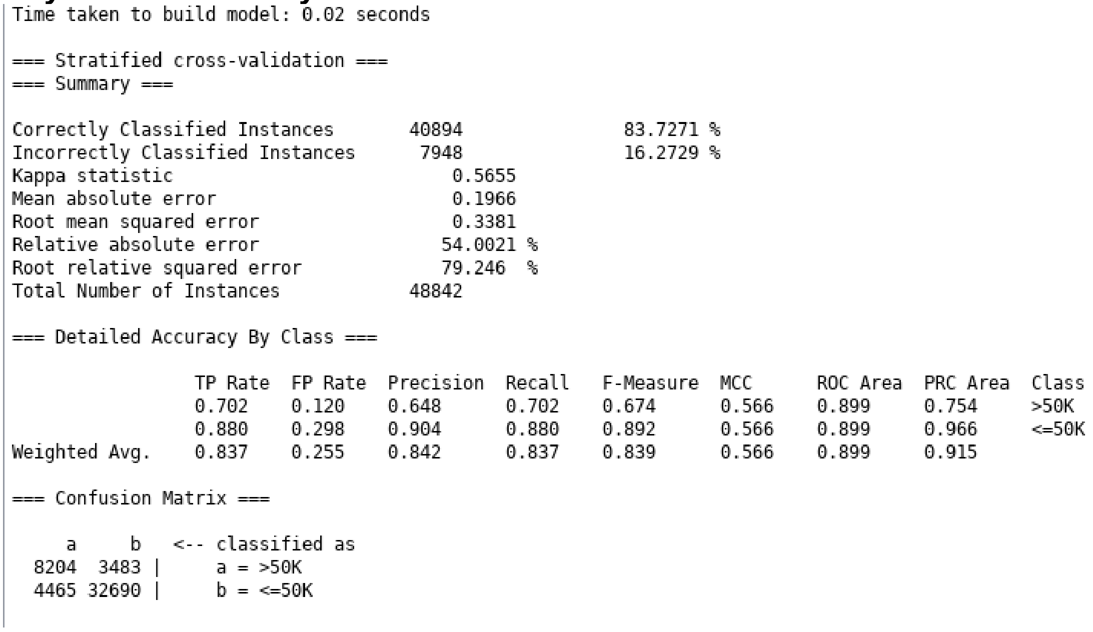
Comp309 assignment 1

Part One:

I am using the Adult dataset for this assignment, in preprocessing for all following classifications, I have removed the fnlwgt, education, and relationship attributes. Education is represented by education-num, and relationship can be derived from gender and marital status.

Baysians: Naive Bayes:



```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances       40894               83.7271 %
Incorrectly Classified Instances      7948               16.2729 %
Kappa statistic                          0.5655
Mean absolute error                      0.1966
Root mean squared error                  0.3381
Relative absolute error                 54.0021 %
Root relative squared error             79.246  %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.702    0.120    0.648      0.702    0.674      0.566    0.899     0.754     >50K
                 0.880    0.298    0.904      0.880    0.892      0.566    0.899     0.966     <=50K
Weighted Avg.    0.837    0.255    0.842      0.837    0.839      0.566    0.899     0.915

=== Confusion Matrix ===

     a     b   <-- classified as
  8204  3483 |     a = >50K
  4465 32690 |     b = <=50K
```

The results of applying a Naive Bayes classifier on the adult dataset show a correct classification accuracy of 82.73%(2.d.p) and an incorrect classification accuracy of 16.27%(2.d.p). However, some other statistics provided show that this method of classification has an easier time classifying in the <=50K range as shown by the precision of 0.904 compared to 0.648 for classifying >50k.

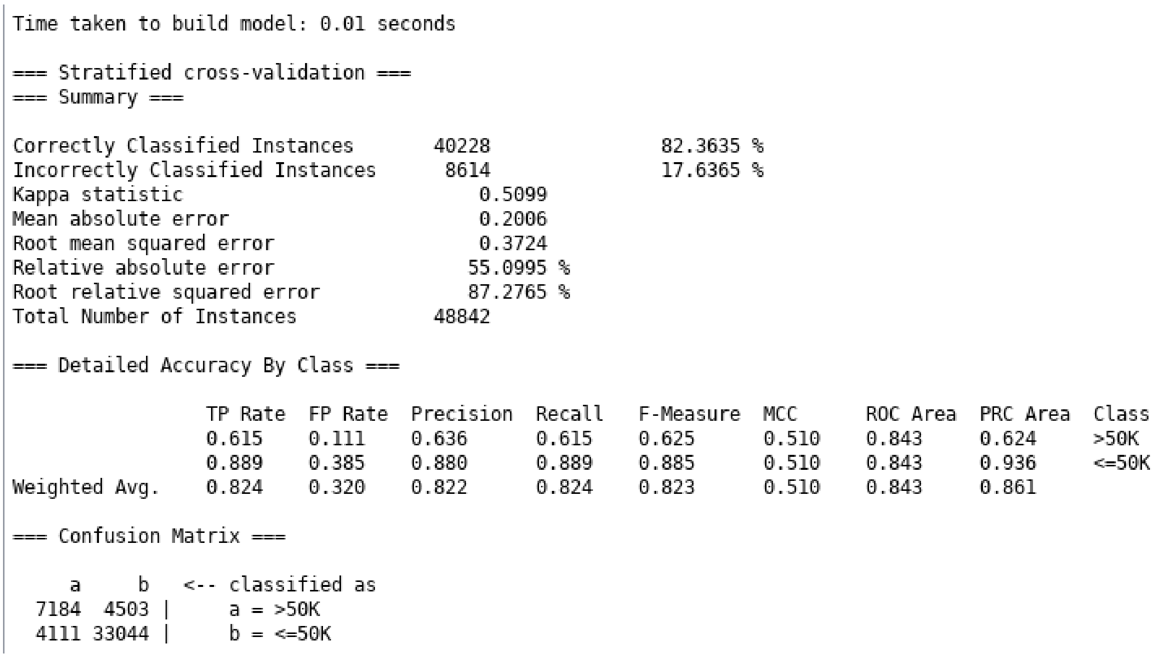
The Naive Bayes method is a quick classifier to create the model for, as it simply needs to check the counts of each attribute against their given class, and it uses these as probabilities to classify unknown data. An appropriate way to represent the results given by the Naive Bayes method could be given in the way of classification rules, basically an equation provided in which all information about a single person could be inputted and a classification would be the outcome.

The Naive Bayes classifier belongs to the Bayesian tribe of AI because of its use of probability-based decision making in the face of uncertainty, using all the available data to create a probabilistic model of that data for use in classification. As the Bayesian tribe focus on the handling of uncertainty and hypothesising based on data they have available.

Naive Bayes is a simple implementation of a Bayesian network, using tables of conditional probabilities of represented attributes at each node of (what is usually represented as) a graphical model. This method of hypothesising is representative of the Bayesian tribe.

The evaluation idea used by the Bayesian tribe, and consequently Naive Bayes, is known as posterior probability. Posterior probability is conditional probability that is assigned after the relevant evidence is taken into account. Naive Bayes uses this in the evaluation of every individual outcome as they are all assumed to be conditionally independent.

Analogizers: IBk (kNN)



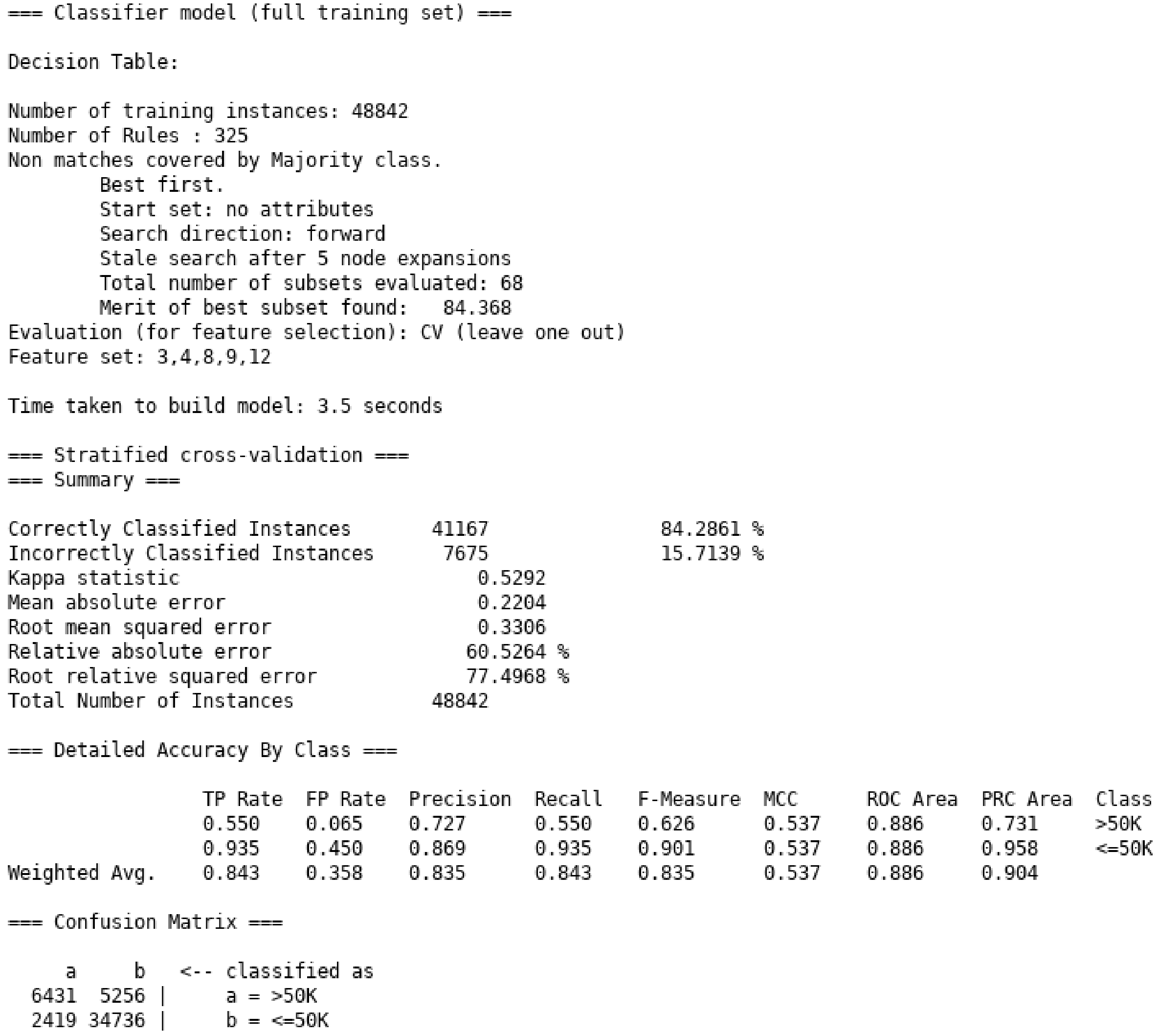
```
Time taken to build model: 0.01 seconds

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

Correctly Classified Instances       40228               82.3635 %
Incorrectly Classified Instances      8614               17.6365 %
Kappa statistic                          0.5099
Mean absolute error                      0.2006
Root mean squared error                  0.3724
Relative absolute error                 55.0995 %
Root relative squared error             87.2765 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.615    0.111    0.636      0.615    0.625      0.510    0.843     0.624     >50K
                 0.889    0.385    0.880      0.889    0.885      0.510    0.843     0.936     <=50K
Weighted Avg.    0.824    0.320    0.822      0.824    0.823      0.510    0.843     0.861

=== Confusion Matrix ===

     a     b   <-- classified as
  7184  4503 |     a = >50K
  4111 33044 |     b = <=50K
```

The results of applying a IBk, an implementation of the K nearest neighbour algorithm, on the adult dataset shows a correct classification accuracy of 82.36%(2.d.p) and an incorrect classification accuracy of 17.64%(2.d.p). Again we have a distinction between the precision of accuracy given between classifying the two classes given, with <=50k being over 0.2 higher in precision than >50k.

The Analogizers tribe of AI attempt to match data by similarities, searching for analogues of current data with known patterns or data, this is the principle behind nearest neighbour algorithms. K nearest neighbour searches for the k closest matching data points and classifies the current using Euclidean distances to the nearest instances. The resulting information can then be represented by looking at decision boundaries on plot of the data points.

The lazy nature of the k nearest neighbour algorithm, given the only classification measure is the distance measure, it should only gain more accurate analysis given more data points.

Symbolists: Decision Table

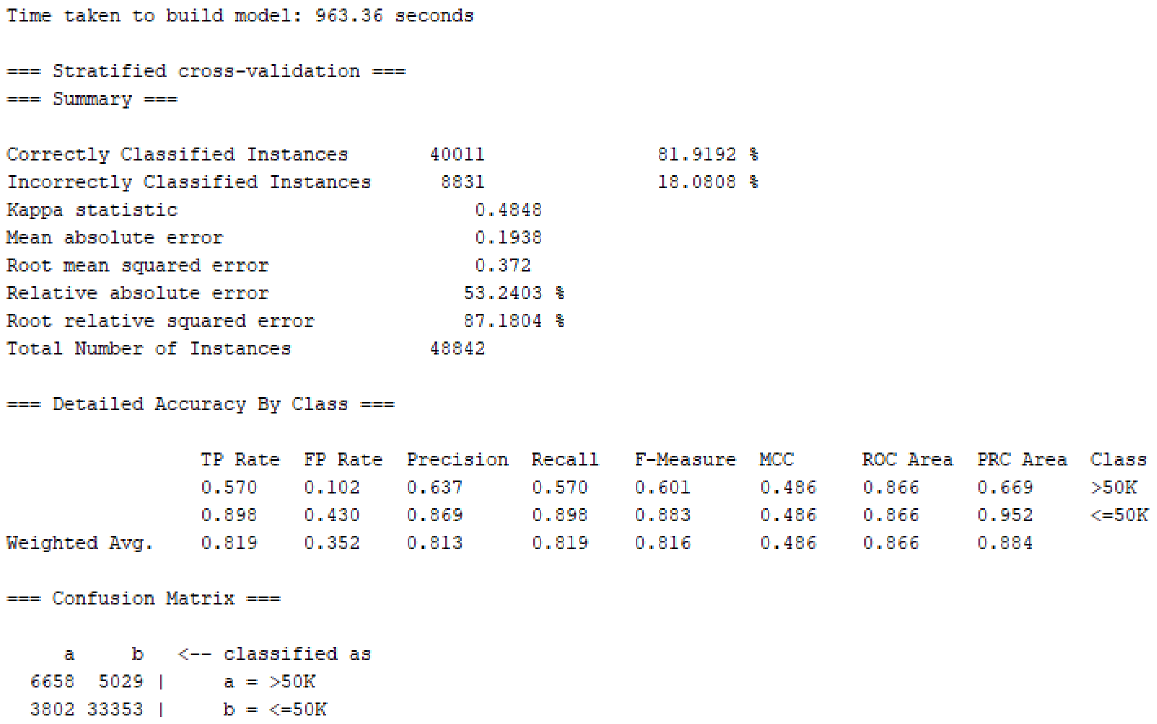


```
=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 48842
Number of Rules : 325
Non matches covered by Majority class.
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 68
        Merit of best subset found:   84.368
Evaluation (for feature selection): CV (leave one out)
Feature set: 3,4,8,9,12

Time taken to build model: 3.5 seconds

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

Correctly Classified Instances       41167               84.2861 %
Incorrectly Classified Instances      7675               15.7139 %
Kappa statistic                          0.5292
Mean absolute error                      0.2204
Root mean squared error                  0.3306
Relative absolute error                 60.5264 %
Root relative squared error             77.4968 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.550    0.065    0.727      0.550    0.626      0.537    0.886     0.731     >50K
                 0.935    0.450    0.869      0.935    0.901      0.537    0.886     0.958     <=50K
Weighted Avg.    0.843    0.358    0.835      0.843    0.835      0.537    0.886     0.904

=== Confusion Matrix ===

     a     b   <-- classified as
  6431  5256 |     a = >50K
  2419 34736 |     b = <=50K
```

With the decision table, we have a correct classification accuracy of 84.29%(2.d.p), and an incorrect classification accuracy of 15.71%(2.d.p), given the symbolists evaluation method being centred around accuracy, these seem to be the most important statistics for it.

The Decision table method used is based on a decision tree, and thus fits within the tribe of the symbolists, as they look at filling gaps within existing knowledge. The decision tree uses the given data and the conclusion for what the outcomes are, and works to create branches which lead to a given outcome. This is a sort of Inverse deduction, in which the outcome is known and the solution is a result of working backwards through pattern recognition, which builds into a hypothesis, and finally results in a theory, this is applied by Symbolist algorithms by using the results in a form of self-improvement. Applying inference from a set of results to what is known to optimize the gaps in knowledge already provided.

Connectionists: Multilayer Perceptron



```
Time taken to build model: 963.36 seconds

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

Correctly Classified Instances       40011               81.9192 %
Incorrectly Classified Instances      8831               18.0808 %
Kappa statistic                          0.4848
Mean absolute error                      0.1938
Root mean squared error                  0.372
Relative absolute error                 53.2403 %
Root relative squared error             87.1804 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.570    0.102    0.637      0.570    0.601      0.486    0.866     0.669     >50K
                 0.898    0.430    0.869      0.898    0.883      0.486    0.866     0.952     <=50K
Weighted Avg.    0.819    0.352    0.813      0.819    0.816      0.486    0.866     0.884

=== Confusion Matrix ===

     a     b   <-- classified as
  6658  5029 |     a = >50K
  3802 33353 |     b = <=50K
```

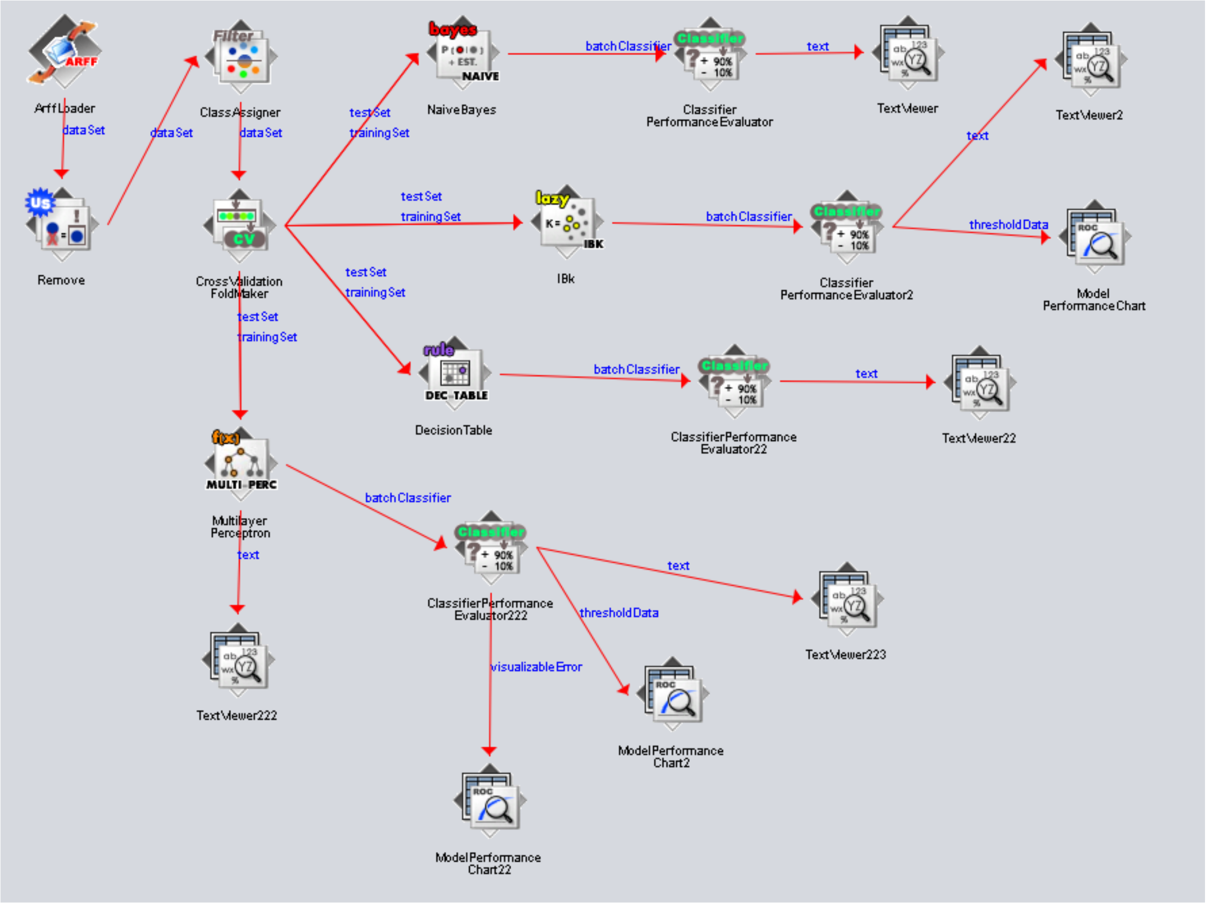
Using a multilayer perceptron for classification results in correct classification accuracy of 81.92% (2.d.p) and an incorrect classification accuracy of 18.08% (2.d.p), and as a part of the connectionists, the mean absolute error is an important piece of information which tell us the average distance the models predictions are from the actual data points, in this case 0.1938.

The multilayer perceptron belongs to the connectionist tribe of AI as it is a form of neural network, in its basest sense it is the goal of the connectionists to reverse engineer the brain like this. The multilayer perceptron is a feed forward neural network, usually represented by a directed graph, in which several layers of input nodes are used to produce outputs directed towards the output nodes.

The many attributes within the adult dataset result in the creation of 42 sigmoid nodes, in which the sigmoid function is evaluated on the input data within the hidden layers of the perceptron.

The multilayer perceptron uses a backpropagation algorithm to self-evaluate this algorithm looks at the resulting squared error statistics to judge how well it has performed, and adjust accordingly using a gradient descent algorithm to adjust the weights of the input measures.

Part Two: Data Pipeline



Business Understanding:
This dataset is a summary of over 48000 Americans and classifies them on if they make more or less than $50000 per year and has applications in targeted marketing looking for a demographic of middling to higher level salaries/incomes. With the information gathered one would be looking to be able to classify any persons outside the dataset with information gathered elsewhere to see if they could be a candidate for any sort of marketing strategy using the given demographic.

Data understanding:
This dataset has over 48000 records, labelled binomially by salaries of >50K or <=50K. With a majority of the records belonging to the <=50K group (76%). The dataset is made of 14 different attributes in which I have used 11 within my classifications, consisting of 5 continuous attributes, 1 binomial attribute, and 5 polynomial attributes.
The continuous attributes are: age, education-num(a numerical representation of the removed polynomial attribute education, showing highest earned education level), capital-gain, capital-loss and hours worked per week.
The nominal attributes are: workclass (the type of employment class, such as self-employed, federal, never worked etc), marital-status, occupation, race and sex.

Data Preparation:

The pipeline should allow one to assess the importance of some of the datapoints and choose a classifier which can accurately classify datapoints as the information may be reduced, and some attributes modified if necessary. The algorithms selected all come from separate tribes of AI and all are robust enough to deal with the various types of attribute data supplied in the dataset.

Modelling:

This pipeline suits the 4 stated tribes of AI in this assignment, Bayesians, Symbolists, Connectionists, and Analogizers.

Evaluation:

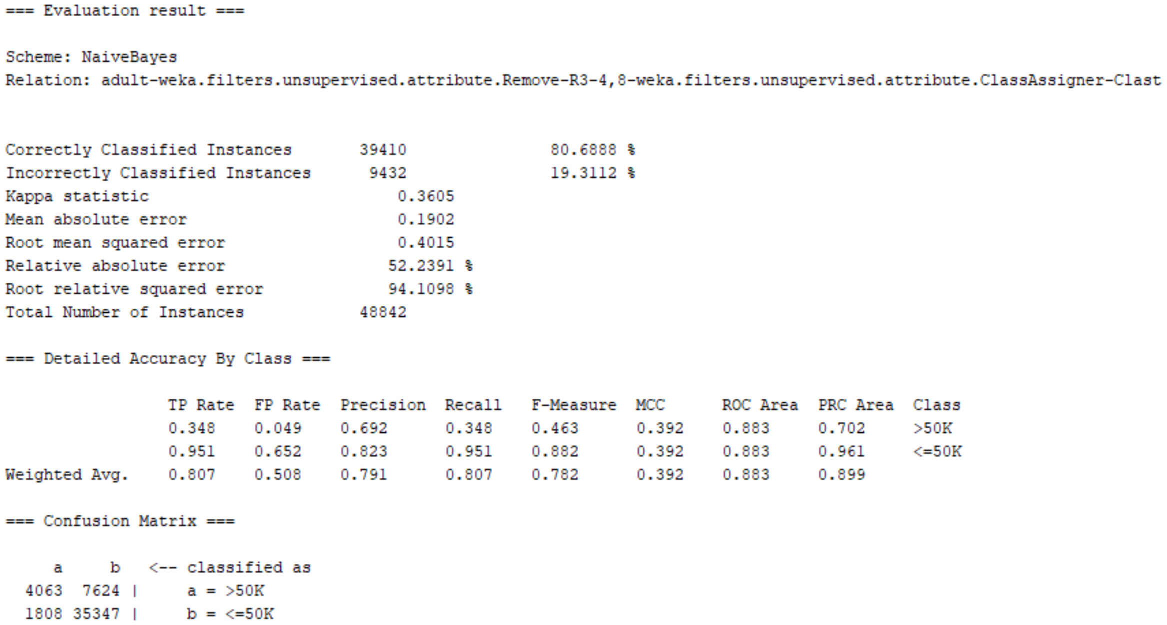
Within each algorithm chosen it is possible to evaluate a solution from each one.

Deployment:

The model produced should be able to be deployed easily enough, some tinkering with attributes may be required for better accuracy.

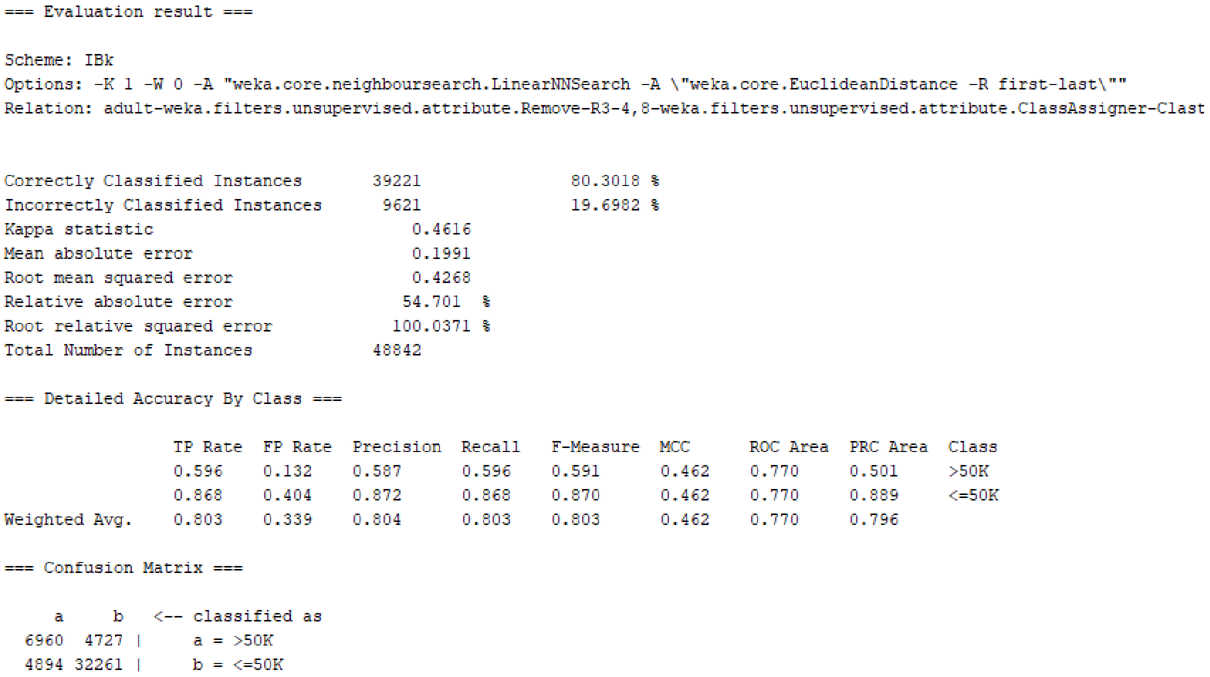
Part 2.3: Pipeline vs Explorer.

Naïve Bayes:



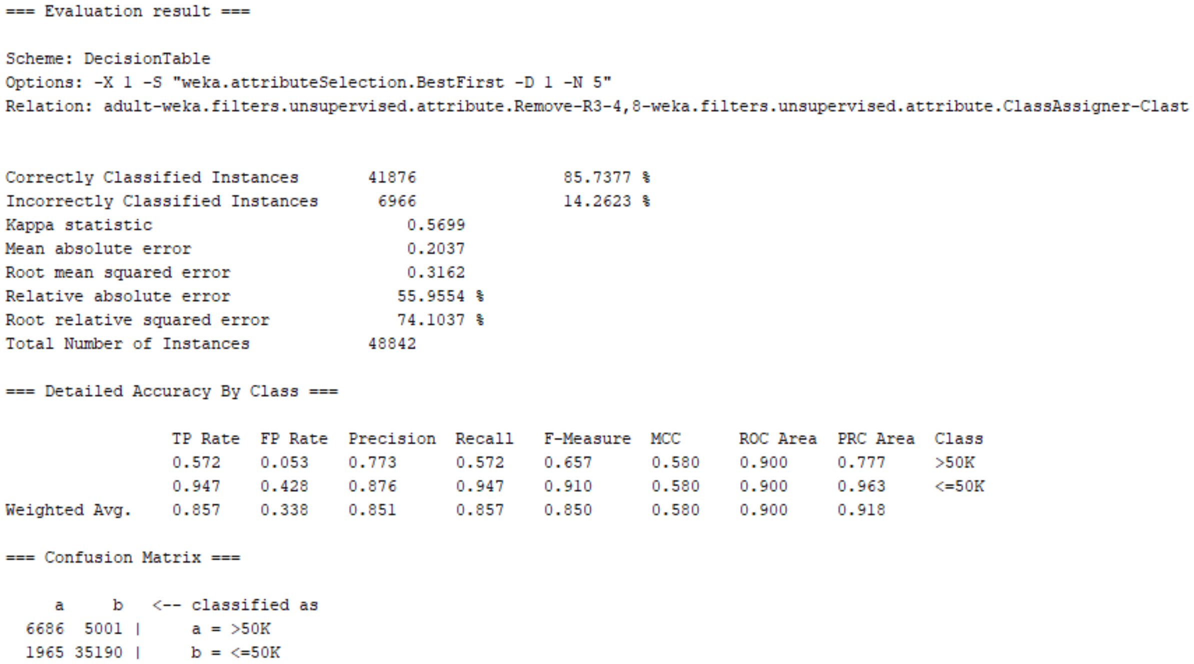
```
=== Evaluation result ===

Scheme: NaiveBayes
Relation: adult-weka.filters.unsupervised.attribute.Remove-R3-4,8-weka.filters.unsupervised.attribute.ClassAssigner-Clast


Correctly Classified Instances       39410               80.6888 %
Incorrectly Classified Instances      9432               19.3112 %
Kappa statistic                          0.3605
Mean absolute error                      0.1902
Root mean squared error                  0.4015
Relative absolute error                 52.2391 %
Root relative squared error             94.1098 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.348    0.049    0.692      0.348    0.463      0.392    0.883     0.702     >50K
                 0.951    0.652    0.823      0.951    0.882      0.392    0.883     0.961     <=50K
Weighted Avg.    0.807    0.508    0.791      0.807    0.782      0.392    0.883     0.899

=== Confusion Matrix ===

     a     b   <-- classified as
  4063  7624 |     a = >50K
  1808 35347 |     b = <=50K
```

kNN (IBk):



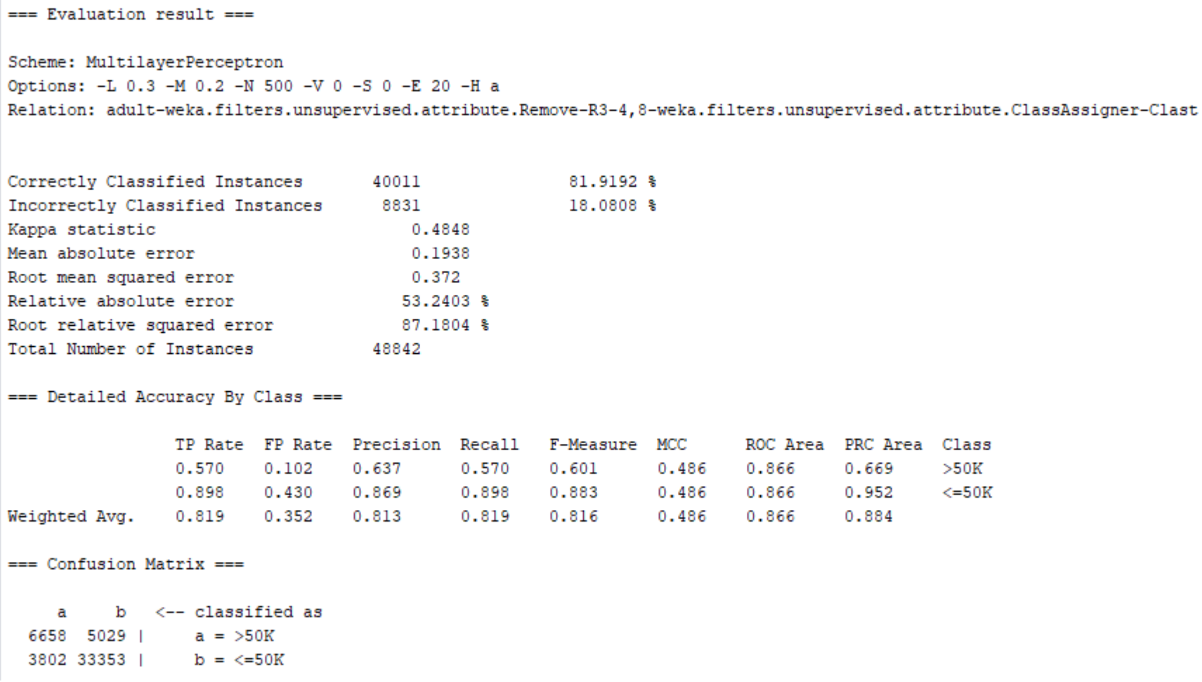
```
=== Evaluation result ===

Scheme: IBk
Options: -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation: adult-weka.filters.unsupervised.attribute.Remove-R3-4,8-weka.filters.unsupervised.attribute.ClassAssigner-Clast


Correctly Classified Instances       39221               80.3018 %
Incorrectly Classified Instances      9621               19.6982 %
Kappa statistic                          0.4616
Mean absolute error                      0.1991
Root mean squared error                  0.4268
Relative absolute error                 54.701  %
Root relative squared error            100.0371 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.596    0.132    0.587      0.596    0.591      0.462    0.770     0.501     >50K
                 0.868    0.404    0.872      0.868    0.870      0.462    0.770     0.889     <=50K
Weighted Avg.    0.803    0.339    0.804      0.803    0.803      0.462    0.770     0.796

=== Confusion Matrix ===

     a     b   <-- classified as
  6960  4727 |     a = >50K
  4894 32261 |     b = <=50K
```

Decision Table:



```
=== Evaluation result ===

Scheme: DecisionTable
Options: -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Relation: adult-weka.filters.unsupervised.attribute.Remove-R3-4,8-weka.filters.unsupervised.attribute.ClassAssigner-Clast


Correctly Classified Instances       41876               85.7377 %
Incorrectly Classified Instances      6966               14.2623 %
Kappa statistic                          0.5699
Mean absolute error                      0.2037
Root mean squared error                  0.3162
Relative absolute error                 55.9554 %
Root relative squared error             74.1037 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.572    0.053    0.773      0.572    0.657      0.580    0.900     0.777     >50K
                 0.947    0.428    0.876      0.947    0.910      0.580    0.900     0.963     <=50K
Weighted Avg.    0.857    0.338    0.851      0.857    0.850      0.580    0.900     0.918

=== Confusion Matrix ===

     a     b   <-- classified as
  6686  5001 |     a = >50K
  1965 35190 |     b = <=50K
```

Multilayer Perceptron:



```
=== Evaluation result ===

Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: adult-weka.filters.unsupervised.attribute.Remove-R3-4,8-weka.filters.unsupervised.attribute.ClassAssigner-Clast


Correctly Classified Instances       40011               81.9192 %
Incorrectly Classified Instances      8831               18.0808 %
Kappa statistic                          0.4848
Mean absolute error                      0.1938
Root mean squared error                  0.372
Relative absolute error                 53.2403 %
Root relative squared error             87.1804 %
Total Number of Instances            48842

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.570    0.102    0.637      0.570    0.601      0.486    0.866     0.669     >50K
                 0.898    0.430    0.869      0.898    0.883      0.486    0.866     0.952     <=50K
Weighted Avg.    0.819    0.352    0.813      0.819    0.816      0.486    0.866     0.884

=== Confusion Matrix ===

     a     b   <-- classified as
  6658  5029 |     a = >50K
  3802 33353 |     b = <=50K
```

There is little to no difference in the results based on the data pipeline approach as I rebuilt my methods for testing in the explorer as my pipeline.