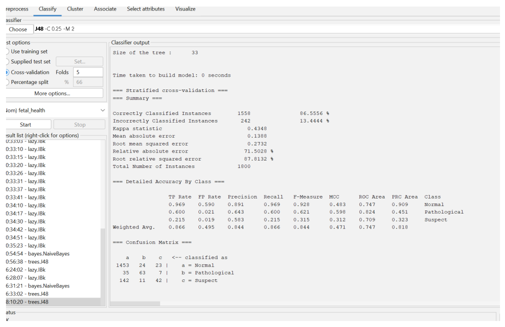
1.7

I first applied the principal components filter to get the 2 principal components.



		Name
1	☐	0.362histogram_min-0.325mean_value_of_short_term_variability-0.295light_decelerations+0.287hist...
2	☐	-0.385histogram_median-0.382histogram_mode-0.361histogram_max-0.336histogram_mean+0.24...
3	☐	fetal_health

Decision tree



Classifier: J48 -C 0.25 -M 2

Test options: Cross-validation Folds 5

(Nom) fetal_health

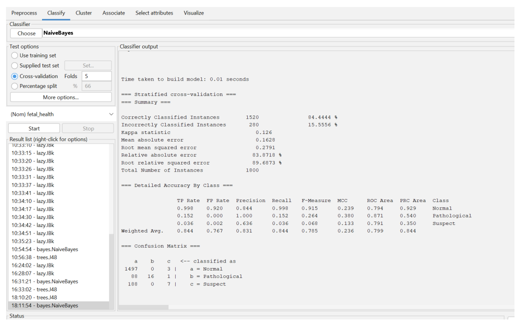
Result list (right-click for options)

```
0:33:03 - lazy.IBk
0:33:10 - lazy.IBk
0:33:15 - lazy.IBk
0:33:20 - lazy.IBk
0:33:26 - lazy.IBk
0:33:31 - lazy.IBk
0:33:37 - lazy.IBk
0:33:41 - lazy.IBk
0:34:10 - lazy.IBk
0:34:17 - lazy.IBk
0:34:30 - lazy.IBk
0:34:42 - lazy.IBk
0:34:51 - lazy.IBk
0:35:23 - lazy.IBk
0:54:54 - bayes.NaiveBayes
0:56:38 - trees.J48
6:24:02 - lazy.IBk
6:28:07 - lazy.IBk
6:31:21 - bayes.NaiveBayes
6:33:02 - trees.J48
8:10:20 - trees.J48
```

Classifier output

```
Size of the tree :      33


Time taken to build model: 0 seconds

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

Correctly Classified Instances        1558               86.5556 %
Incorrectly Classified Instances       242               13.4444 %
Kappa statistic                          0.4348
Mean absolute error                      0.1388
Root mean squared error                  0.2732
Relative absolute error                 71.5028 %
Root relative squared error             87.8132 %
Total Number of Instances             1800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.969    0.590    0.891      0.969    0.928      0.483    0.747     0.909     Normal
                 0.600    0.021    0.643      0.600    0.621      0.598    0.824     0.451     Pathological
                 0.215    0.019    0.583      0.215    0.315      0.312    0.709     0.323     Suspect
Weighted Avg.    0.866    0.495    0.844      0.866    0.844      0.471    0.747     0.818

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1453   24   23 |    a = Normal
   35   63    7 |    b = Pathological
  142   11   42 |    c = Suspect
```

Naive bayes



Classifier: NaiveBayes

Test options: Cross-validation Folds 5

(Nom) fetal_health

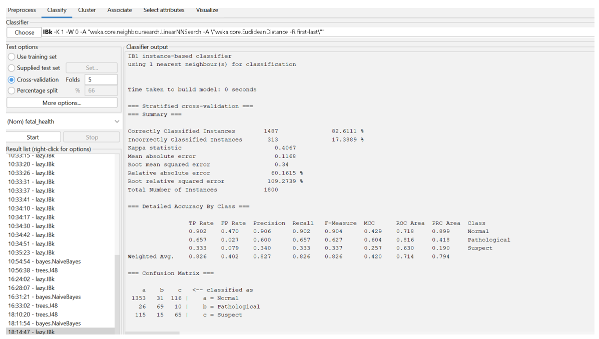
Result list (right-click for options)

```
10:33:10 - lazy.IBk
10:33:15 - lazy.IBk
10:33:20 - lazy.IBk
10:33:26 - lazy.IBk
10:33:31 - lazy.IBk
10:33:37 - lazy.IBk
10:33:41 - lazy.IBk
10:34:10 - lazy.IBk
10:34:17 - lazy.IBk
10:34:30 - lazy.IBk
10:34:42 - lazy.IBk
10:34:51 - lazy.IBk
10:35:23 - lazy.IBk
10:54:54 - bayes.NaiveBayes
10:56:38 - trees.J48
16:24:02 - lazy.IBk
16:28:07 - lazy.IBk
16:31:21 - bayes.NaiveBayes
16:33:02 - trees.J48
18:10:20 - trees.J48
18:11:54 - bayes.NaiveBayes
```

Classifier output

```
Time taken to build model: 0.01 seconds

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

Correctly Classified Instances        1520               84.4444 %
Incorrectly Classified Instances       280               15.5556 %
Kappa statistic                          0.126
Mean absolute error                      0.1628
Root mean squared error                  0.2791
Relative absolute error                 83.8718 %
Root relative squared error             89.6873 %
Total Number of Instances             1800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.920    0.844      0.998    0.915      0.239    0.794     0.929     Normal
                 0.152    0.000    1.000      0.152    0.264      0.380    0.871     0.540     Pathological
                 0.036    0.002    0.636      0.036    0.068      0.133    0.791     0.350     Suspect
Weighted Avg.    0.844    0.767    0.831      0.844    0.785      0.236    0.799     0.844

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1497    0    3 |    a = Normal
   88   16    1 |    b = Pathological
  188    0    7 |    c = Suspect
```

KNN



Classifier: IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options: Cross-validation Folds 5

(Nom) fetal_health

Result list (right-click for options)

```
10:33:15 - lazy.IBk
10:33:20 - lazy.IBk
10:33:26 - lazy.IBk
10:33:31 - lazy.IBk
10:33:37 - lazy.IBk
10:33:41 - lazy.IBk
10:34:10 - lazy.IBk
10:34:17 - lazy.IBk
10:34:30 - lazy.IBk
10:34:42 - lazy.IBk
10:34:51 - lazy.IBk
10:35:23 - lazy.IBk
10:54:54 - bayes.NaiveBayes
10:56:38 - trees.J48
16:24:02 - lazy.IBk
16:28:07 - lazy.IBk
16:31:21 - bayes.NaiveBayes
16:33:02 - trees.J48
18:10:20 - trees.J48
18:11:54 - bayes.NaiveBayes
18:14:47 - lazy.IBk
```

Classifier output

```
IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        1487               82.6111 %
Incorrectly Classified Instances       313               17.3889 %
Kappa statistic                          0.4067
Mean absolute error                      0.1168
Root mean squared error                  0.34
Relative absolute error                 60.1615 %
Root relative squared error            109.2739 %
Total Number of Instances             1800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.902    0.470    0.906      0.902    0.904      0.429    0.718     0.899     Normal
                 0.657    0.027    0.600      0.657    0.627      0.604    0.816     0.418     Pathological
                 0.333    0.079    0.340      0.333    0.337      0.257    0.630     0.190     Suspect
Weighted Avg.    0.826    0.402    0.827      0.826    0.826      0.420    0.714     0.794

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1353   31  116 |    a = Normal
   26   69   10 |    b = Pathological
  115   15   65 |    c = Suspect
```

After extracting the top 2 Principal Components using Weka, I evaluated the performance of the Decision Tree, Naive Bayes, and KNN classifiers. The results are as follows

- **Decision Tree Classifier:**
1. Accuracy: 86.5556%
2. Incorrectly Classified Instances: 13.4444%

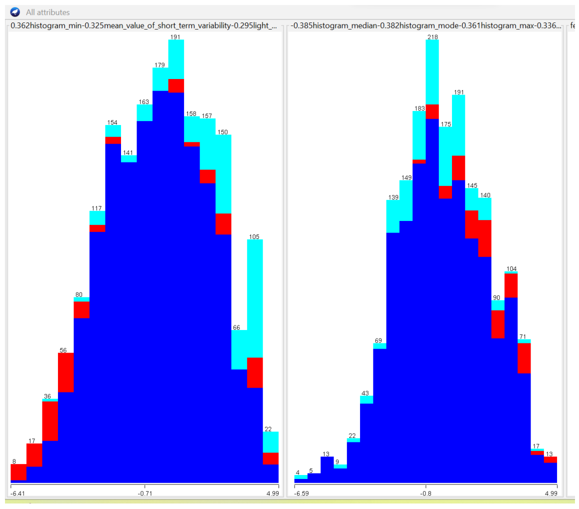
- **Naive Bayes**
1. Accuracy: 84.4444%
2. Incorrectly Classified Instances: 15.5556%
3.

- **KNN:**
1. Accuracy: 82.6111%
2. Incorrectly Classified Instances: 17.3889%

Upon comparing the results with the classifiers' performance on the original dataset, I observed the following:

- The Decision Tree classifier consistently outperformed the other two, followed by Naive Bayes, and then KNN..
- The top 2 Principal Components provide a compact representation of the data but might not capture all the information necessary for classification as accuracy is lower when using the principal components.

Visualize the Principal Components



The histograms showcase the distribution of the data points for each of the top 2 Principal Components. The blue bars represent the majority class while the red and cyan bars appear to represent other classes. A few observations:

1. The first histogram shows a peak around the -0.71 mark, with a significant number of data points aggregated in this region.

2. The second histogram has its peak close to the -0.8 mark. The spread of data points in both histograms suggests a somewhat Gaussian (bell-shaped) distribution, albeit with some skewness.
3. The presence of the red and cyan bars at various points in the histograms indicate that there is some overlap among the classes, which might pose challenges in classification.

Having run various models and feature selections, I expected the principal components to encapsulate the salient features and variances of the original data. The visual comparison between the PCA histograms and the original dataset distributions affirms this to some extent. However, the class overlaps evident in both the PCA and original histograms reiterate the challenges posed by such overlaps in achieving optimal classification results.