Ordinary Degree in Computing: Data Mining

David Kelly B00060572



Data Mining Assessment:

Mine a dataset

10/12/2014

Contents

**1.0 Business Understanding** ……………………………………………………………............ 3

1.1 Background …………………………………………………………………… 3

1.1.1 Goal …………………………………………………………………… 3

1.1.2 Project Plan …………………………………………………………… 3

**2.0 Data Understanding** ……………………………………………………………............ 4

2.1 Initial Data …………………………………………………………………… 4

2.2 Data Description …………………………………………………………………… 5

2.3 Data Exploration …………………………………………………………………… 5

2.3.1 Correlated Data …………………………………………………………… 6

2.3.2 Attribute Data …………………………………………………………… 6

2.3.3 Skewed Data …………………………………………………………… 6

2.3.4 Predictive Attributes …………………………………………………… 7

2.4 Data Quality …………………………………………………………………… 7

**3.0 Data Preparation**  ……………………………………………………………............ 8

3.1 Select Data …………………………………………………………………… 8

3.2 Cleaning Data …………………………………………………………………… 9

3.3 Constructing Data …………………………………………………………………… 9

3.4 Integrating Data …………………………………………………………………… 10

3.5 Formatting Data …………………………………………………………………… 12

**4.0 Modeling**  ……………………………………………………………............ 13

4.1 Select Modeling Techniques …………………………………………………… 13

4.2 Neural Network Model …………………………………………………………… 13

4.3 Naïve Bayes Model …………………………………………………………… 14

**5.0 Evaluation** …………………………………………………………………… 15

5.1 Results …………………………………………………………………… 15

5.1.1 Decision Tree …………………………………………………………… 15

5.1.2 k-NN …………………………………………………………… 15

5.1.3 Naïve Bayes …………………………………………………………… 16

5.1.4 Neural Network …………………………………………………………… 16

5.2 Review …………………………………………………………………… 16

1.0 Business understanding

1.1 Background

This report is related to the data collected during the 2013 Australian Tennis Open. The columns within the data set represent match attributes such as, result, player names, aces and other match data. The objective of this report is to document the CRISP-DM phases. The project goal is to create a model that will mine the data set and accurately predict the winner of each game. The initial data set has many errors and will need to be cleaned of any uninformative data. The model created will aim to have a prediction rate of over 90% accuracy. The final challenge in creating this model is to not over fit the data, causing the model to be constricted to just this data set. Ideally the finished model will be accurate and flexible enough to be implemented on projects with the same input data.

1.1.1 Goal

* Create a model with over 90% accuracy at predicting match winners
* Create a flexible model with out over fitting the data
* Create the model following the CRISP-DM phases

1.1.2 Project plan (CRISP-DM phases)

|  |  |
| --- | --- |
| **Data Understanding** | Collect Initial Data |
|  | Describe Data |
|  | Explore Data |
|  | Verify Data Quality |
| **Data Preparation** | Select Data |
|  | Clean Data |
|  | Construct Data |
|  | Integrate Data |
|  | Format Data |
| **Modeling** | Select Modeling Techniques |
|  | Generate Test Data |
|  | Build Model |
|  | Assess Model |
| **Evaluation** | Evaluate Results |
|  | Review Process |
|  | Determine Next Steps |
| **Deployment** | Plan Deployment |
|  | Monitor & Maintain |
|  | Review Project |

2.0 Data understanding

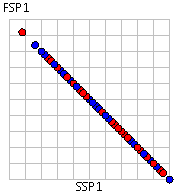
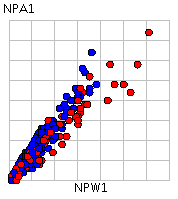
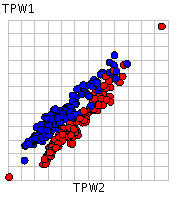
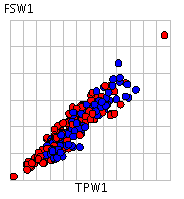
2.1 Initial Data

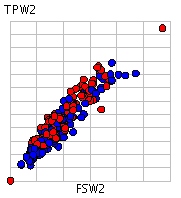
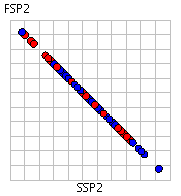
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **Type** | **Mean** | **Min** | **Max** | **Deviation** | **Mode** |
|  |  |  |  |  |  |  |
| Result | Binomial | 0.5 | 0 | 1 | NA | 1(134) |
| Player 1 | Polynomial | NA | NA | NA | NA | R.Nadal(7) |
| Player 2 | Polynomial | NA | NA | NA | NA | R.Federer(6) |
| Round | Integer | 1.192 | 1.000 | 15.000 | +/- 1.511 | NA |
| FSP1 | Integer | 61.768 | 40 | 86 | =/- 8.047 | NA |
| FSW1 | Integer | 38.236 | 3 | 109 | +/- 16.978 | NA |
| SSP1 | Integer | 38.232 | 14 | 60 | +/- 8.047 | NA |
| SSW1 | Integer | 16.744 | 1 | 47 | +/-8.682 | NA |
| ACE1 | Integer | 6.948 | 0 | 41 | +/- 6.849 | NA |
| DBF1 | Integer | 4.210 | 0 | 50 | +/-4.142 | NA |
| WNR1 | Integer | 25.228 | 0 | 111 | +/-17.622 | NA |
| UFE1 | Integer | 28.295 | 0 | 81 | +/-17.247 | NA |
| BPC1 | Integer | 3.984 | 0 | 45 | +/-3.438 | NA |
| BPW1 | Integer | 8.933 | 0 | 28 | +/-5.224 | NA |
| NPA1 | Integer | 10.958 | 0 | 37 | +/-7.095 | NA |
| NPW1 | Integer | 16.282 | 1 | 61 | +/-11.206 | NA |
| TPW1 | Integer | 91.051 | 5 | 231 | +/-35.781 | NA |
| ST11 | Integer | 4.854 | 0 | 7 | +/-1.923 | NA |
| ST21 | Polynomial | NA | NA | NA | NA | 6(128) |
| ST31 | Polynomial | NA | NA | NA | NA | NA(89) |
| ST41 | Polynomial | NA | NA | NA | NA | NA(193) |
| ST51 | Polynomial | NA | NA | NA | NA | NA(231) |
| FSP2 | Integer | 61.332 | 39 | 86 | +/-7.859 | NA |
| FSW2 | Integer | 38.177 | 0 | 114 | +/-17.445 | NA |
| SSP2 | Integer | 38.909 | 14 | 100 | +/-8.737 | NA |
| SSW2 | Integer | 16.768 | 1 | 57 | +/-8.915 | NA |
| ACE2 | Integer | 5.916 | 0 | 32 | +/-5.722 | NA |
| DBF2 | Integer | 4.433 | 0 | 18 | +/-3.137 | NA |
| WNR2 | Integer | 25.094 | 0 | 82 | +/-16.410 | NA |
| UFE2 | Integer | 29.886 | 0 | 96 | +/-18.700 | NA |
| BPC2 | Integer | 3.585 | 0 | 10 | +/-2.340 | NA |
| BPW2 | Integer | 8.324 | 0 | 22 | +/-4.864 | NA |
| NPA2 | Integer | 11.809 | 0 | 49 | +/-8.832 | NA |
| NPW2 | Integer | 17.517 | 0 | 66 | +/-12.747 | NA |
| TPW2 | Integer | 90 | 1 | 230 | +/-36.305 | NA |
| ST12 | Integer | 4.771 | 0 | 7 | +/-1.957 | NA |
| ST22 | Polynomial | NA | NA | NA | NA | 6(99) |
| ST32 | Polynomial | NA | NA | NA | NA | NA(89) |
| ST42 | Polynomial | NA | NA | NA | NA | NA(193) |
| ST52 | Polynomial | NA | NA | NA | NA | NA(231) |
| Gender | Binomial | NA | NA | NA | NA | Male(127) |

2.2 Data Description

* 254 Rows of data
* 1 Special attribute
* 40 Regular attributes
* Class label = “Result” data type = “Binominal”
* “Round” attribute has one value outside 1-7. Presumed incorrect value
* Many of the “ST” set result data types are incorrect
* Many of the “NP” set results for both player 1 and player 2 are missing 45 data
* The “Results” and “Round” attributes are discrete values “0-1” and “1-7” respectively
* The “Gender” attribute is type text and therefore categorical
* The “Player1” and “Player2” attributes represent the name of each player and therefore nominal

2.3 Data Exploration



2.3.1 Correlated Data

Reading the data through plot view and a scatter matrix enabled me to see correlating attributes in the data set. This information allows me to remove an attribute from each correlation as they both indicate the same data. The most correlated attributes are.

* FSP1 - SSP1
* FSW1 - TPW1
* NPA1 - NPW1
* TPW1 - TPW2
* FSP2 - SSP2
* TPW2 - FSW2

2.3.2 Attribute Data

* Round must contain an incorrect value as the values range from 1-7 with one value being 15. Further the average is 1.992 with a deviation of 1.511
* FSW1 ranges from 3 - 109
* WNR1 ranges from 0 - 111
* UFE1 range from 0 - 89
* TPW1 and TPW2 add no additional information
* ST21, ST31, ST41, ST51 all have incorrect data types
* FSW2 ranges from 0 - 114
* SSP2 ranges fro 14 - 100
* WNR2 ranges from 0 - 82
* UFE2 ranges from 0 - 96
* ST22, ST32, ST42, ST52 all have incorrect data types

2.3.3 Skewed Data

Using the histogram plot to discover skewed data with possible outliers,

the following attributes show a positive skew: Round, FSW1, SSW1, DBF1, WNR1, UFE1,

BPC1, BPW1, NPA1, NPW1, SSP2, SSW2,

BPW2, NPA2, NPW2

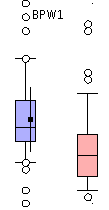
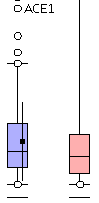
The following attributes show negative skew: FSP1, FSW2, UFE2, WNR2

Finally the following attributes have no skew: SSP1

2.3.4 Predictive Attributes

Trying to use the initial data set with no cleaning to find attributes that might indicate a correlation with winning proved difficult. A lot of the data is still very noisy and doesn’t add any insight in to the outcome of a winner. Having said that there are a few attributes that do hint at a relationship with winners.

Using the color matrix quartile with the color set to result shows lots of uninformative plots, however the ACE1 attribute shows that a greater number of wins are achieved by players that scored more aces in their games. Further players that won a point with their first serve also seem to have a greater connection to winning game. Of course the players that have gained most points have a connection to winning games but this is too direct of a connection. Players that won more break points have a greater connection to winning games too.

2.4 Data Quality

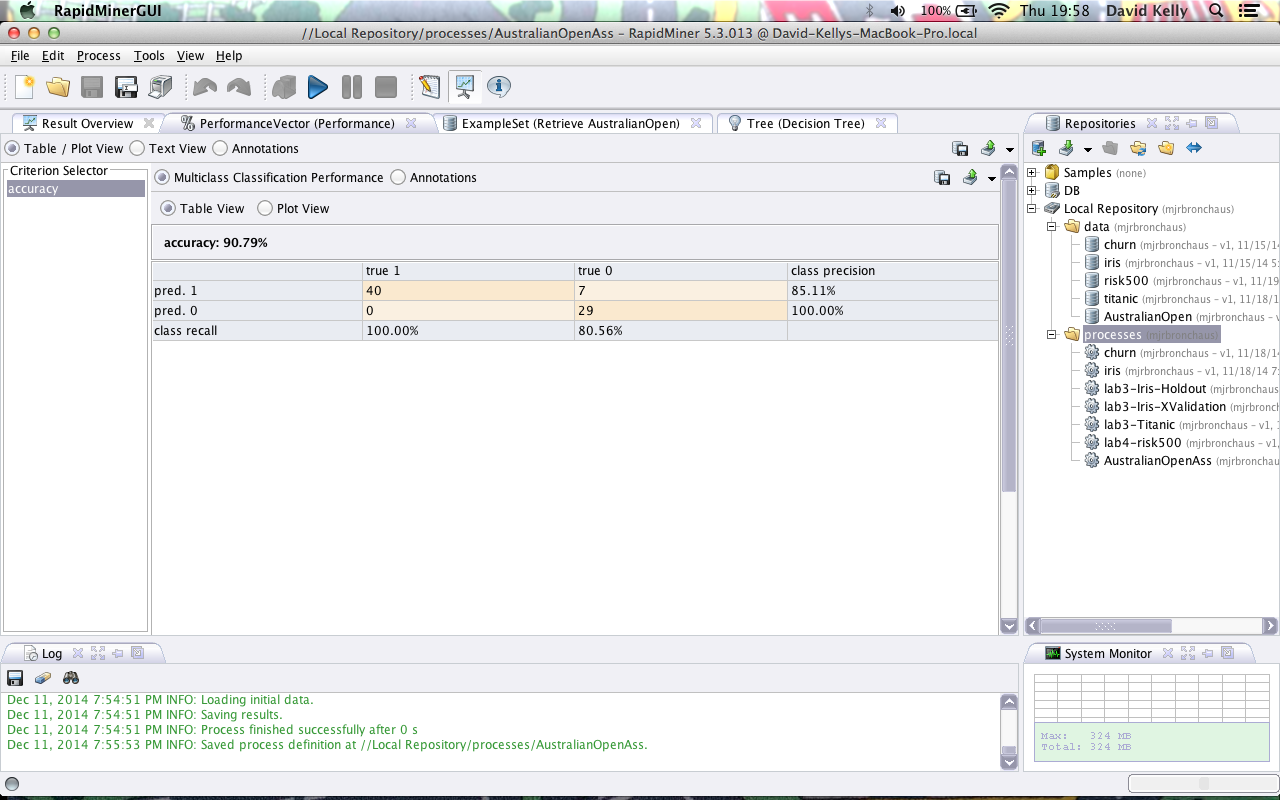
The initial data set has a lot of irregular data, errors and outliers, which would lead to poor results if not amended. There are 52 rows of data that have missing attributes. This is quite a large amount of rows, nearly 20%. 45 NPW and NPA values are missing from the data set, again nearly 20%. Using the histogram plot to confirm the output of the detect outlier operation, Nearly half the attributes have outlier data. This must be cleaned up in order to achieve an accurate model. As stated earlier in the report nearly all the set results have incorrect data types and must be corrected in the data preparation phase. The data set is also too small and should have at least 20 x attributes in rows. The initial data has 254 rows but the correct calculation for rows shows that number should be closer to 800 rows.

* The Data set has a lot of missing values that may lead to removal of attribute or bootstrapping
* There are lots of outliers in the data set that need removal
* There are numerous instances of irregular data in the data set, These values must be fixed
* There are 254 rows in the data set ideally that number should be closer to 800 rows. 20 x 40

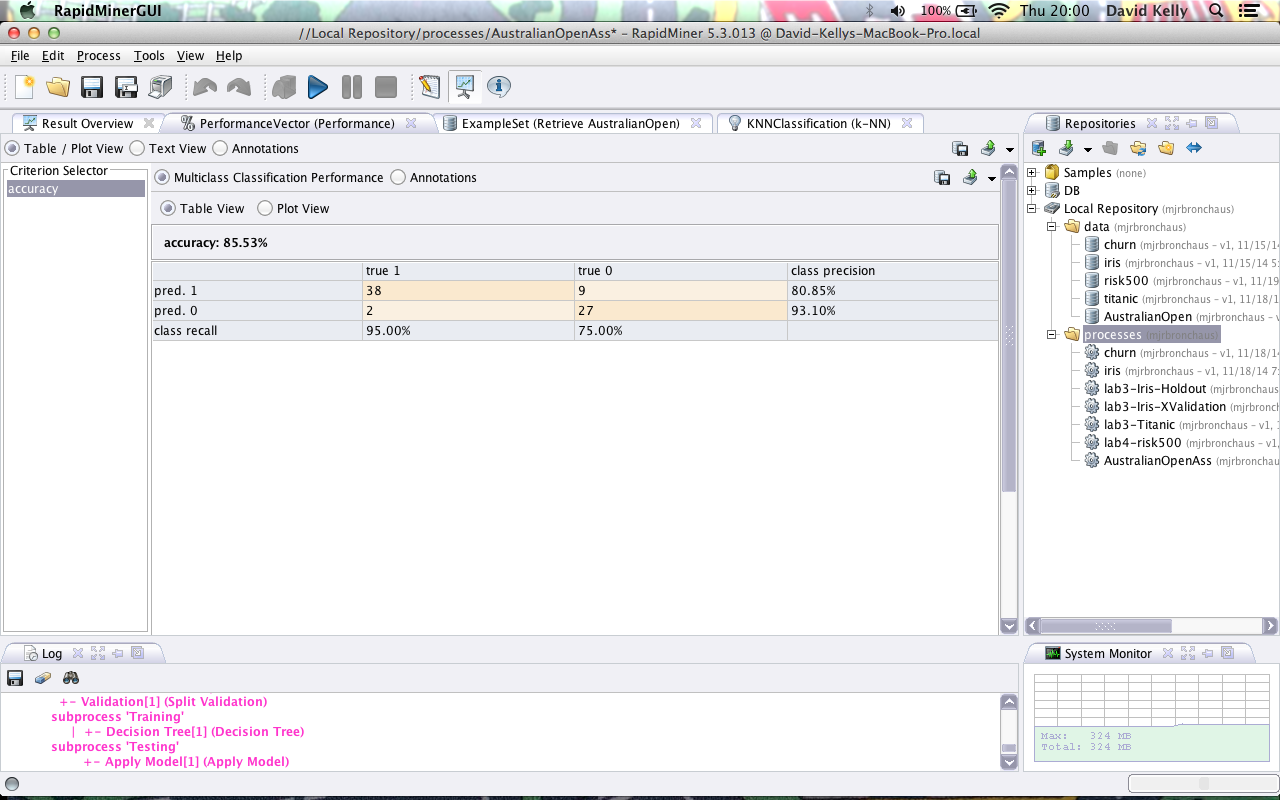
3.0 Data Preparation

3.1 Select Data

Before selecting the data to remove and clean I first decided to run a Performance vector with both a binary tree and k-NN to confirm that the data I had established as being irregular and uninformative was indeed accurate. The binary tree plot visually shows what attributes do not influence the prediction of match winners.



I was surprised by the accuracy results returned after using split validation with a split ratio value of 0.7. I set the sampling type to stratified and input the binary tree in the training field. The training data operated on 76 rows and had an accuracy of 90.79%. I then decided to try the k-NN operator instead of the binary tree.

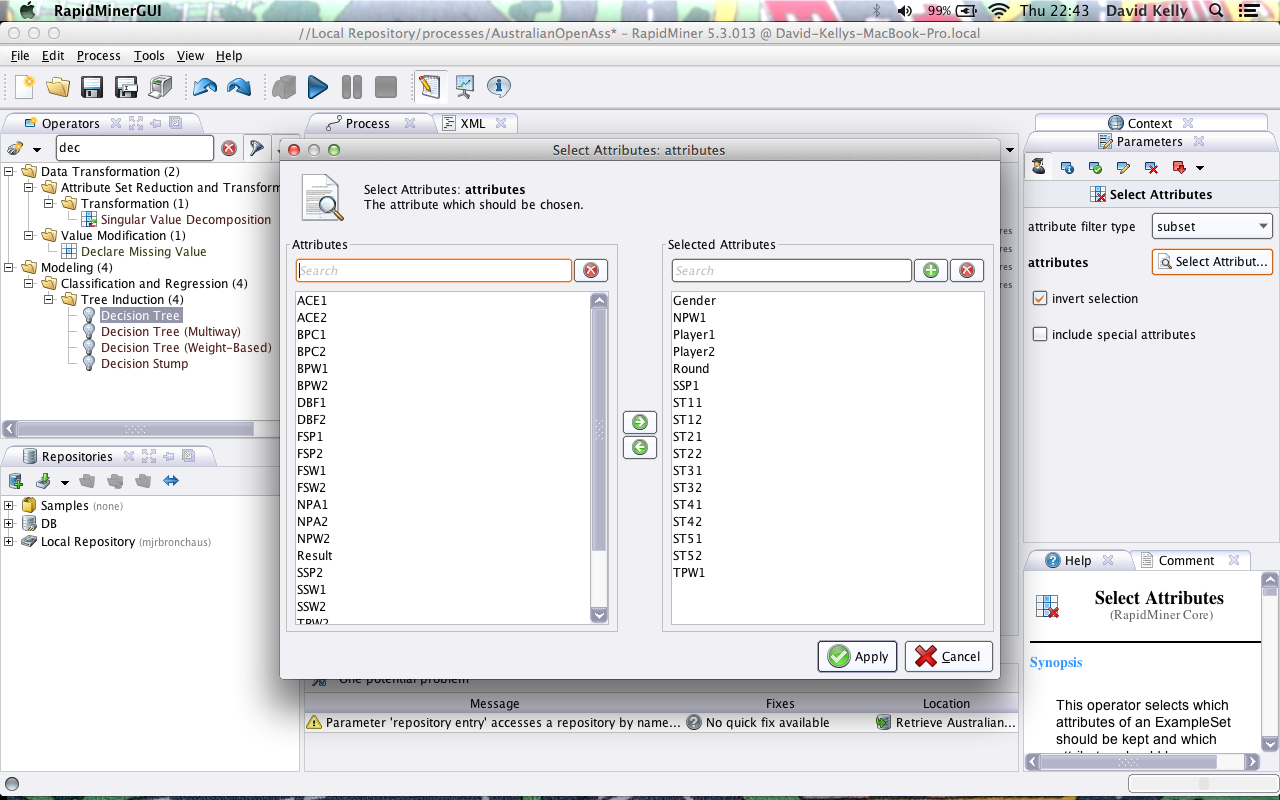


Running the process with the initial value for K set to 1 returned an accuracy of 85.53%. With K set to 1 within the training set and the validation operator split ratio set to 0.7.

3.2 Cleaning Data

I began cleaning the data by removing attributes that I felt had no overall bearing on the results of the process. Using the select attributes operator and I selected attributes from the exploratory analysis phase that had skewed data and irrelevant information.

First of all I removed attributes that were correlated. The correlation was discovered in the exploratory phase earlier. SSP1, TPW1, NPW1 could all go as they are simply duplicate information. I then removed all the set results ST1, ST2, ST3, ST4, ST5 as they had nearly 20% missing data. 20% missing data is too much to keep the column. The Accuracy went from 89.4% up to 90.17% using the decision tree.



After removing the columns missing too much data to be replaced, I decided to also remove attributes that had not influence on the process results such as Player1, Player2, Round and Gender. After these changes the accuracy was at 87%.

I then began the task of replacing all the missing values on the rows remaining. There are 52 rows with missing data remaining in the set. To replace the data I chose the replace missing data operator and connected it to the select attribute operator. I set the default replacement data to average; this gets the mean of the attribute and inputs it in to the missing fields. After doing this to the data the accuracy of the process went up 2% with the decision tree. The accuracy is now 89%.

3.3 Constructing Data

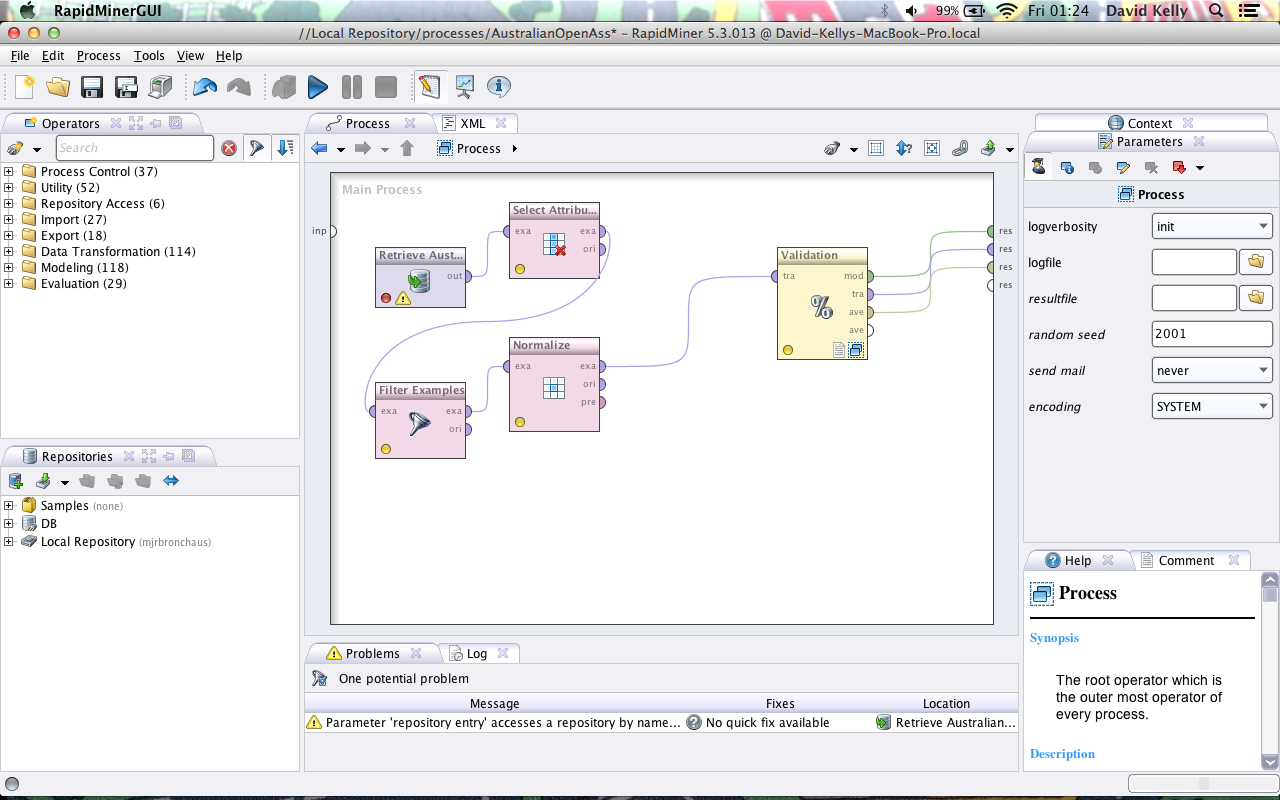
The data set still has 254 rows but all rows now have full attribute data. Attributes that had too much missing data have been removed and rows that had missing data have been filled.

After much alteration by filling in rows with few missing attributes such as, ACE1, DBF1, FSP1 and FSP2. I was unable to gain any percentage of accuracy. I then used a filter examples operation to remove the attributes missing large values such as NPA1, NPA2 and NPW2. I couldn’t find any gain advantage from altering these rows and had a final accuracy percentage of 89.67%.

I decided to go back to my k-NN attempt to try achieving a better accuracy percentage. Revisiting my k-NN method with the current operations in my process gave me an initial value of 76.9%. I began removing operators and altering the rows of data I had previously removed.

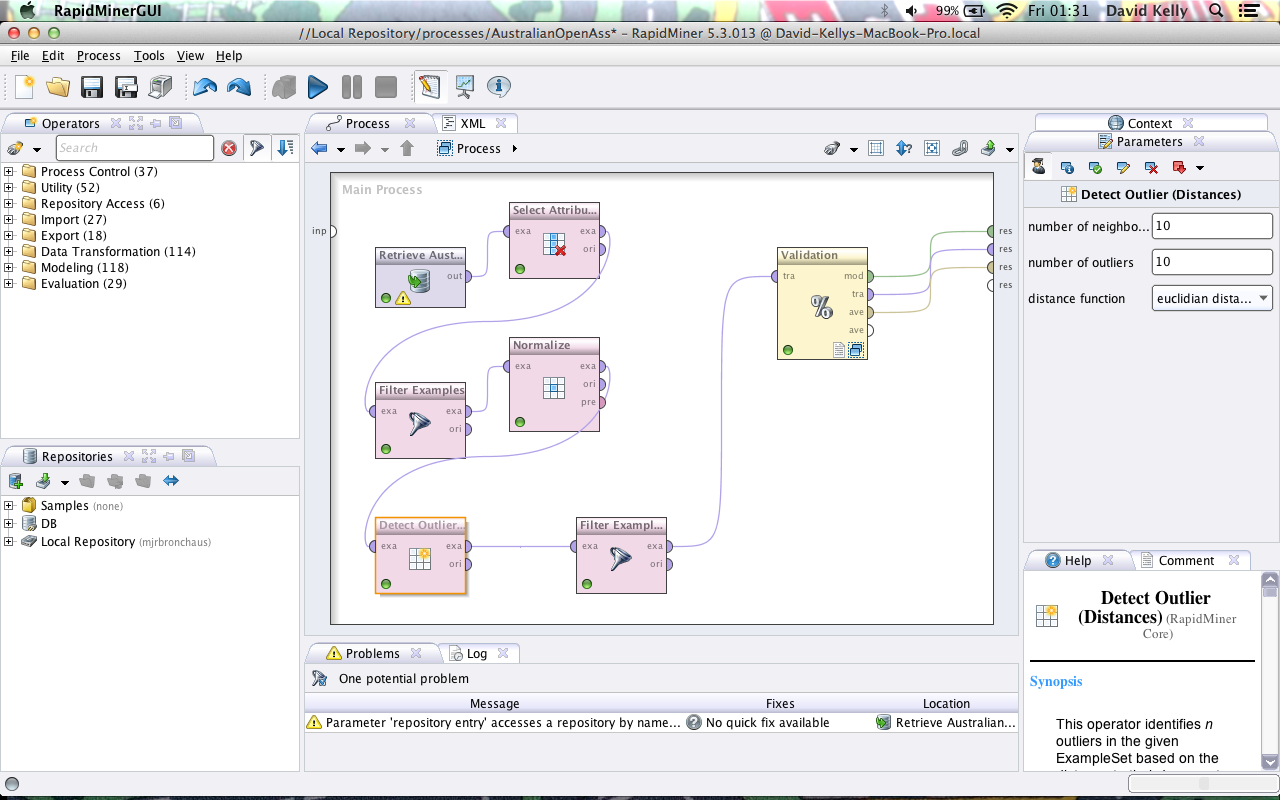
3.4 Integrating Data

Immediately I started finding more joy with k-NN. I removed the same attributes from my first attempt with the decision tree and set my K value to 10 and had an initial accuracy percentage of 85%.



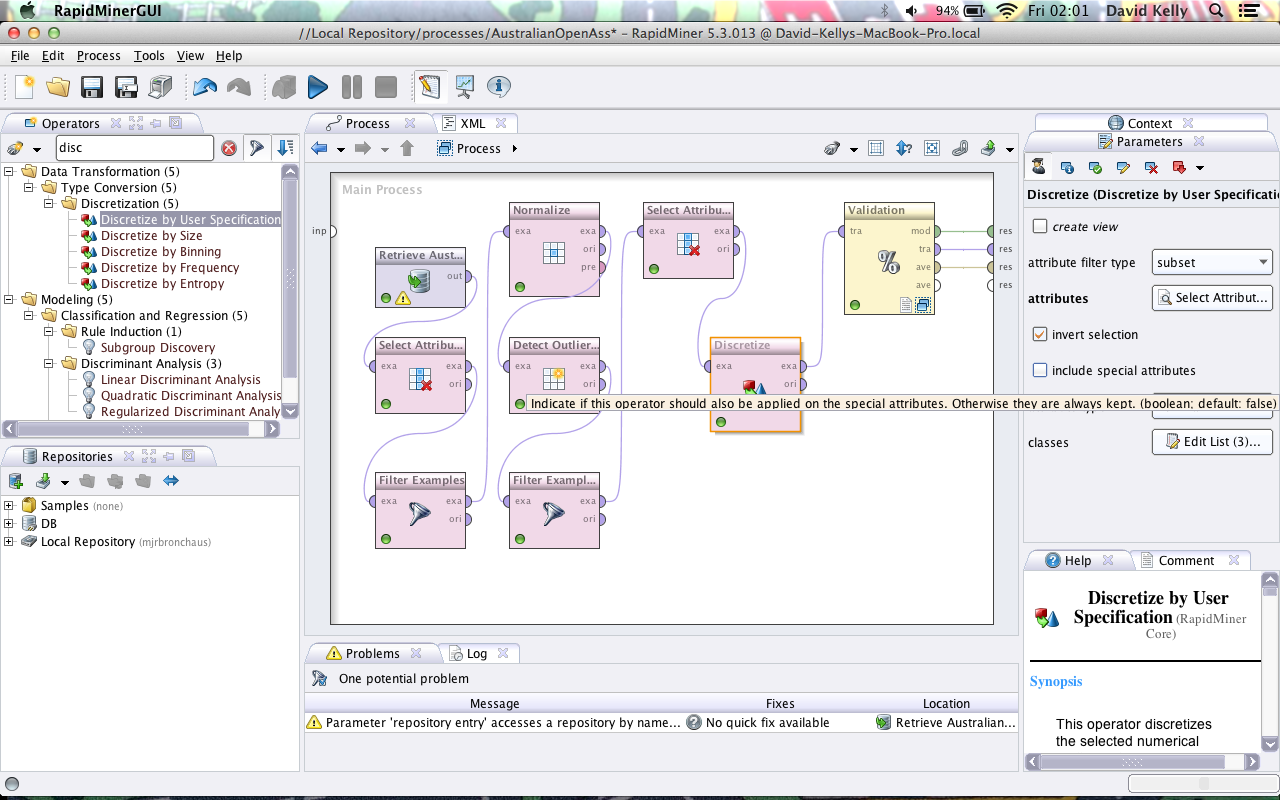
I used the select attribute operator to remove the correlating data and the ST sets that had too many missing values. I filtered the data with no missing values and passed them through a normalize operator to rescale the transform range between 0 and 10. The attributes varied in size too much and for k-NN to work efficiently I needed to flatten the range of values. This gave back an accuracy percentage of 89%.

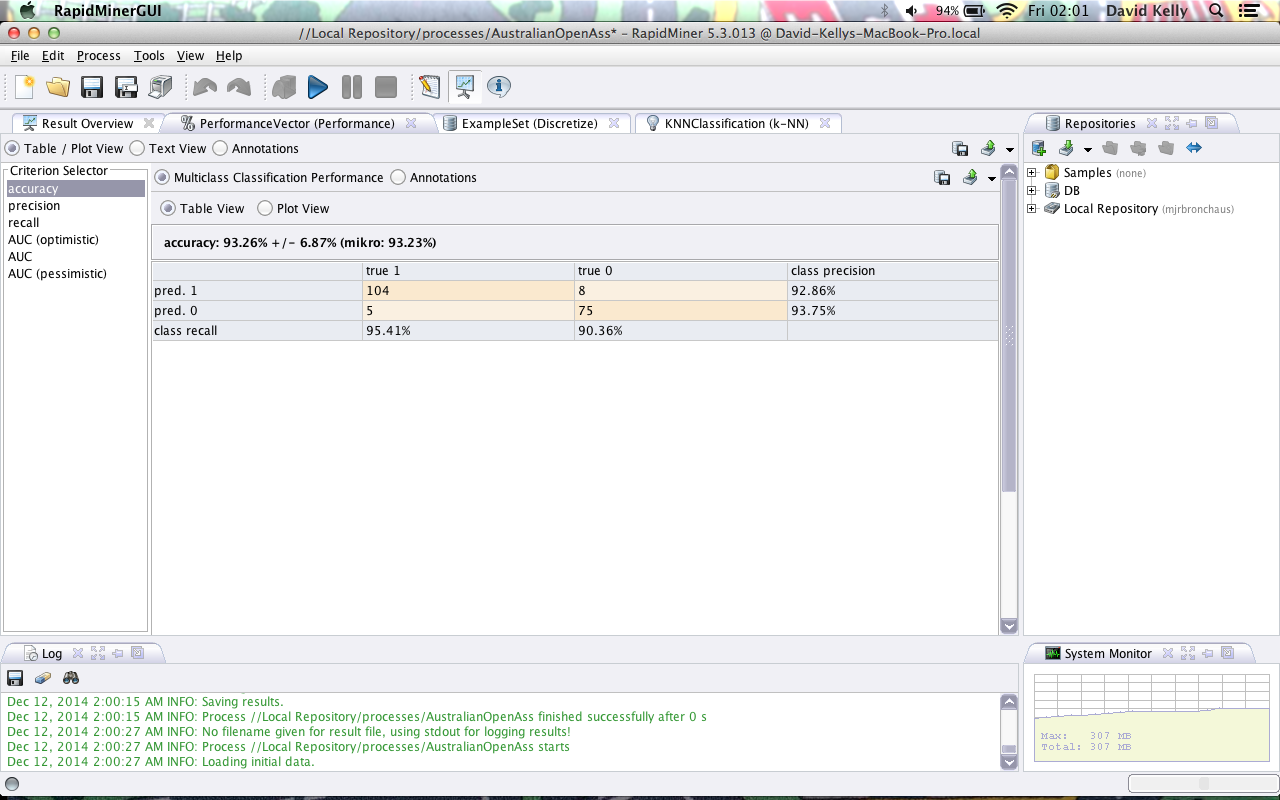
I then implemented a detect outlier operator to find 10 of the furthest outliers and passed the results through a filter example which removed them from the data set. This bumped up the accuracy percentage to 92.16%



After detecting and removing the outliers, I was left with an attribute column of “false” values. In order to remove the column I added a select attribute operator and set the regular expression to “outlier” and included special attributes to remove the now useless column.

I added a sample operator to the process to examine if all the rows in the data set were needed to retain the accuracy percentage. After implementing the sample operator and selecting half the data set the accuracy percentage dropped to 75%. I decided that the data set needed to remain the same size for the process to achieve a higher percentage of accuracy.





The final step I took in achieving the highest accuracy percentage I could was binning. I used discretize by user specification and made 3 groups wit values .3, .6 and .9. I added the break points created and break points won data in to these three groups. Up on doing so I achieved an accuracy of 93.26%. After trying both a decision tree and k-NN I found that k-NN was better suited to this task and I had trouble achieving a percentage higher than 89% with a decision tree method.

3.5 Formatting Data

I now had to format the data to be understood by a model. I had to ensure that the final values from the data set were set to categorical or numeric for a neural network model. I am also going to be using a Naïve Bayes Model so after formatting for categorical and numeric I will have to set up the data as categorical for the Naïve Bayes model to work.

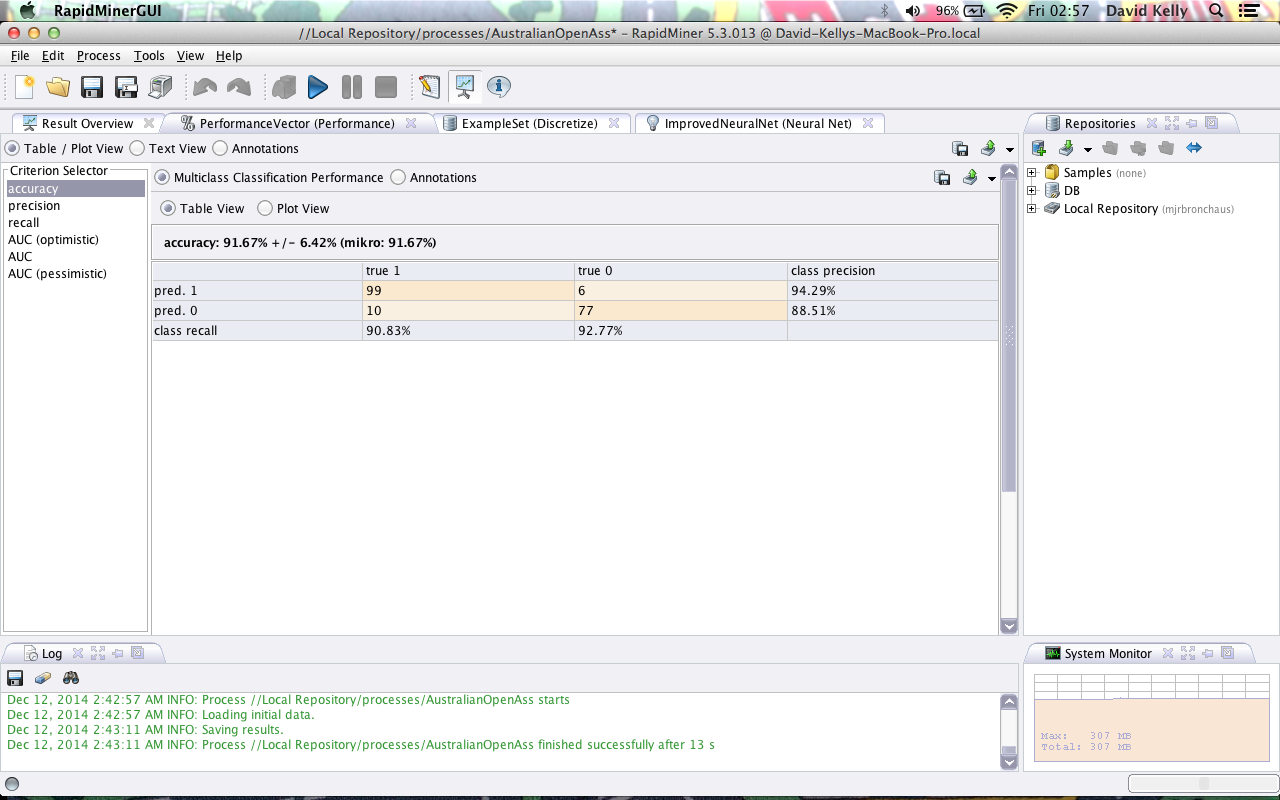
4.0 Modeling

4.1 Select Modeling Techniques

I have decided to use both Neural Network and Naïve Bayes to model my data. I prepared my data using both decision trees and k-NN, so I will use two of the other algorithms I have studied to model the data. The Model is built and the data is ready for testing. I will begin with Neural Network.

4.2 Neural Network Model

The neural network model is most efficient with normalized numeric input. I have formatted the data the data and set the values to numeric. I ran the neural network operator from within a building block with all the default operation values. A neural network is capable of predicting complex relationships, which might help increase accuracy percentage of my model.

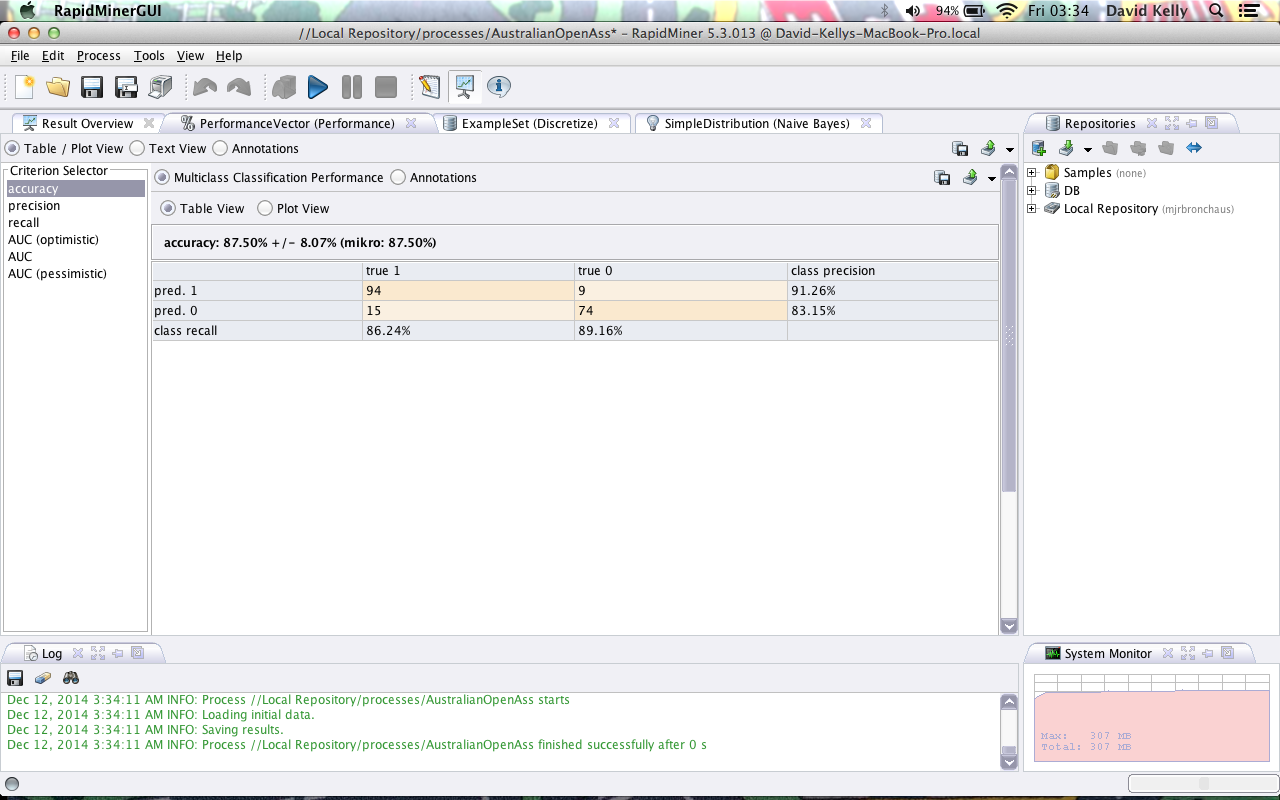


The neural network model ran with 91.67% accuracy and precision of 90%. I realized after running the model that the neural network operator had the built in normalization checked as default. I unchecked the built in feature and the accuracy percentage increased to 93.7%.

I have learned from neural networks that over fitting can be an issue. The model can memorize the training data but when implemented on a new set of data presented to the model, errors can occur. The output of the network is not easily understood and is represented by nodes and hidden nodes. Neural networks can be slow depending on the complexity of the pattern. In my case I have avoided the pitfalls of over fitting and the pattern is not complex enough to cause large time costs.

4.3 Naïve Bayes Model

Although Naïve Bayes can be a slow model to learn, with my data set I didn’t find any issues. Naïve Bayes has the ability to handle missing data and outliers that could be good for a project like this as I may have missed something that I missed in the training set. Naïve Bayes accept all input types but categorical achieves the best results. The data set I am using is numeric and may have an impact on the accuracy percentage of the Naïve Bayes.



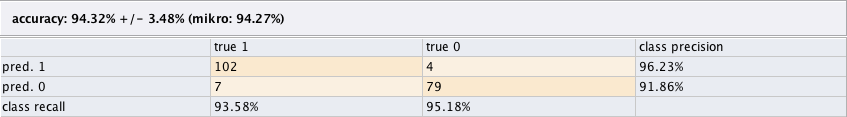
The Naïve Bayes model shows an accuracy of 87.5% on the test data. This could be because the data is numeric or because my data set is too small. I decreased the sample size and tried the model again and there is a drop in percentage accuracy. I feel that if I had a larger data set then the accuracy might be higher.

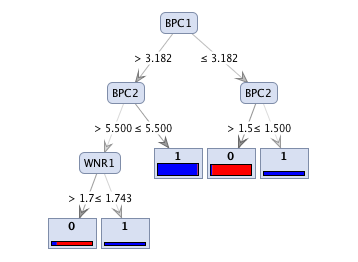
5.0 Evaluation

5.1 Results

The final results of the model are respectable and achieve the goals set out at the beginning of the project. The goal was to create a model that can predict a match winner with over 90% accuracy. Using the models, Decision Tree and neural network I was able to reach accuracy levels of over 92%. The Naïve Bayes model was the lowest scoring accuracy percentage with 87.5% with k-NN receiving an accuracy percentage of 89.55%.

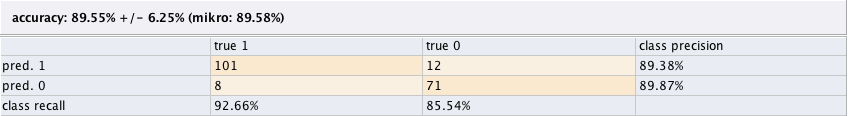
5.1.1 Decision Tree Results

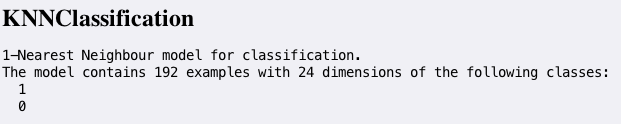




I initially began the project with a Decision Tree but found problems with achieving an accurate prediction with this model. After completing an accurate model with k-NN I decided to go back to the Decision Tree model and see if I could enhance the prediction accuracy. By altering the attributes slightly I was able to attain an accuracy percentage of 94.32%

5.1.2 k-NN Results

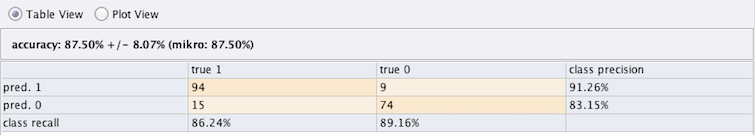




I began using the k-NN model when I had cleaned up the data in my process. I was using k-NN when I normalized the data. I was receiving higher percentage of accuracy until I began tweaking the data in my process to suit Neural Networks. The final accuracy percentage does not reflect the best k-NN accuracy percentage I achieved.

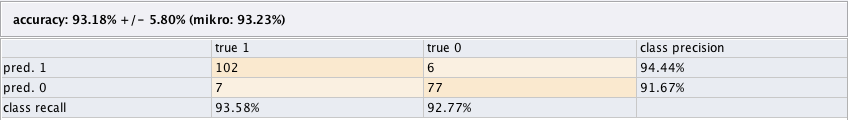
The final accuracy percentage was 89.55%.

5.1.3 Naïve Bayes Results



The best Naïve Bayes accuracy percentage I achieved was 87.50%. I realize that this model is more suited to categorical data and not numeric values. I also feel that the data sample is too small for the model to accurately predict a match winner. When I decreased the sample data size the accuracy of the model decreased also.

5.1.4 Neural Network Results



The neural network model had the longest cost on learning the data. Although the cost wasn’t huge on a small data set like this project, I can understand why the model is sometimes overlooked as it also has a difficult results screen. The read from the Decision Tree had greater accuracy, had less cost and displays a readable result. The Neural Network model achieved an accuracy percentage of 93.18%. Before running this model I first had to format the data to allow the Neural Network to learn the data.

5.2 Review

I was initially surprised to find that after my inability to find a successful method of implementing a Decision Tree, I achieved my highest accuracy percent after working on the process and cleaning the data set further. The result does not surprise me anymore as I just reduced all the noise and outlier data across the data set. This shows that the final data quality of the set after cleaning was of a high standard.

If I was to recreate this project I would enlarge the data set size in the testing phase as I fell the original data set was too small and made it very difficult to get an accuracy prediction any higher than the values I was achieving.

In the end I ran the data set across 4 models and had similar accuracy percentage although only two models (Neural Network, Decision Tree) achieved the goals set out at the beginning of the project. I followed all the steps in the CRISP-DM model, made the data set flexible enough to work with different models and achieved accuracy levels of 93% and 94%.

I feel that the project was successful to a degree. I have learned many methods within the phases of CRISP-DM. In the data understanding phase I learned how to find attributes and factors that can be considered noisy and irregular and also attributes that can have minimal or no effect on data results (text values in numeric processes). In the data preparation phase I learned how to use operators within rapid miner to manipulate the data set and remove, replace, alter and scale values within an attribute set. In the modeling phase I learned which models work best on certain data set types, the size of data sets needed to create models with accurate prediction percentages.

In conclusion the project was successful and I have learned many methods and techniques involved in the CRISP-DM model.