UP877962

ADTA report

part 1

Question 1 - Basic analytics

The following diagram (figure 1.1) shows the solution presented for question one, there are slightly more nodes than requested however their use is justified by the insight gained from them. The graph visualisations largely focus on attributes relating to the social grade attribute as this will be used during classification and as such some correlations may be drawn between information gained in this question and later ones.

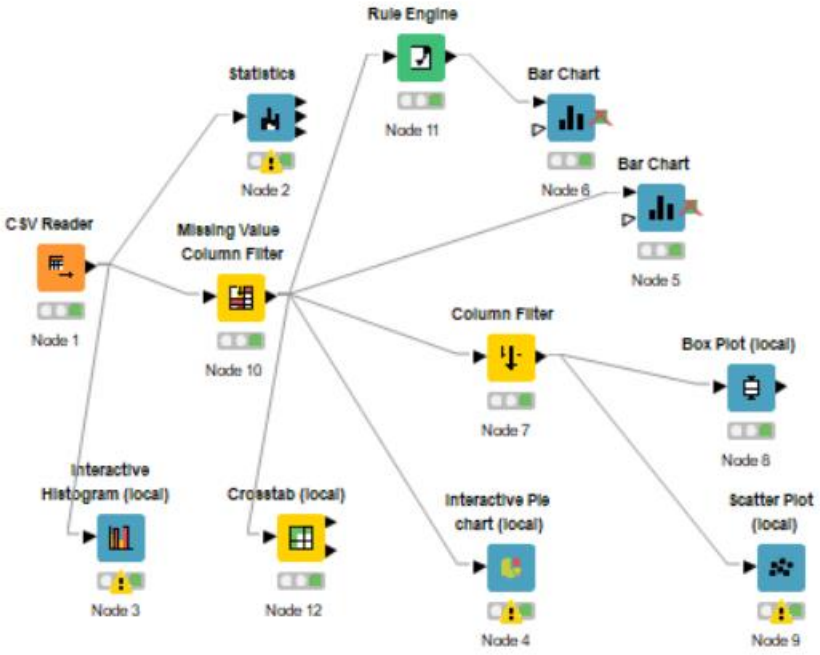
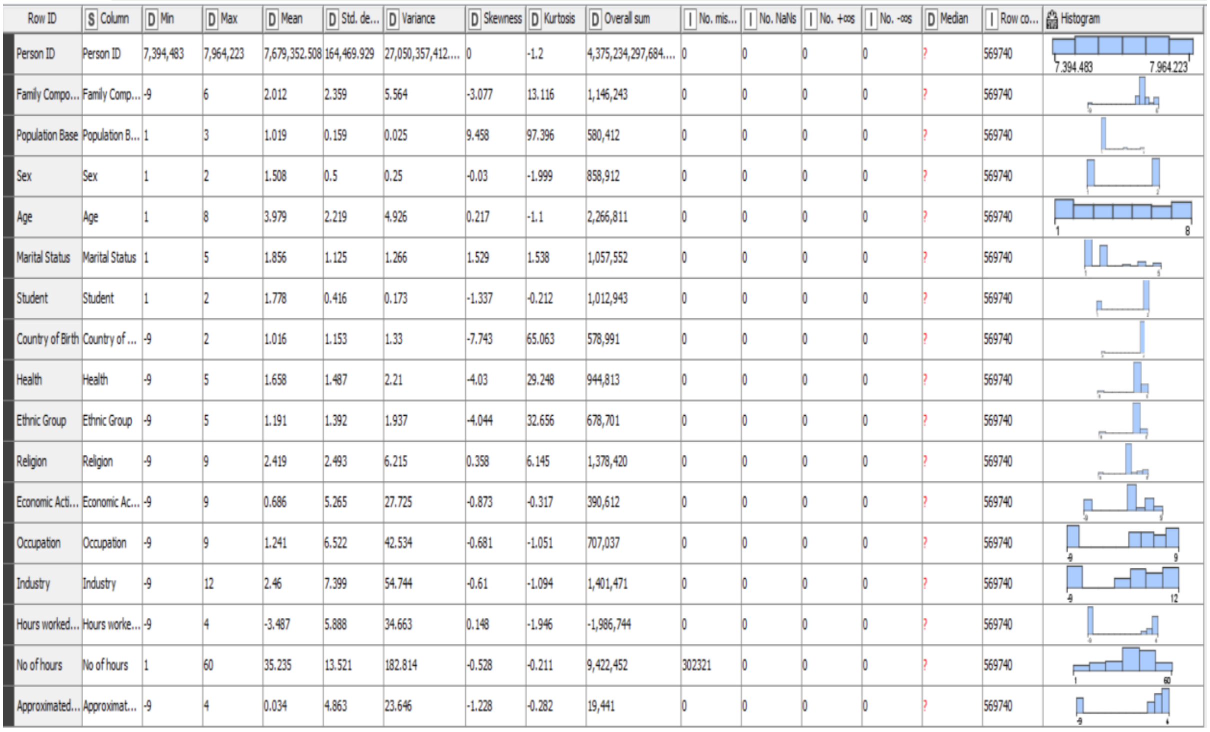


Figure 1.1, An image of the KNIME workflow used for this question.

The Number of hours column does not appear in the variables list provided with the census data and has over 50% missing values so it has been assumed an error and removed from any visualisations except for the following overview of all columns in the dataset (Table 1.1), this column will also be removed from some questions as it is not very useful in terms of gaining insight into the data. The person ID column was also removed for the purposes of visualising the scatter plot and box plot nodes, this attribute has no meaning in these graphs and could also be removed from the dataset for future questions as there is no insight to be gained from it.

Table 1.1, The table below shows the basic statistics for all the columns in the dataset.



Row ID	Column	Min	Max	Mean	Std. de...	Variance	Skewness	Kurtosis	Overall sum	No. mis...	No. NaNs	No. +∞s	No. -∞s	Median	Row co...	Histogram
Person ID	Person ID	7,394,483	7,964,223	7,679,352.508	164,469.929	27,050,357,412....	0	-1.2	4,375,234,297,684....	0	0	0	0	?	569740	7.394.483 7.964.223
Family Compo...	Family Comp...	-9	6	2.012	2.359	5.564	-3.077	13.116	1,146,243	0	0	0	0	?	569740	
Population Base	Population B...	1	3	1.019	0.159	0.025	9.458	97.396	580,412	0	0	0	0	?	569740	
Sex	Sex	1	2	1.508	0.5	0.25	-0.03	-1.999	858,912	0	0	0	0	?	569740	
Age	Age	1	8	3.979	2.219	4.926	0.217	-1.1	2,266,811	0	0	0	0	?	569740	1 8
Marital Status	Marital Status	1	5	1.856	1.125	1.266	1.529	1.538	1,057,552	0	0	0	0	?	569740	
Student	Student	1	2	1.778	0.416	0.173	-1.337	-0.212	1,012,943	0	0	0	0	?	569740	
Country of Birth	Country of ...	-9	2	1.016	1.153	1.33	-7.743	65.063	578,991	0	0	0	0	?	569740	
Health	Health	-9	5	1.658	1.487	2.21	-4.03	29.248	944,813	0	0	0	0	?	569740	
Ethnic Group	Ethnic Group	-9	5	1.191	1.392	1.937	-4.044	32.656	678,701	0	0	0	0	?	569740	
Religion	Religion	-9	9	2.419	2.493	6.215	0.358	6.145	1,378,420	0	0	0	0	?	569740	
Economic Acti...	Economic Ac...	-9	9	0.686	5.265	27.725	-0.873	-0.317	390,612	0	0	0	0	?	569740	
Occupation	Occupation	-9	9	1.241	6.522	42.534	-0.681	-1.051	707,037	0	0	0	0	?	569740	-9 9
Industry	Industry	-9	12	2.46	7.399	54.744	-0.61	-1.094	1,401,471	0	0	0	0	?	569740	-9 12
Hours worked...	Hours worke...	-9	4	-3.487	5.888	34.663	0.148	-1.946	-1,986,744	0	0	0	0	?	569740	
No of hours	No of hours	1	60	35.235	13.521	182.814	-0.528	-0.211	9,422,452	302321	0	0	0	?	569740	1 60
Approximated...	Approximat...	-9	4	0.034	4.863	23.646	-1.228	-0.282	19,441	0	0	0	0	?	569740	

The following box plot (Figure 1.2) shows all the variables except the No of hours attribute for reasons stated, it can be seen from this chart that the religion variable has the largest number of outliers with family composition, marital status, health, and ethnic group also experiencing large numbers of outliers.

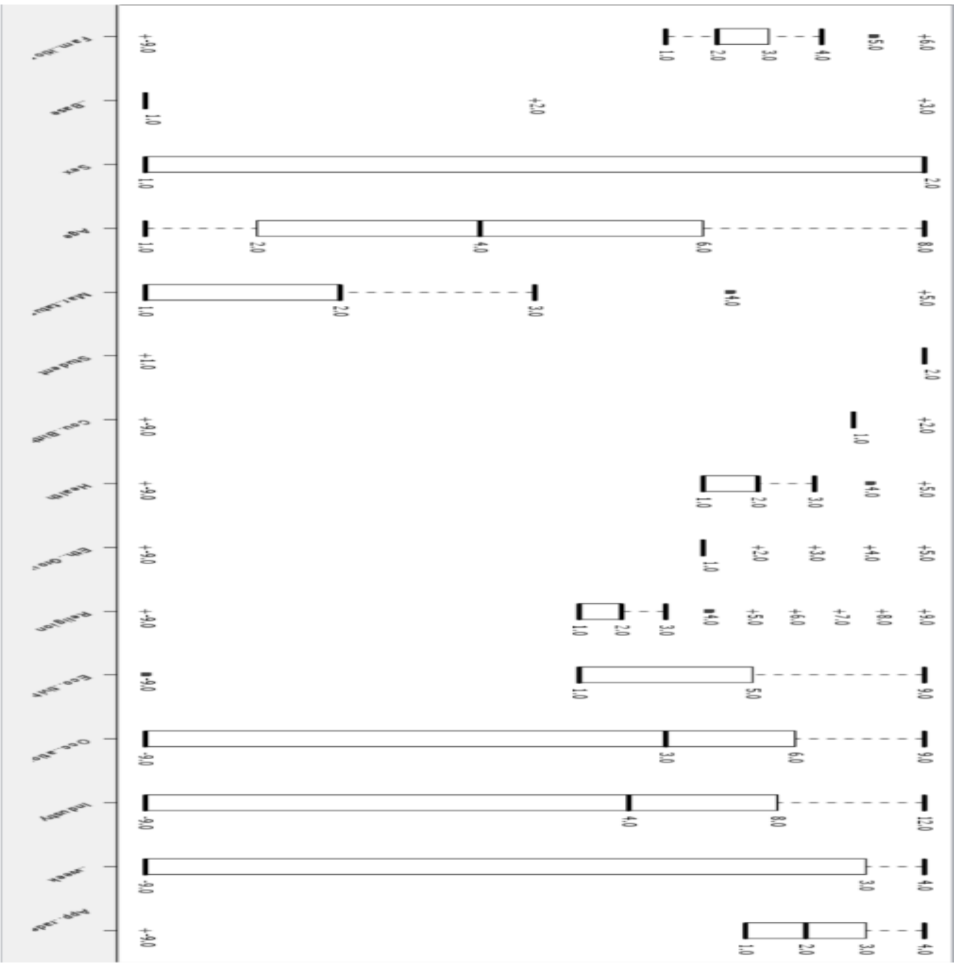


Figure 1.2, A box plot showing the remaining variable, this chart helps to identify outliers.

For the purposes of readability, it was decided to change the column names for the resident type of variable from the original “H” and “C” into “Community care resident” and “Non community care resident” in preparation for the bar chart output (Figure 1.3), this was done using a rule engine node. For the future questions this may be incorporated along with any other rules required. As can be seen most people in the census live in a home care environment.

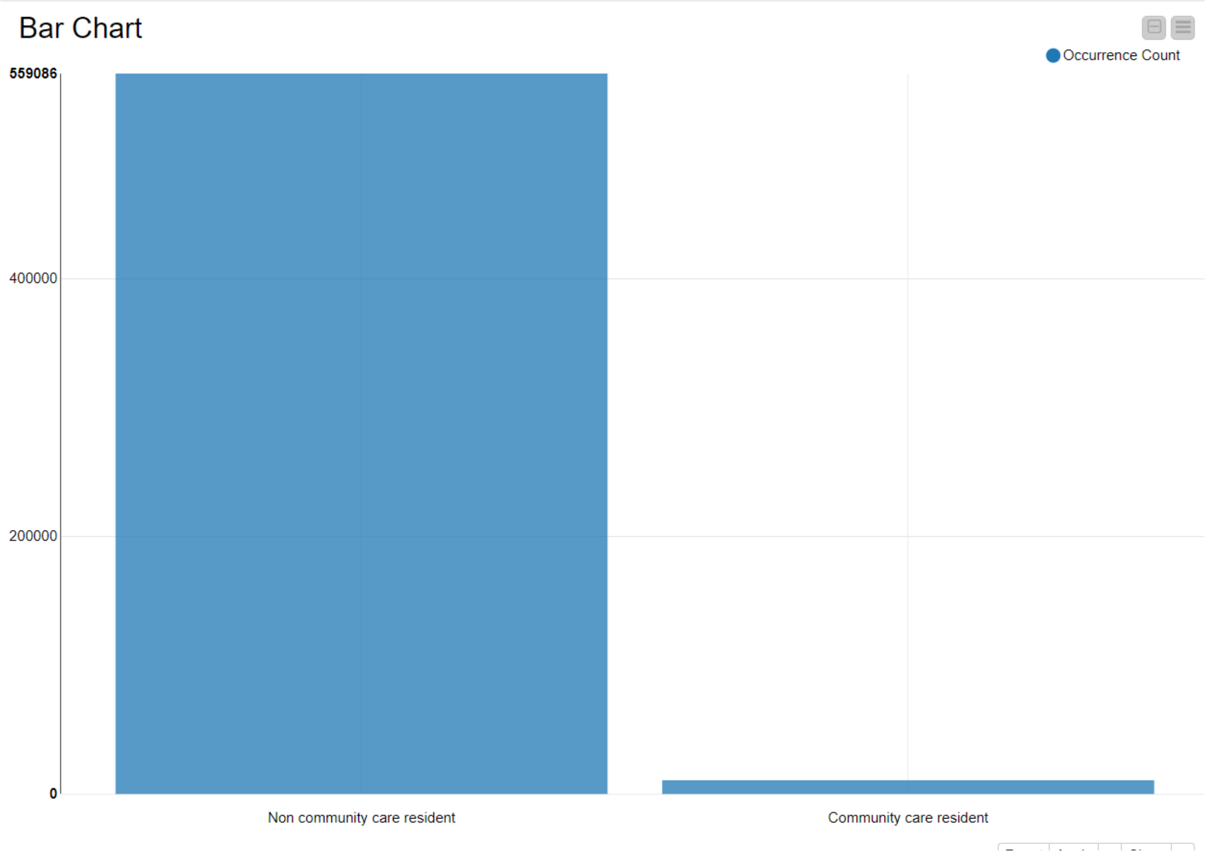


Figure 1.3, A bar chart displaying the residence types and their distribution.

The following pie chart (Figure 1.4) displays a subsection 2500 rows displaying the count of occurrences in the various approximate social grade categories, it can be seen from this chart that most records fall within the -9 (No code) or 4 (semi-skilled, unskilled, unemployed, the lowest category).

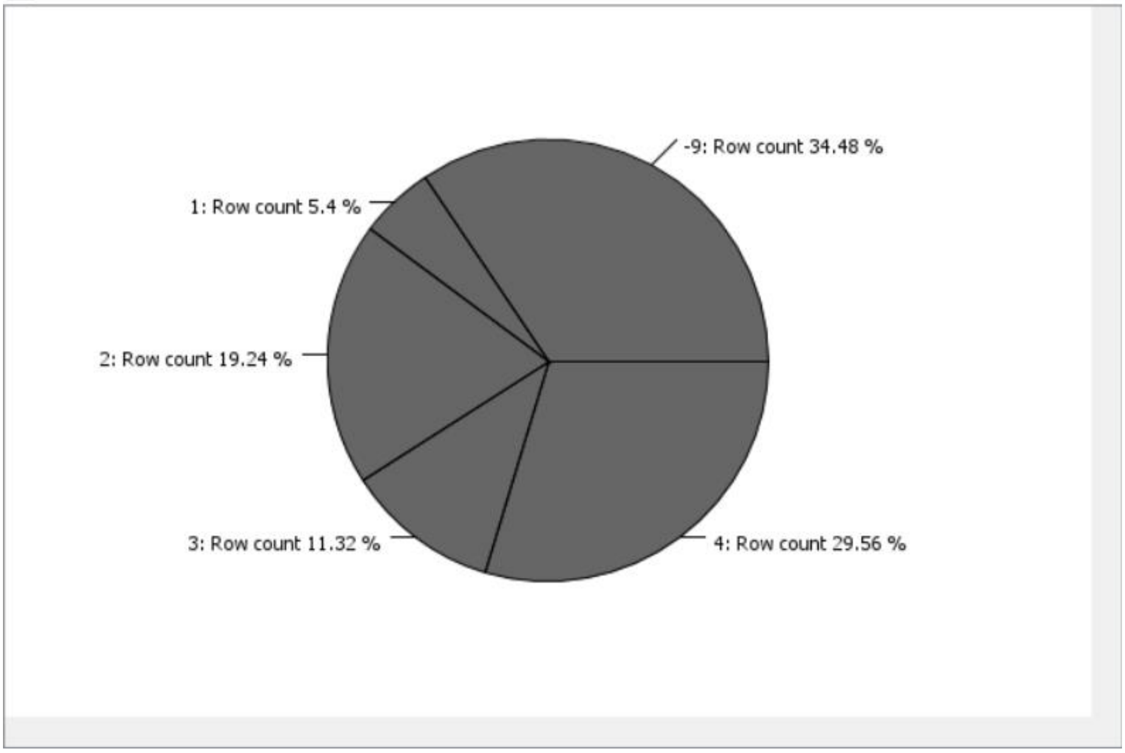
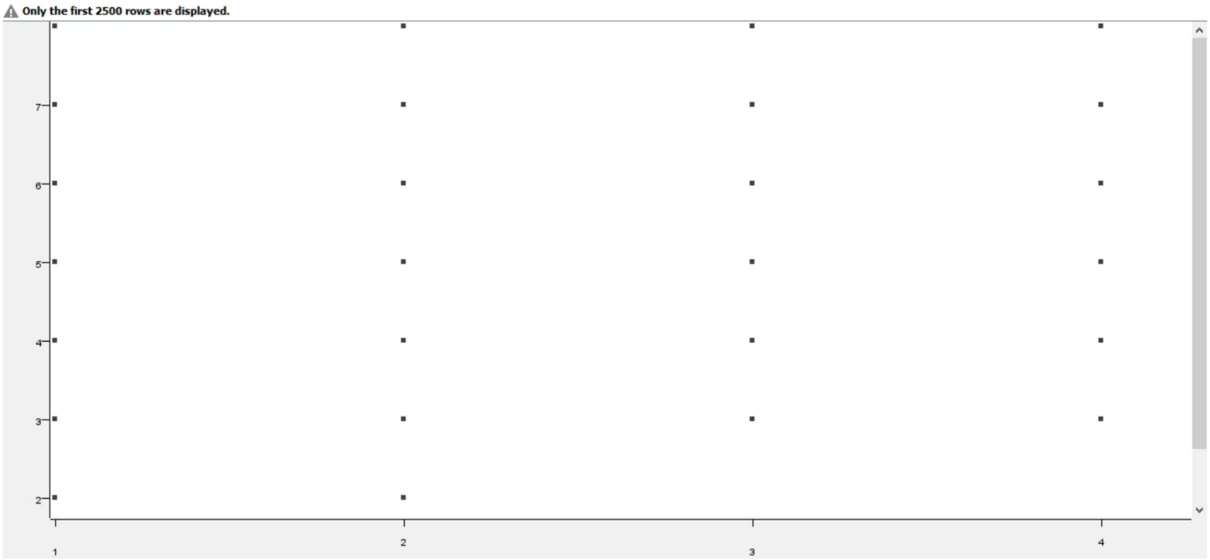


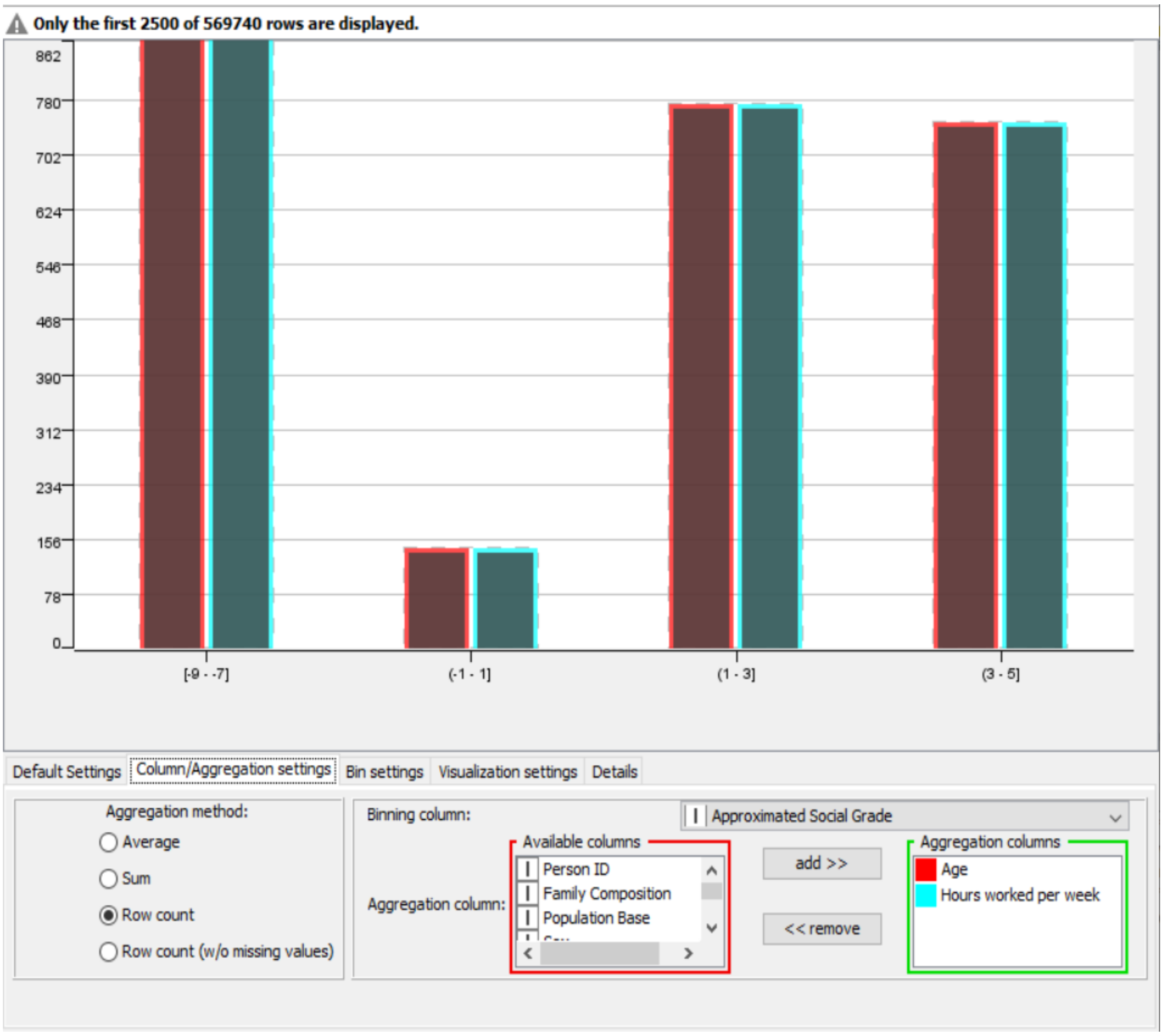
Figure 1.4, A pie chart displaying a subset of the approximate social grade attributes and their occurrences.

The following scatter plot shows the correlation between age and marital status, the obvious insight here is that younger people are less likely to be divorced or separated.



⚠ Only the first 2500 rows are displayed.

Figure 1.5, A scatter plot displaying age (Y axis) and marital status (X axis).



⚠ Only the first 2500 of 569740 rows are displayed.

Aggregation method:

○ Average

○ Sum

● Row count

○ Row count (w/o missing values)

Binning column: Approximated Social Grade

Aggregation column:

Available columns
- Person ID
- Family Composition
- Population Base

Aggregation columns
- 🟥 Age
- 🟦 Hours worked per week

Figure 1.6, A histogram displaying the row counts for age and hours worked per week with approximated social grade used for binning.

After analysing the data there are some columns that can be removed from the data regarding later questions such as the person ID, number of hours, also religion and ethnic group are unlikely to have much meaning to the intended prediction of social grade due the number of outliers in the data.

Question 2 - Clustering

The algorithms chosen for clustering were K-means and basic hierarchal clustering, for both algorithms the No of hours and personID columns were removed this is due to the No of hours column containing mostly missing values and not being a part of the census data overall(does not appear in the variable list supplied), the personID column was removed as it has no bearing on the outcome of clustering and in the case of this data the row number will be sufficient to differentiate between data objects. Simple K-means was also used to try and benchmark results from the K-means clustering but unfortunately it was difficult to achieve any sort of meaningful clustering with k-means despite trying many possible variations of inputs such as only 2 columns and the whole dataset and with values of K ranging from three to twenty. This is likely since K-means is only suitable where the mean is applicable and K-medoids may have been more appropriate to use in this case. The image below (Figure 2.1) displays the workflow used for the k-means clustering.

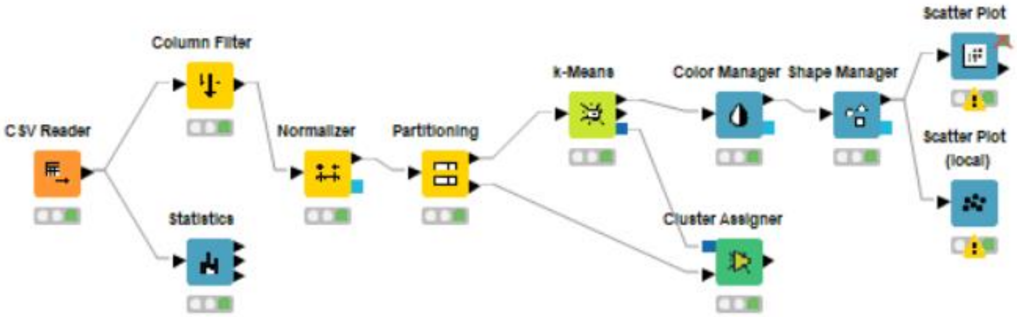


Figure 2.1, KNIME workflow for K-means clustering.

The image below (Figure 2.2) details the workflow used for hierarchical clustering, the simpler method without using distance matrix calculations gave better results although this may be more down to the parameters used, due to this only this method was used.

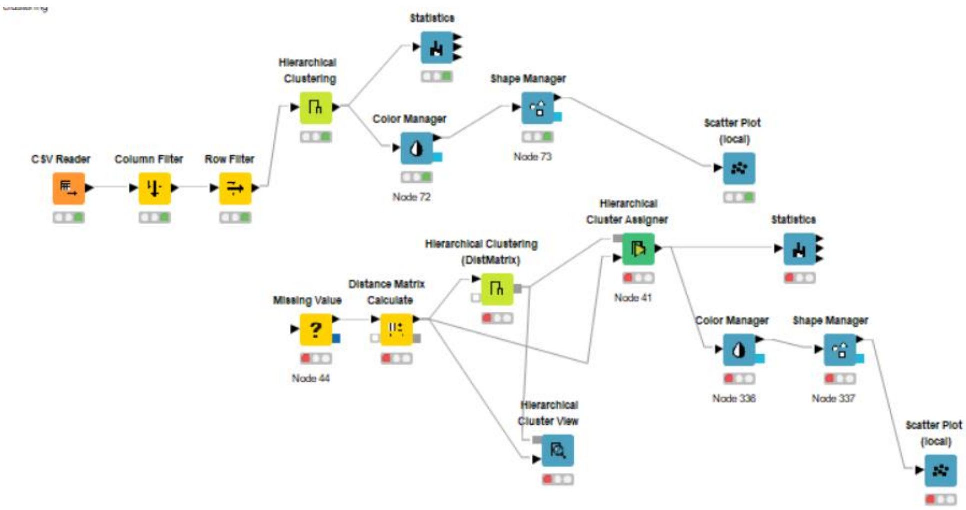
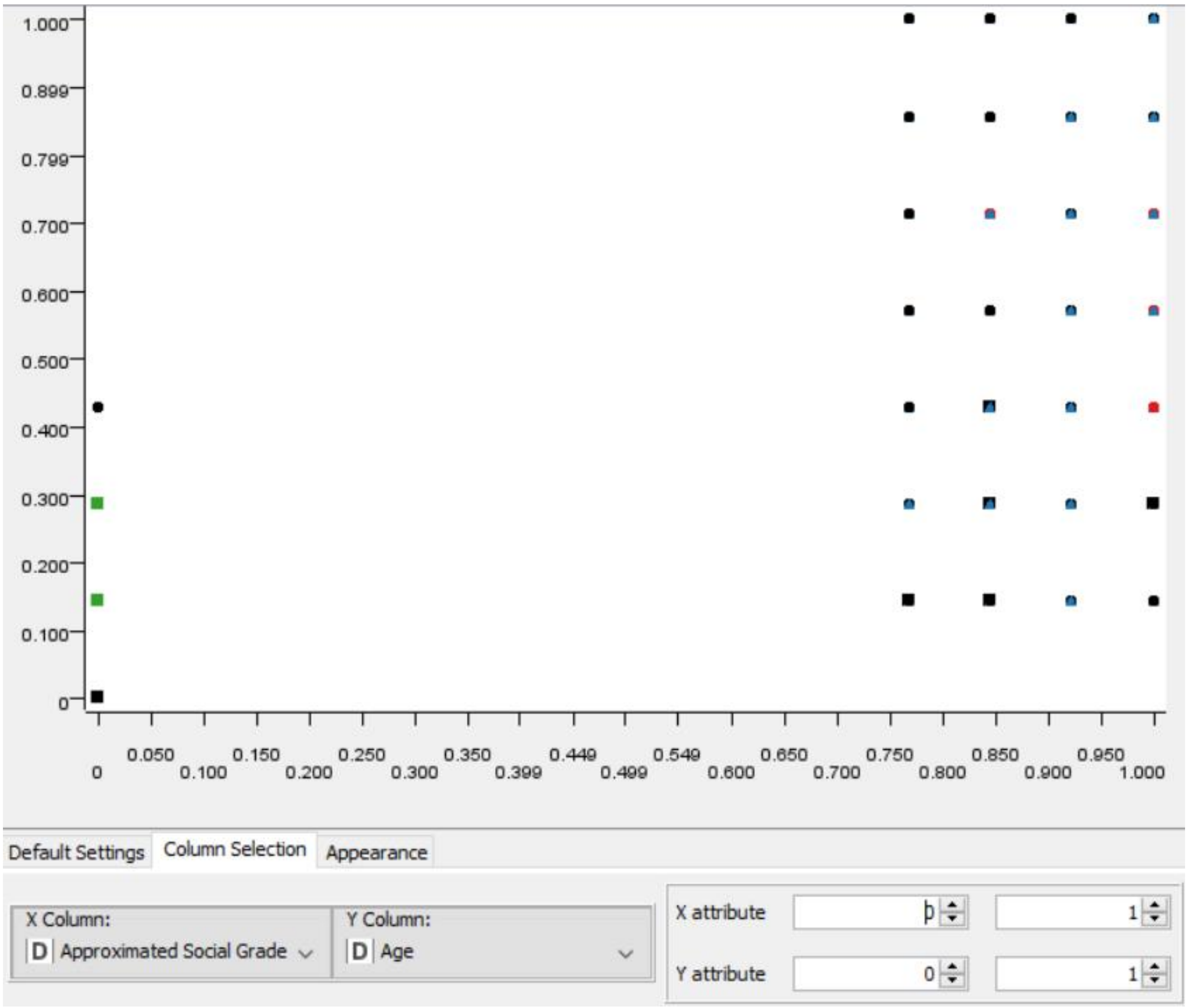


Figure 2.2, KNIME workflow for hierarchical clustering.

K-means clustering (Figure 2.3) with 5 clusters on the age and approximated social grade columns, clustering was not very successful here with no clear distinction between clusters.



X Column: D Approximated Social Grade

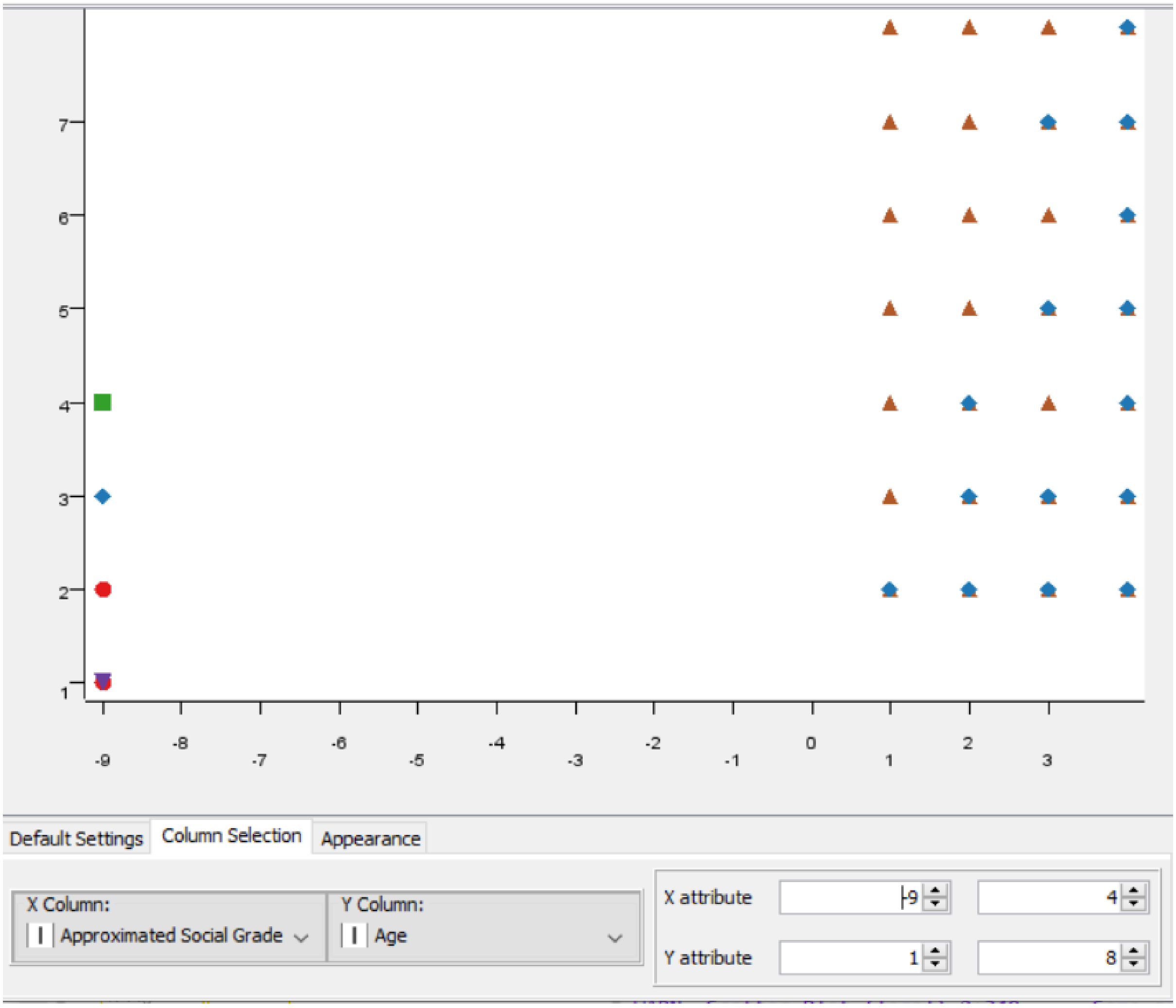
Y Column: D Age

X attribute 0 1

Y attribute 0 1

Figure 2.3, A scatter plot displaying the clustering results.

As can be seen from the following scatter plot (Figure 2.4) the hierarchical clustering with 5 clusters performed better with clear distinctions between all but 2 clusters, although not a lot of insight can be gained from this data.



X Column: Approximated Social Grade

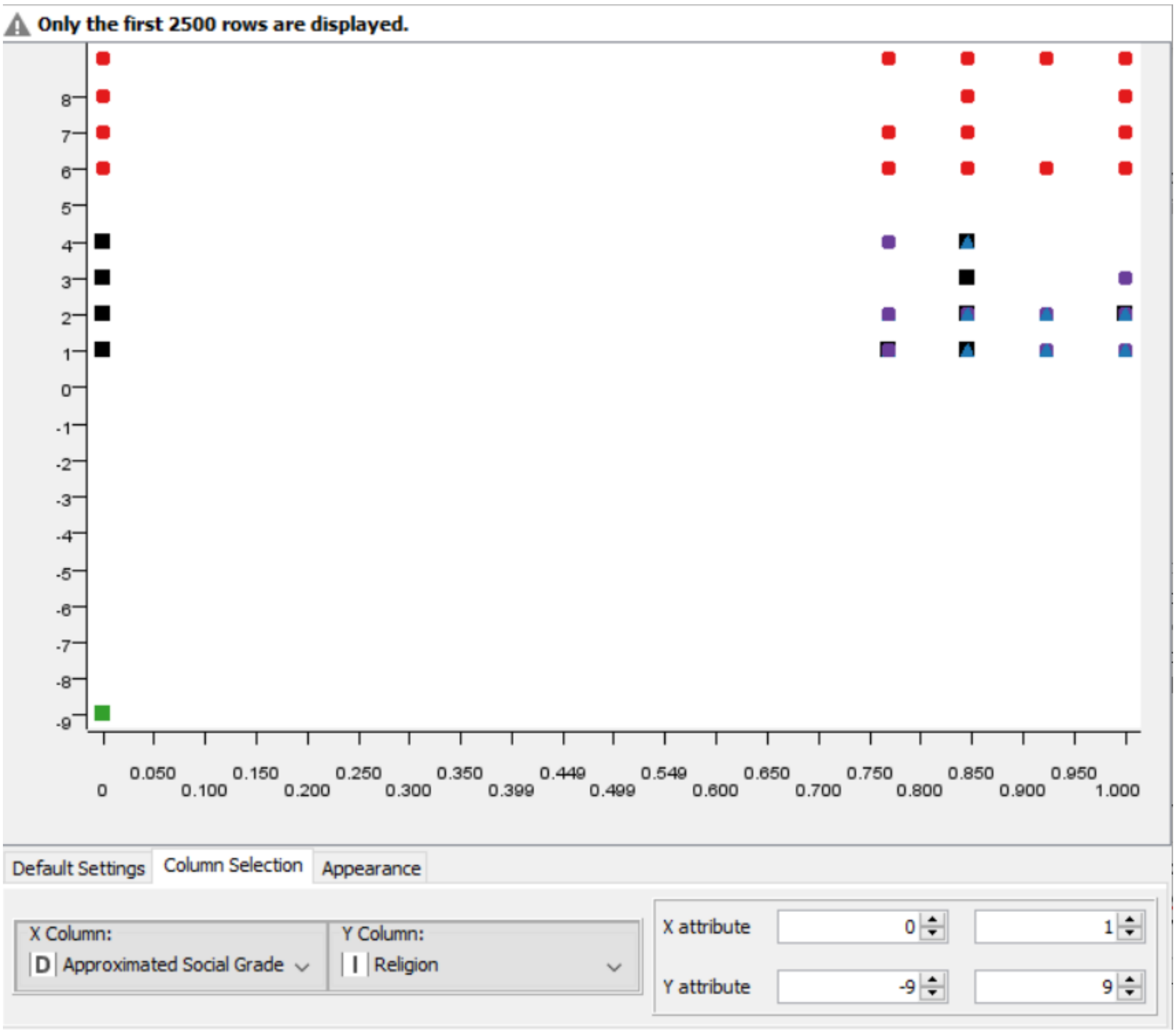
Y Column: Age

X attribute -9 4

Y attribute 1 8

Figure 2.4, A scatter plot displaying the clustering results.

K-means clustering with 5 clusters performed on the approximated social grade and religion columns (Figure 2.5), k-means gave reasonable results with these columns with some clear distinction between cluster 1 and 0 (red and green) however there are still overlaps between the other 3 clusters.


Only the first 2500 rows are displayed.

X Column: D Approximated Social Grade

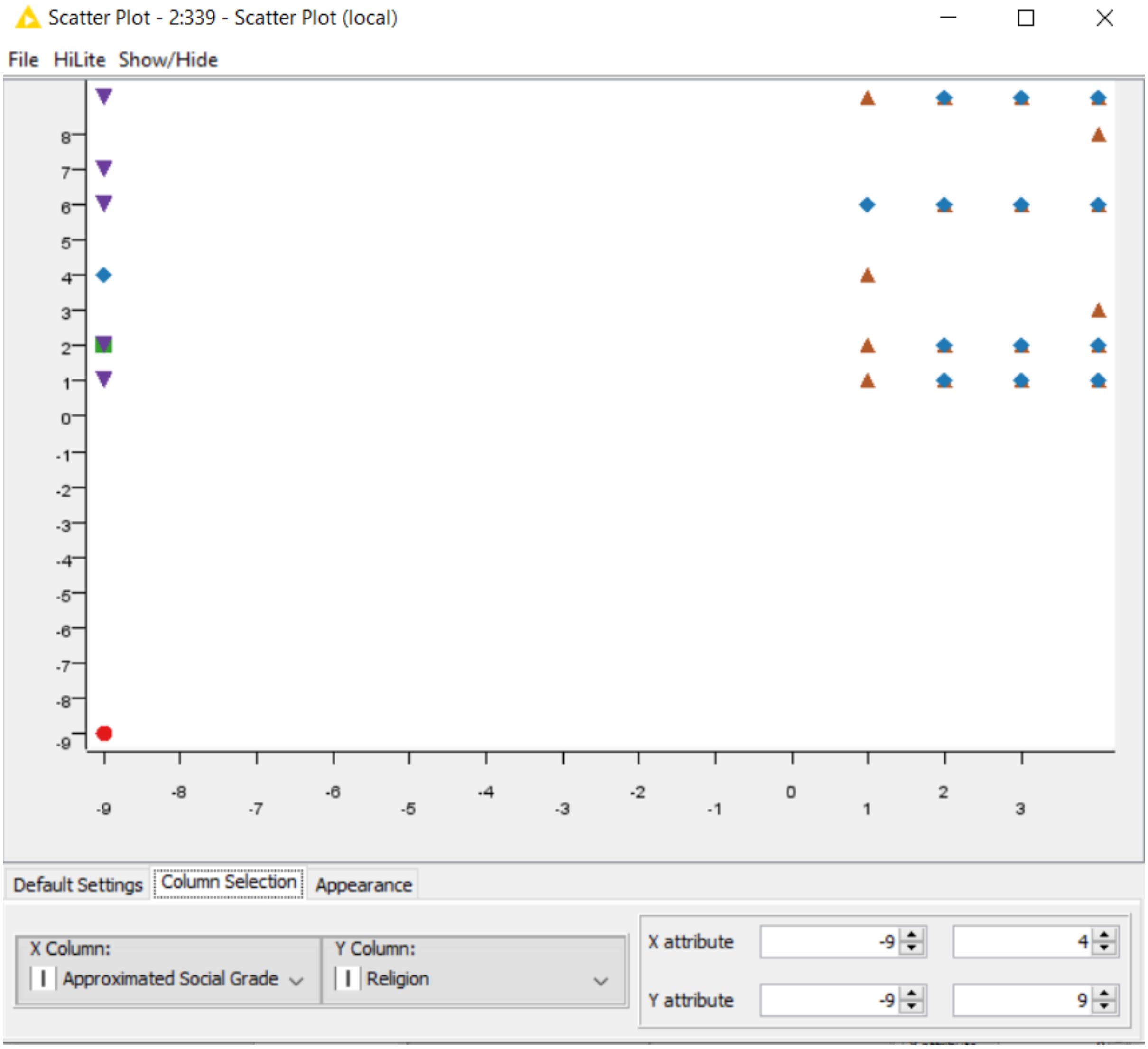
Y Column: I Religion

X attribute 0 1

Y attribute -9 9

Figure 2.5, A scatter plot displaying the clustering results.

The following diagram (Figure 2.6) shows hierarchical clustering performed on the same columns with the same parameters, k-means performed better in this case with only one distinct cluster.



X Column: Approximated Social Grade

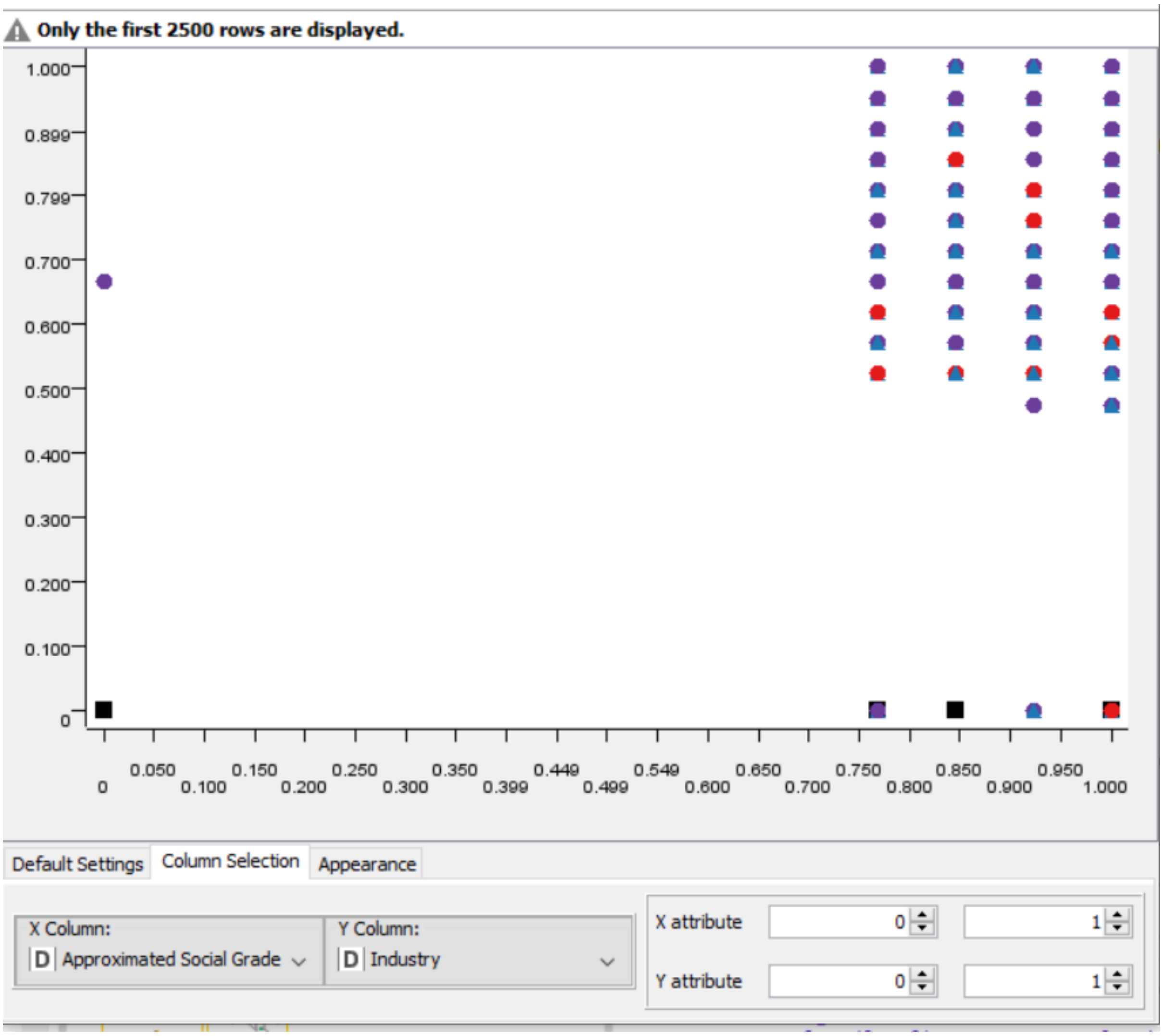
Y Column: Religion

X attribute -9 4

Y attribute -9 9

Figure 2.6, A scatter plot displaying the clustering results.

K-means performed on approximate social grade and industry columns (Figure 2.7).



Only the first 2500 rows are displayed.

X Column: D Approximated Social Grade

Y Column: D Industry

X attribute 0 1

Y attribute 0 1

Figure 2.7, Scatter plot displaying the clustering results.

Hierarchical clustering performed with the same columns and parameters (Figure 2.8). The hierarchical performed slightly better in this case with clear distinctions between 2 clusters.

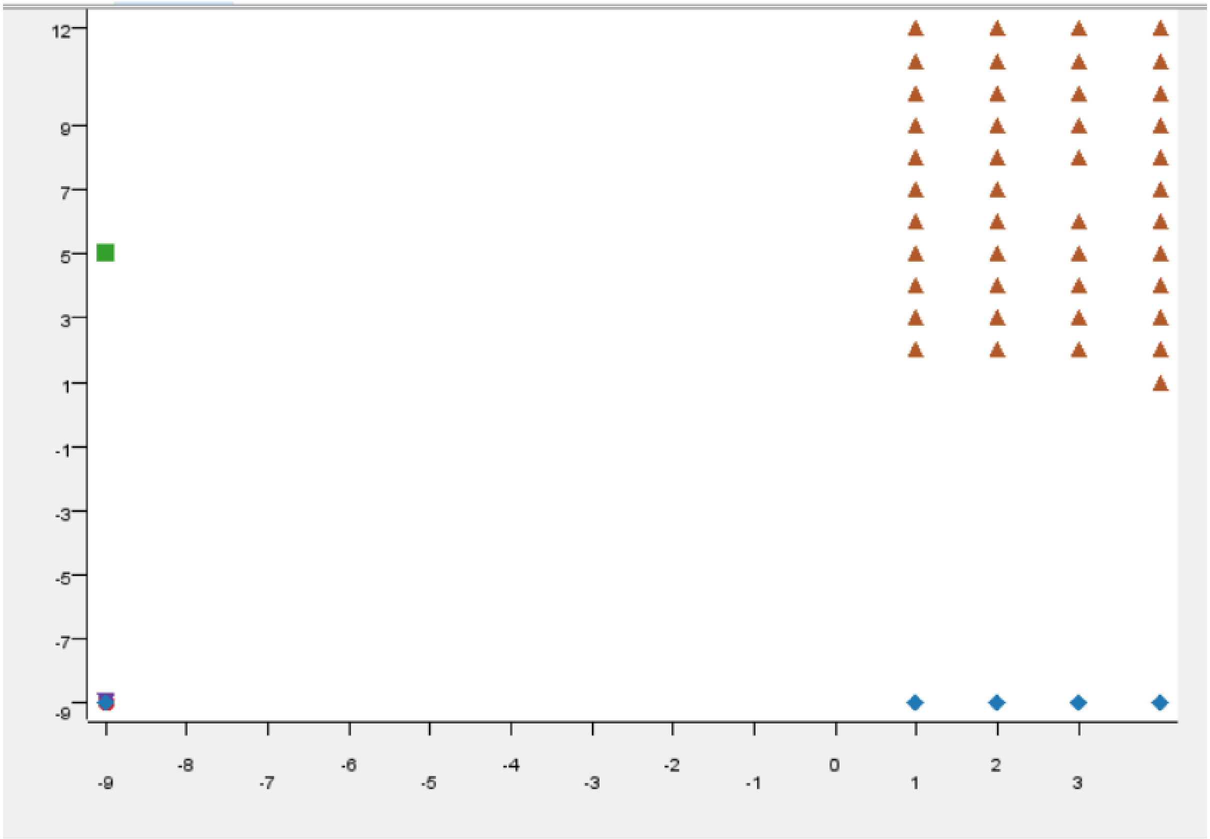


Figure 2.8, A scatter plot displaying the clustering results.

Comparison

The basic hierarchical clustering seemed to give better results with some clear differentiation between clusters when used with the exact same conditions used in the K-means algorithm as can be seen graphically in the scatter plots above, although the data itself does not seem to be very suitable for clustering in its present state and some more pre-processing was likely required to get any good insight into the data.

Question 3 – Classification

For the purposes of classification, the three chosen algorithms for the question were decision tree classification, random forest classification and the naïve bayes classification. For all three algorithms it was decided to transform the numeric types into strings to enable the classifiers to work on the data, this method was chosen for its simplicity and since it is likely to be less computationally expensive than stringing together multiple rule engines to manipulate the columns individually.

Due to the computational requirements of the random forest classifier, it was required to work on a subset of the original data, there were attempts to run the classifier on the whole dataset both of which ran for three hours before the java heap memory allocation that KNIME is allocated was exhausted and the process crashed and as KNIIME is being used over Appsanywhere it was not possible to manually assign more memory. The other two algorithms used ran in similar time (~2 minutes) on the whole dataset which shows how much more computationally demanding random forest classification is in comparison to the others.

The following figures show the three workflows used.

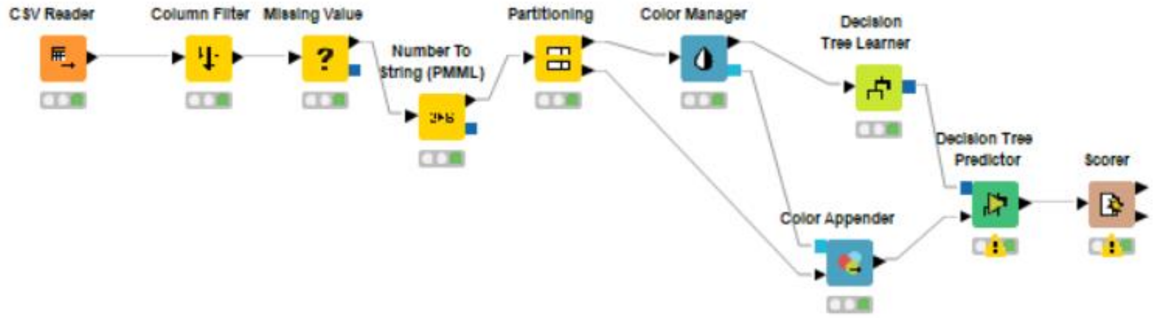


Figure 3.1, The KNIME workflow used for decision tree classification.

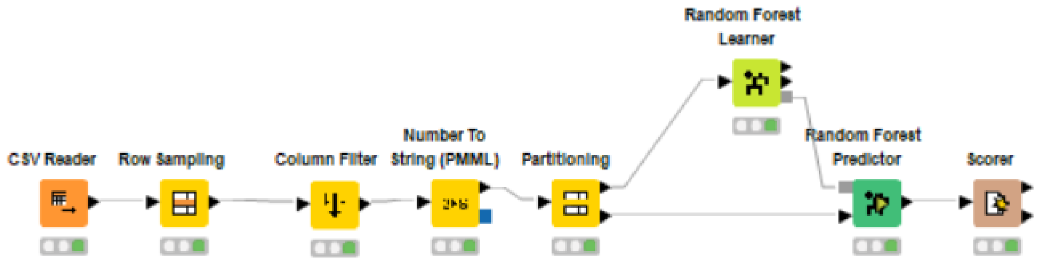


Figure 3.2, The KNIME workflow used for random forest classification.

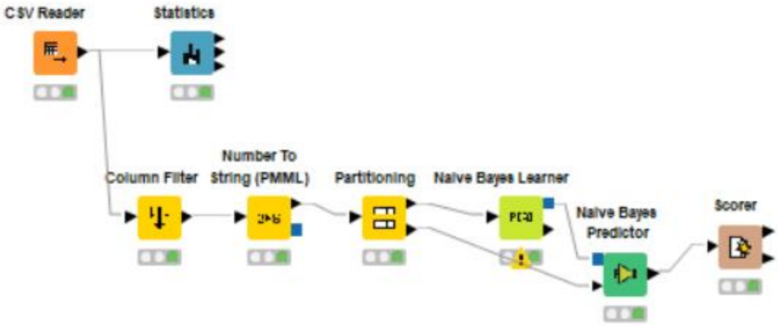
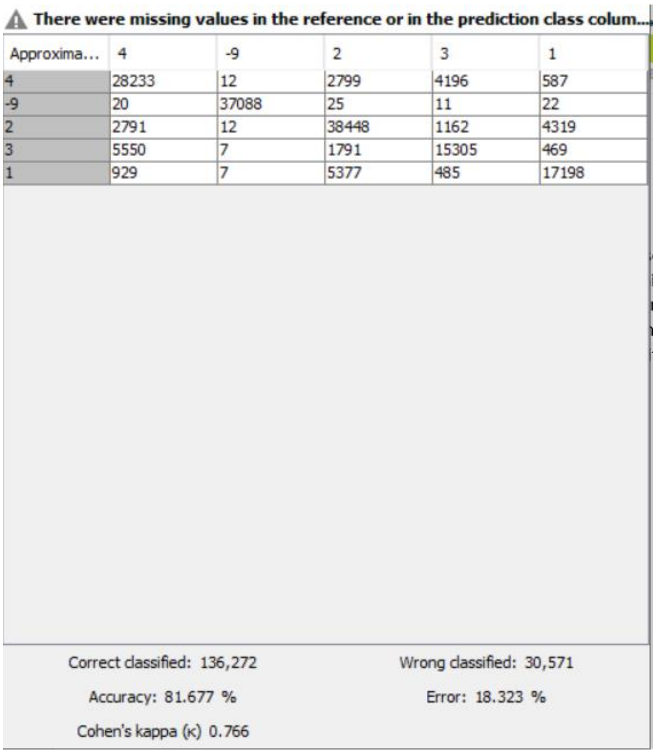


Figure 3.3, The KNIME workflow used for naïve bayes classification.

Comparison

The following section outlines the comparisons between the selected classification algorithms. In each case the partition size selected was 70% to ensure the fairest possible comparison between the algorithms although the random forest classifier was run on only a subset containing 10% of the original data due to hardware limitations, this brought the run times into line with the other classifiers. The No of hours column was removed before classification. Missing values for decision trees were replaced with the most frequent value.

As can be seen in the figures below the random forest classifier was the most accurate with an accuracy of 83.496 albeit with 10% of the original dataset although the decision tree classifier was only 2% less accurate at 81.677 and considering that this classifier used the entire dataset the performance/computational cost ratio means that for larger datasets the decision tree classifier may be more appropriate. The Naïve Bayes classifier performed the worst of the selected methods with an accuracy of only 57.644% suggesting that it is not particularly suited to the dataset used. In terms of the most appropriate classifier to be used on this dataset based on observations would be the decision tree learner purely for the accuracy/computation cost ratio versus the others.



⚠ There were missing values in the reference or in the prediction class colum...

Approxima...	4	-9	2	3	1
4	28233	12	2799	4196	587
-9	20	37088	25	11	22
2	2791	12	38448	1162	4319
3	5550	7	1791	15305	469
1	929	7	5377	485	17198

Correct classified: 136,272

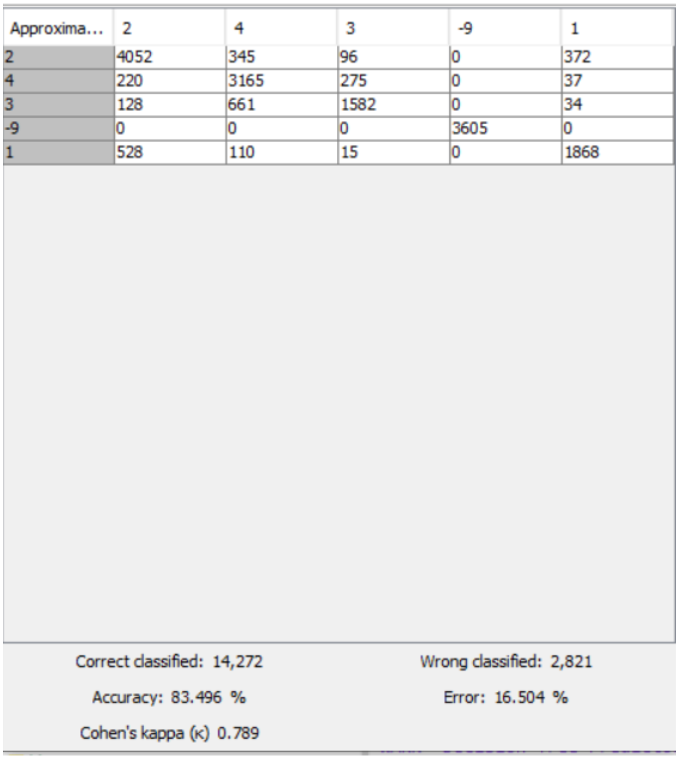
Wrong classified: 30,571

Accuracy: 81.677 %

Error: 18.323 %

Cohen's kappa (κ) 0.766

Figure 3.4, Confusion matrix for decision tree classification



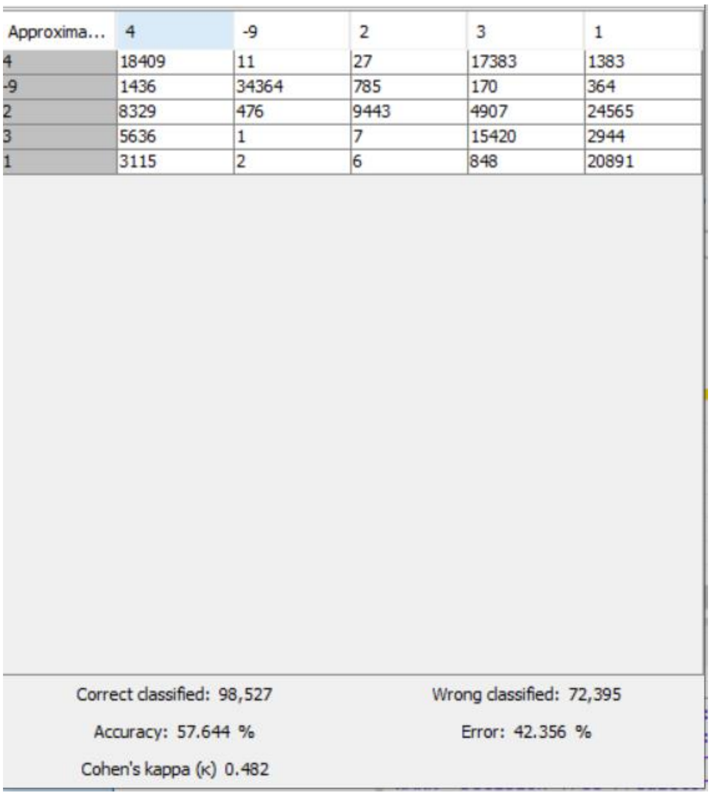
Approxima…	2	4	3	-9	1
2	4052	345	96	0	372
4	220	3165	275	0	37
3	128	661	1582	0	34
-9	0	0	0	3605	0
1	528	110	15	0	1868

Correct classified: 14,272 Wrong classified: 2,821

Accuracy: 83.496 % Error: 16.504 %

Cohen's kappa (κ) 0.789

Figure 3.5, Confusion matrix for random forest classification.



Approxima…	4	-9	2	3	1
4	18409	11	27	17383	1383
-9	1436	34364	785	170	364
2	8329	476	9443	4907	24565
3	5636	1	7	15420	2944
1	3115	2	6	848	20891

Correct classified: 98,527 Wrong classified: 72,395

Accuracy: 57.644 % Error: 42.356 %

Cohen's kappa (κ) 0.482

Figure 3.6, Confusion matrix for naïve bayes.

Question 4 – Association Rules

For the purposes of association rule mining (Figure 4.1) the personID and No of hours columns were removed and the data was again converted into a string format to allow easier manipulation (rule engines were considered, however deemed unnecessary). Th association rule learner was configured to only display rules with a confidence and support value of more than 40% this was undertaken to remove rules that are unlikely to be useful.

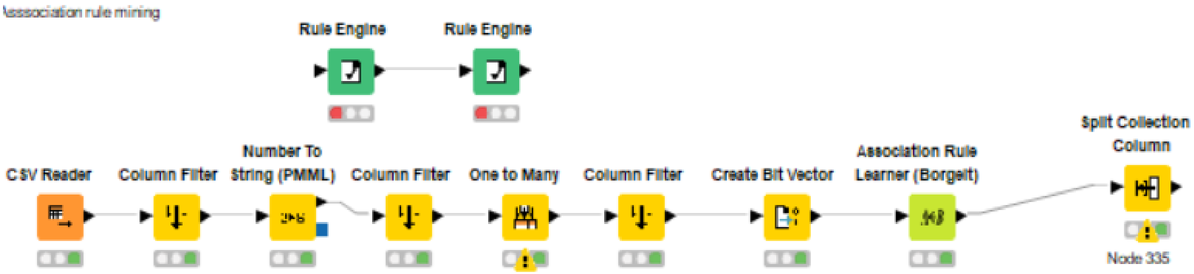
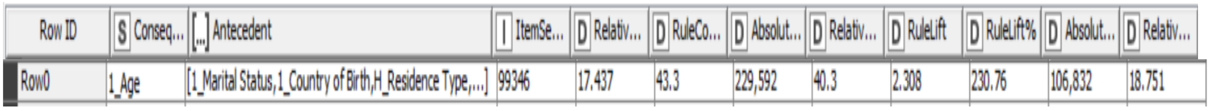


Figure 4.1, The KNIME workflow used.

The following section details 5 rules chosen from those generated and a breakdown of their makeup and related confidence, support and lift as well as a basic example of their use.

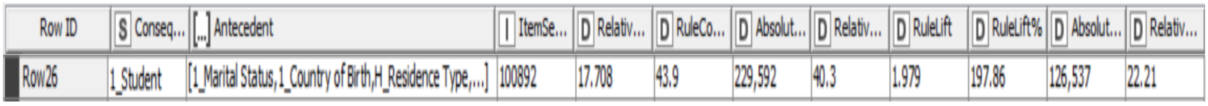
Rule 1 - Row 0



Row ID	S Conseq...	[...] Antecedent	I ItemSe...	D Relativ...	D RuleCo...	D Absolut...	D Relativ...	D RuleLift	D RuleLift%	D Absolut...	D Relativ...
Row0	1_Age	[1_Marital Status,1_Country of Birth,H_Residence Type,...]	99346	17.437	43.3	229,592	40.3	2.308	230.76	106,832	18.751

The consequent for this rule is the 1_age which corresponds to the age range 0-15 years old, the antecedents are 1_marital status (single, never married), 1_country of birth (uk), H_residence type (lives at home) and 1_population base (usual resident). This rule has a relative confidence of 43.3% and a support of 40.3% the ruleLift value for this rule is 230% which is understandable considering that most children will fall within these antecedents.

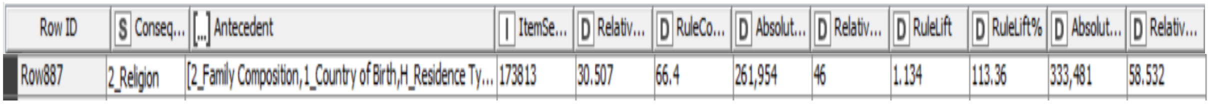
Rule 2 – Row 26



Row ID	S Conseq...	[...] Antecedent	I ItemSe...	D Relativ...	D RuleCo...	D Absolut...	D Relativ...	D RuleLift	D RuleLift%	D Absolut...	D Relativ...
Row26	1_Student	[1_Marital Status,1_Country of Birth,H_Residence Type,...]	100892	17.708	43.9	229,592	40.3	1.979	197.86	126,537	22.21

the consequent for this rule is 1_student (yes). The antecedents are 1_marital status (single, never married), 1_country of birth (uk), H_residence type (lives at home) and 1_population base (usual resident which are identical to the antecedents for the rule before. The relative confidence value of this rule is 43.9% and a support value of 40.3 which is very similar to the previous rule however this rule's lift value is only 197.86% which although still high shows that the first rule has a better correlation between consequents and antecedents than this rule despite them being very similar in support/confidence values.

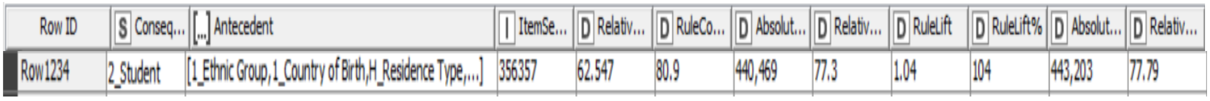
Rule 3 – Row 887



Row ID	S Conseq...	[...] Antecedent	I ItemSe...	D Relativ...	D RuleCo...	D Absolut...	D Relativ...	D RuleLift	D RuleLift%	D Absolut...	D Relativ...
Row887	2_Religion	[2_Family Composition,1_Country of Birth,H_Residence Ty...	173813	30.507	66.4	261,954	46	1.134	113.36	333,481	58.532

The consequent for this rule is 2_religion (Christian) the antecedents are 2_family composition (married/civil partnership), 1_country of birth(uk), H_residence type (lives at home) and 1_population base (usual resident). The confidence for this rule is 66.4% which is higher than other rules thus shown and a support value of 44% which is also slightly higher than previous examples, the lift value for this rule however is only 113.36% which demonstrates that although the support value may be ~20% higher for this rule there is a far loser correlation between the variables used. This also potentially implies family home environments in this dataset are more likely to be Christian.

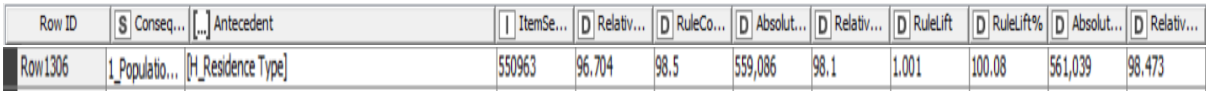
Rule 4 – Row 1234



Row ID	S Conseq...	[...] Antecedent	I ItemSe...	D Relativ...	D RuleCo...	D Absolut...	D Relativ...	D RuleLift	D RuleLift%	D Absolut...	D Relativ...
Row1234	2_Student	[1_Ethnic Group,1_Country of Birth,H_Residence Type,...]	356357	62.547	80.9	440,469	77.3	1.04	104	443,203	77.79

The consequent for this rule is 2_student (No), the antecedents are 1_ethnic group (white) 1_country of birth(uk), H_residence type (lives at home) and 1_population base (usual resident). This rule has a relative confidence value of 80.9% and a support value of 77.4% and a lift value of 104% these values show that this rule is accurate, occurs often and there is a large correlation between the antecedents.

Rule 5 – Row 1306



Row ID	S Conseq...	[...] Antecedent	I ItemSe...	D Relativ...	D RuleCo...	D Absolut...	D Relativ...	D RuleLift	D RuleLift%	D Absolut...	D Relativ...
Row1306	1_Populatio...	[H_Residence Type]	550963	96.704	98.5	559,086	98.1	1.001	100.08	561,039	98.473

The consequent for this rule is 1_popualtion base (usual resident), the antecedent is H_residence type (lives at home). This rule has a relative confidence of 98.5% and a support value of 98.1 it also has a lift value of 100.08%. This rule is very simple and a subset of the more complex rules meaning that it is based on this dataset the 2nd most accurate rule presented.

Question 5 – Regression

For the purposes of regression, the chosen algorithms were linear regression and regression trees. For both algorithms, the personID column was removed and rows with missing numerical values were also removed so that only the rows that feature a No of hours value are contained within the algorithm. Both workflows had relative partition sizes of 70% for fair comparison. The following figures show the workflows used.

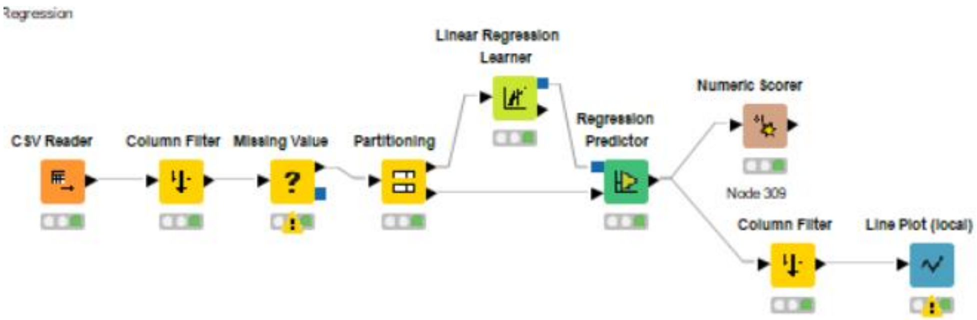


Figure 5.1, Linear regression.

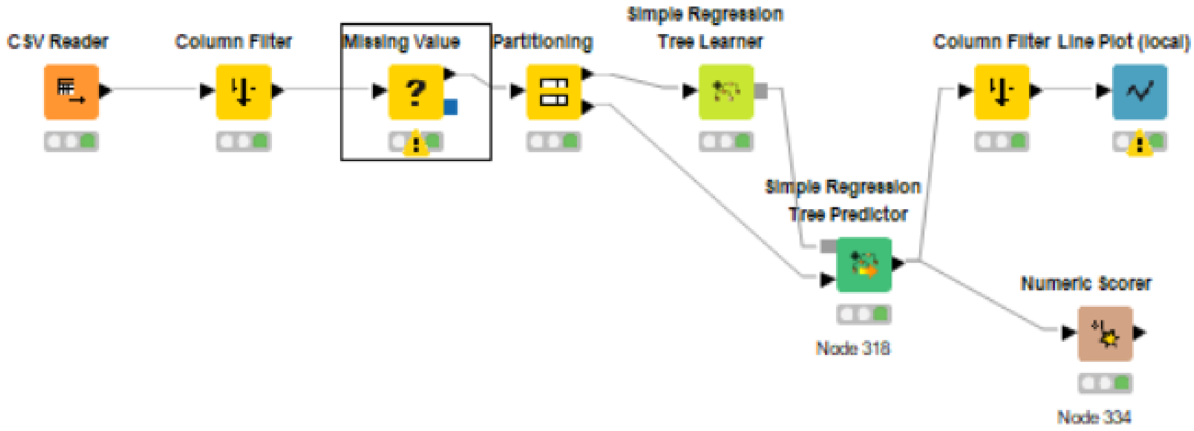
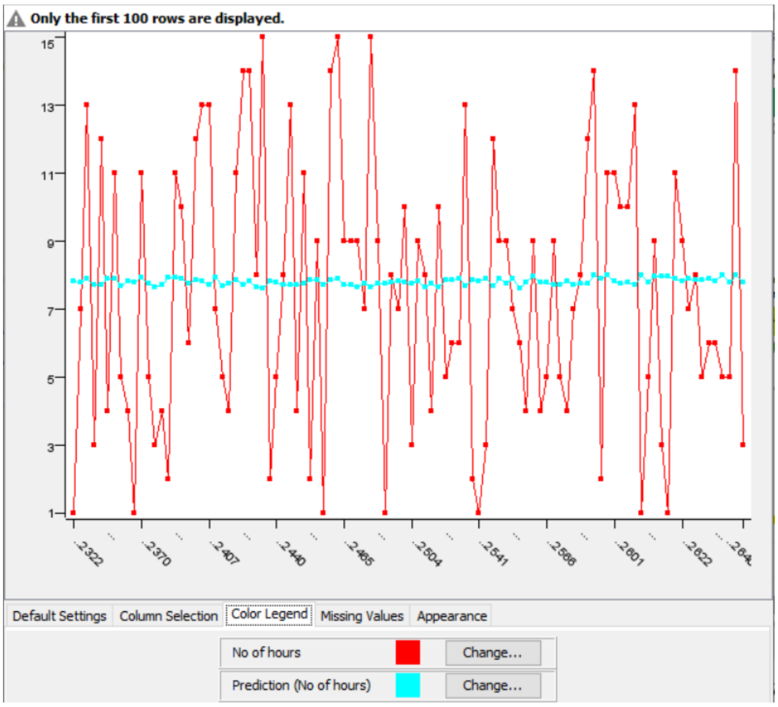


Figure 5.2, Regression tree.

The following figures show the results of both algorithms chosen for the question.

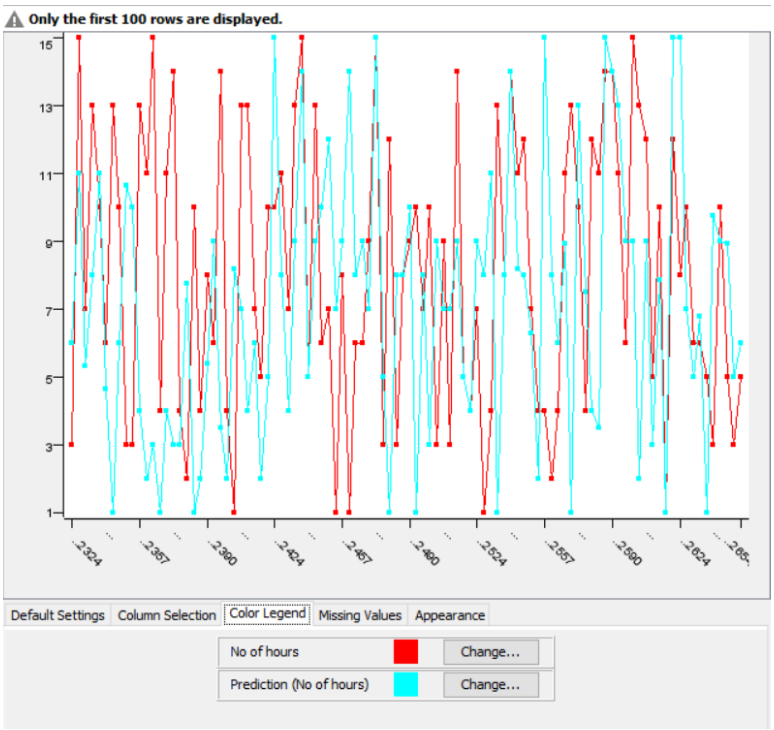


Only the first 100 rows are displayed.

No of hours 🟥

Prediction (No of hours) 🟦

Figure 5.3, Linear regression line plot



Only the first 100 rows are displayed.

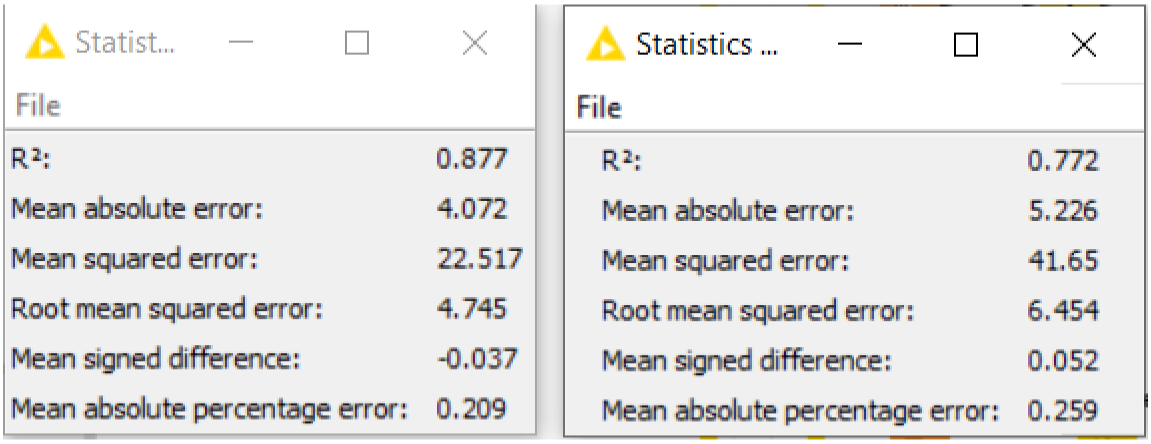
No of hours 🟥

Prediction (No of hours) 🟦

Figure 5.4, Regression tree line plot.

Comparison

Both algorithms used provided showed reasonable predictions with the effects dramatically more noticeable visually with the regression tree algorithm however upon inspection of the statistic via the numeric scorer node linear regression performed significantly better on this dataset and chosen column (Figure 5.5).



R²:	0.877
Mean absolute error:	4.072
Mean squared error:	22.517
Root mean squared error:	4.745
Mean signed difference:	-0.037
Mean absolute percentage error:	0.209

R²:	0.772
Mean absolute error:	5.226
Mean squared error:	41.65
Root mean squared error:	6.454
Mean signed difference:	0.052
Mean absolute percentage error:	0.259

Figure 5.5, Linear regression shown on the left with regression tree on the right.

Part 2

The yelp dataset was chosen for use in this part of the report.

Question 1 – text pre-processing

The below image (Figure 1.1) shows the workflow used for this question, there are not many ways to carry out this process and most discovered online were far less extensive than the ones created in the tutorial, so the lemmatizing workflow was used as this is more accurate than stemming. For each node used its purpose is described below in the order they appear within the workflow.

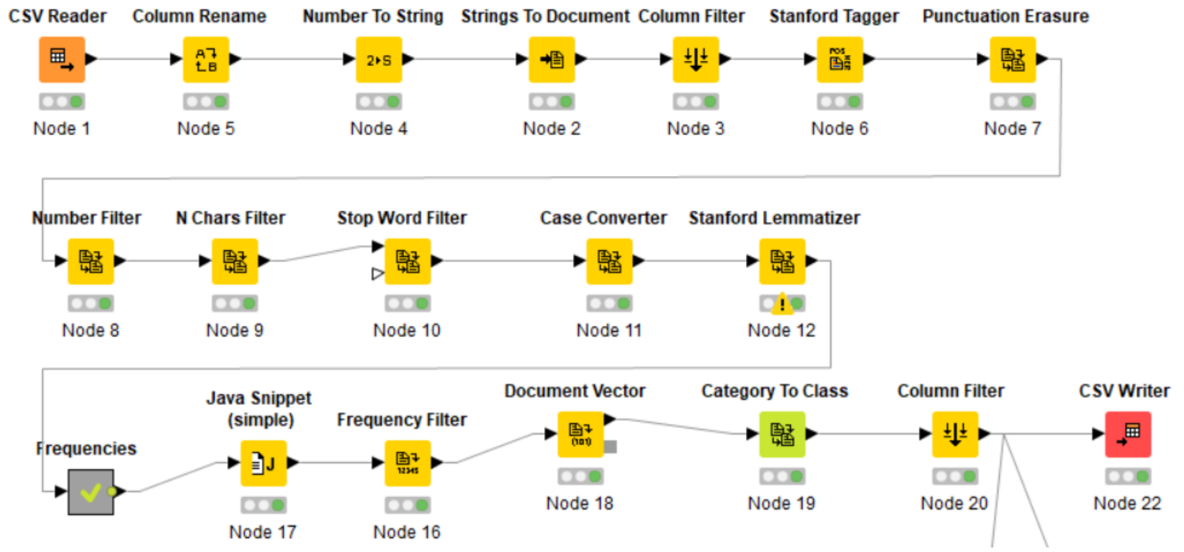


Figure 1.1, The KNIME workflow used for pre-processing.

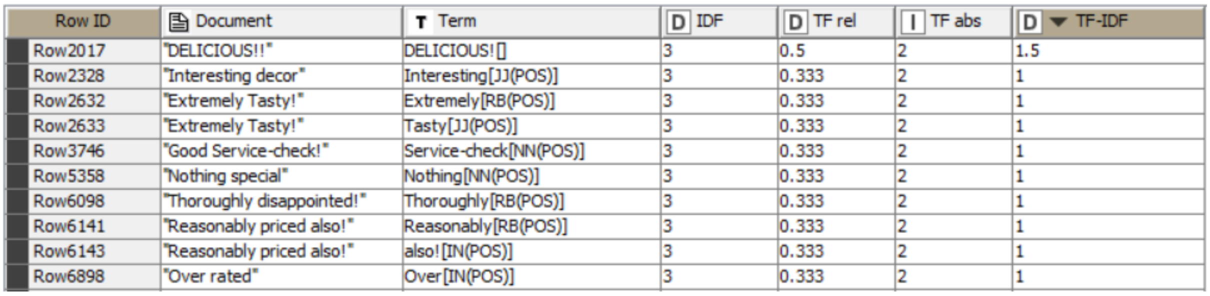
- The first step in pre-processing the .txt file was to change the column delimiter to \t and remove the quote character, as there are no column or row headers these checkboxes in the csv reader were also unselected allowing the .txt file to be read as a .csv file.
- For simplicity and readability col0 was renamed to text and col1 renamed to sentiment, the sentiment column was then changed from numeric type to string, a column filter was then used to filter the document column.
- Natural language processing stage, Stanford tagger used for POS tagging.
- Punctuation was then removed from the document to remove noise.
- Number filter applied to remove any decimal separators or mathematical signs. An initial scan of the data shows that this step may be unnecessary for this dataset although it does no harm to include it in the workflow.

- N char filter applied to remove words with less than 3 characters, most of these words are of little use in the context of the document.
- English stop words removed. These are words considered to have little context.
- All characters changed to lowercase to ensure uniformity.
- Lemmatization of the document – words transformed into their basic meaning i.e., Walked -> walk.
- bag of words performed to create the simplified list of tags (contained within frequencies node).
- Frequencies node used to calculate inverse document frequency and add a column containing the calculation.
- Java snippet used to calculate the TF-IDF (term frequency–inverse document frequency) which is used to determine how important a word is in a document or collection of documents.
- Frequency filter applied to TF-IDF column filtering by the number of terms.
- Document vector created and class column added.
- Column filter applied to only display the calculated tags and the document class column.
- CSV writer used to create new document containing the tagged terms.

Question 2 - basic analytics

The following table shows the top 10 terms based on their TF-IDF and contains other relevant data such as the original document the term was derived from. As can be seen from the table only the top term is unique with the remaining 9 terms identical in all metrics. Due to this factor the report will largely focus on what insight can be gained from the data and whether the process has been effective.

Table 2.1, The top 10 terms.



Row ID	Document	Term	IDF	TF rel	TF abs	▼ TF-IDF
Row2017	"DELICIOUS!!"	DELICIOUS![]	3	0.5	2	1.5
Row2328	"Interesting decor"	Interesting[JJ(POS)]	3	0.333	2	1
Row2632	"Extremely Tasty!"	Extremely[RB(POS)]	3	0.333	2	1
Row2633	"Extremely Tasty!"	Tasty[JJ(POS)]	3	0.333	2	1
Row3746	"Good Service-check!"	Service-check[NN(POS)]	3	0.333	2	1
Row5358	"Nothing special"	Nothing[NN(POS)]	3	0.333	2	1
Row6098	"Thoroughly disappointed!"	Thoroughly[RB(POS)]	3	0.333	2	1
Row6141	"Reasonably priced also!"	Reasonably[RB(POS)]	3	0.333	2	1
Row6143	"Reasonably priced also!"	also![IN(POS)]	3	0.333	2	1
Row6898	"Over rated"	Over[IN(POS)]	3	0.333	2	1

In terms of the insight gained from these terms first and foremost it can be seen that the term “DELICIOUS!” is the most frequent term within the document based on the TF-IDF metric and also the TF relative metric and that there are several terms that give very little context with regards to the sentiment expressed such as “Extremely”, “Nothing” “Thoroughly”, “also!”, “service-check” and “Over” and in the case of these terms the process has removed the original context beyond the point of recognition when compared with the original document text, this reduces the usefulness of the data.

The terms “DELICIOUS!”, “Interesting”, “Tasty” and “Reasonably “are all positive which gives an indication based on this subset that the overall sentiment of the document is positive however, it would have been beneficial to have possibly removed any adjectives which are indicated by the JJ POS(Part-of -speech) tag and also any preposition or subordinating conjunctions indicated by the IN POS tag to ensure that more meaningful tags are retained as in its present state more than 50% of the tags are meaningless in regards to their context.

It is also apparent at this point that certain stages in the pre-processing workflow are not functioning as expected such as the case convertor and punctuation remover. This appears to be affecting the quality of the Stanford tagger it can be seen from the table that the term “DELICIOUS!” has not been assigned a POS tag despite being correctly spelt and is likely due to the exclamation mark on the end. This indicates that the text pre-processing workflow is potentially in the wrong order and could benefit from the Stanford tagging being carried out at a later stage of the workflow specifically after the punctuation erasure and case conversion steps.

Question 3 – Clustering

The following Diagram (Figure 3.1) displays the workflow used for this question; it has been connected to the end of the previous pre-processing workflow.

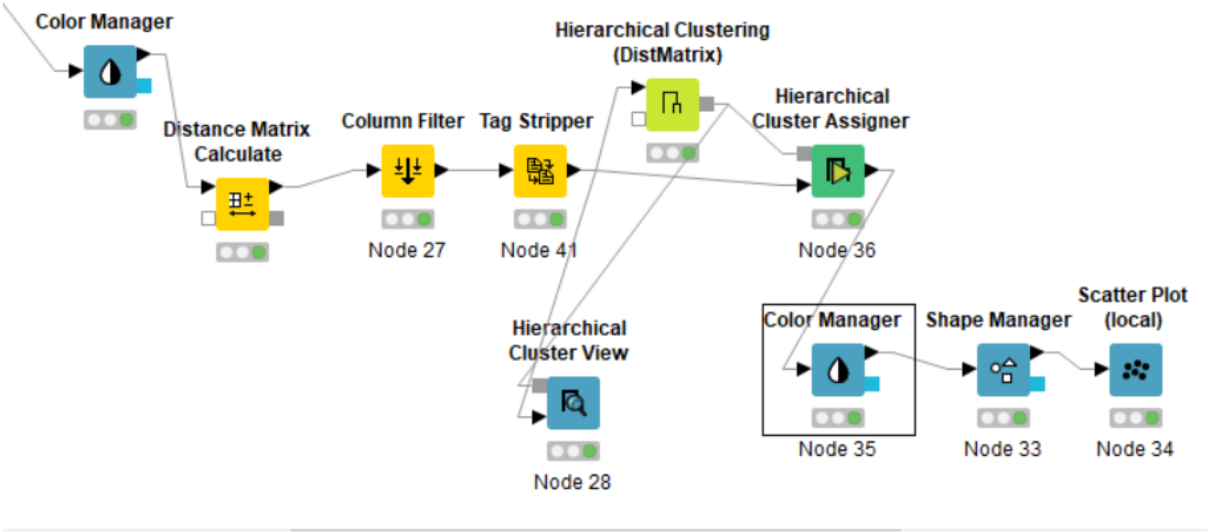
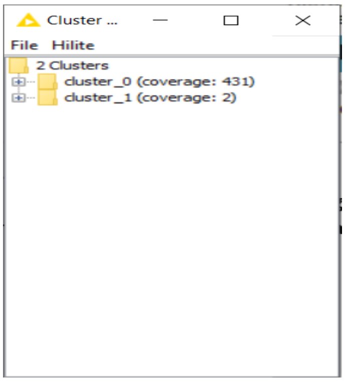


Figure 3.1, The KNIME workflow used.

K-means with K of 2 and K-medoids were attempted however it was difficult to get any sort of meaningful results with any of these as can be seen from the images below with there being a huge difference in the distribution of clusters, for the purposes of answering the question hierarchical clustering has been as used as it gave the best results from the algorithms tested. As can be seen from the image (Figure 3.2) k-means assigned 2 data points to one cluster and the rest to the other, this is not effective clustering.



2 Clusters
cluster_0 (coverage: 431)
cluster_1 (coverage: 2)

Figure 3.2, K-means clusters.

K-medoids partitions, although K-medoids performed slightly better than k-means it still assigned the large majority of the datapoints 415 to the same partition with the remaining 18 datapoints split between 2 other partitions (Table 3.1). This is not effective clustering and k-medoids is not the most suitable clustering algorithm to use on this dataset.

Table 3.1, A table displaying the K-medoids partitions.

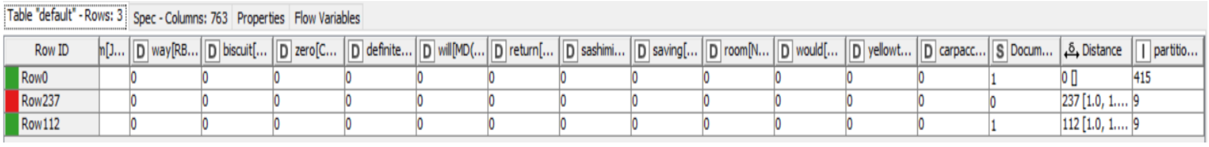


Table "default" - Rows: 3 | Spec - Columns: 763 | Properties | Flow Variables

Row ID	n[J...	D way[RB...	D biscuit[...	D zero[C...	D definite...	D will[MD(...	D return[...	D sashimi...	D saving[...	D room[N...	D would[...	D yellowt...	D carpacc...	S Docum...	Distance	I partitio...
🟩 Row0		0	0	0	0	0	0	0	0	0	0	0	0	1	0 []	415
🟥 Row237		0	0	0	0	0	0	0	0	0	0	0	0	0	237 [1.0, 1....	9
🟩 Row112		0	0	0	0	0	0	0	0	0	0	0	0	1	112 [1.0, 1....	9

The colour manager is used to assign colours to the class labels, in this case red and green are used and the distance. Cosine is used for the distance function however both Euclidean and Manhattan were also considered, the dendrograms shown in the figures below show the difference between the three metrics.

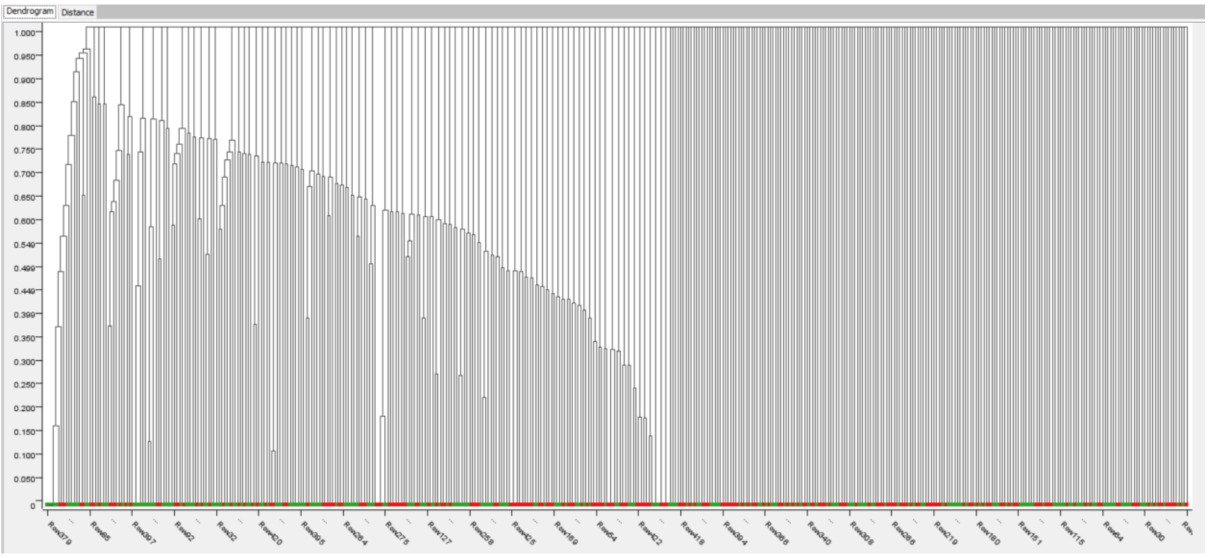


Figure 3.3, Cosine dendrogram.

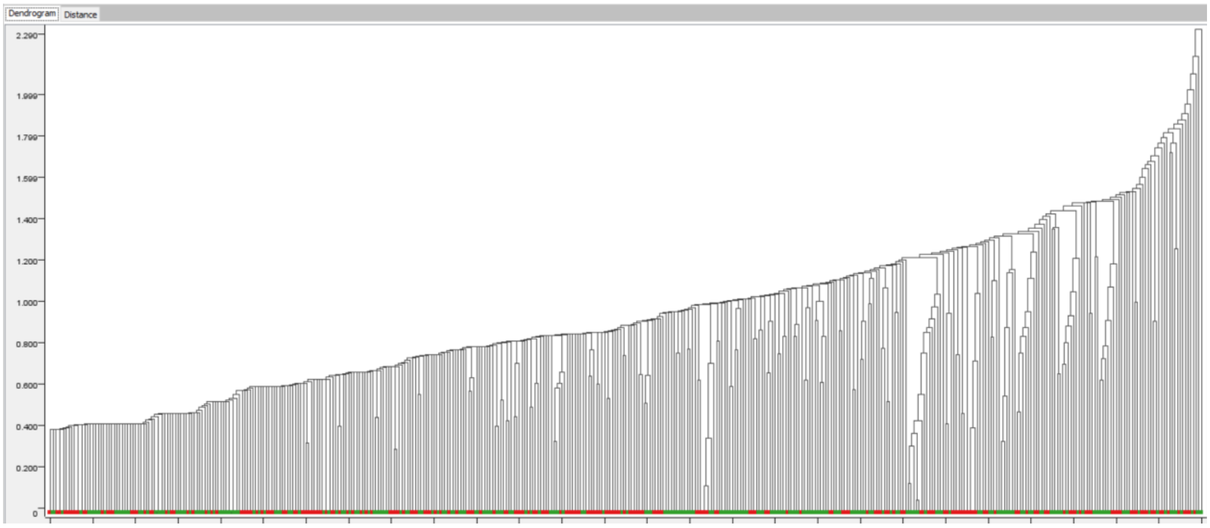


Figure 3.4, Euclidean dendrogram.

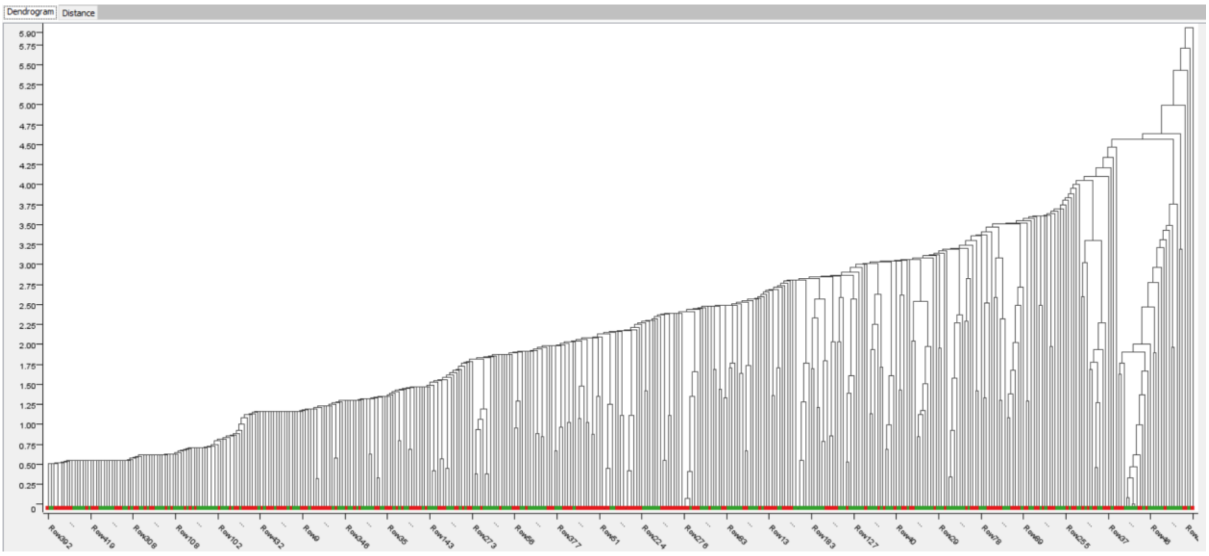


Figure 3.5, Manhattan dendrogram.

Comparison

As can be seen the groupings are better although still not great overall with Cosine used for the distance metric however none of these 3 metrics gave good results. It was actually very difficult to get anything meaningful from clustering with different parameters being used within the hierarchical clustering (distance matrix) node such as the linkage type with the single linkage option giving the best results. the POS tags were removed using the tag stripper node to try and see if this improved results, but the node did not perform as expected and was removed. Some aspects of the pre-processing were also changed or omitted such as the Stanford tagging although the results were not any better than those presented in this report.

Question 4 – Classification

For the purposes of classification, the three techniques presented are decision tree prediction, k-nearest neighbour and SVM (support-vector machine), naïve bayes was also considered however better results were acquired with the other models.

To simplify the workflow space the pre-processing steps from question one were packaged into a metanode and then modified as necessary for each classification technique. For each method of classification various changes were made to the workflow and the effects this had on the classifier's accuracy were tested to allow fine tuning of parameters and then the changes were recorded. ROC curve nodes were not incorporated as part of this solution, this node caused major issues with KNIME and the system overall making them unusable.

Workflows of the 3 methods used.

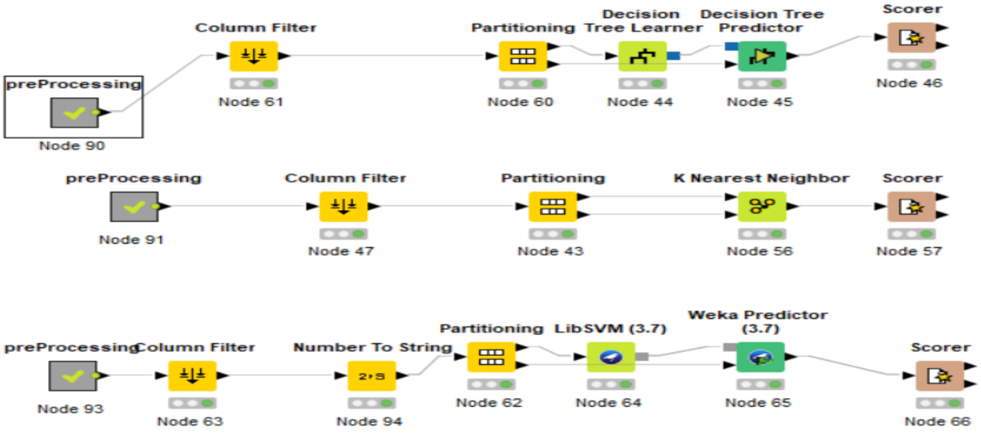
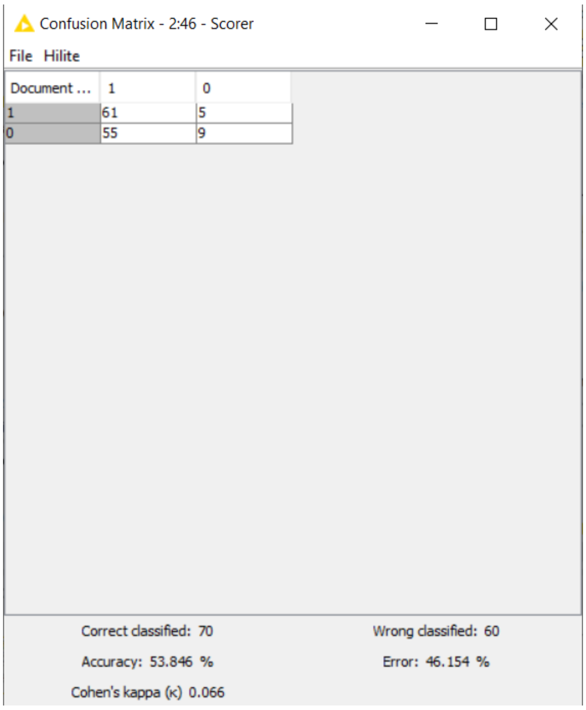


Figure 4.1, KNIME workflows used for text classification.

Decision tree classification

the base accuracy with previous pre-processing from question one was 53.846%.



Document ...	1	0
1	61	5
0	55	9

Correct classified: 70

Wrong classified: 60

Accuracy: 53.846 %

Error: 46.154 %

Cohen's kappa (κ) 0.066

Figure 4.2, Confusion matrix displaying accuracy.

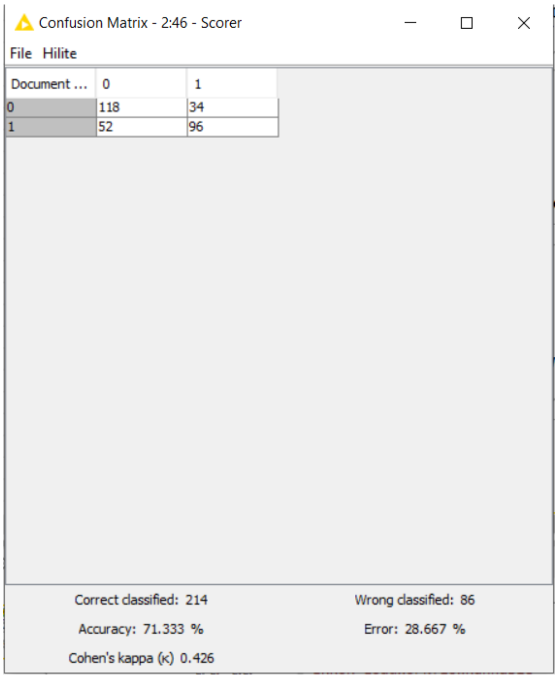
The steps undertaken to improve the model are detailed in the bullet points below,

- Inclusion of tag filtering using the tag filter node filtering adjectives (JJ tag), Preposition or subordinating conjunctions (IN tag) and Nouns, singular or mass (NN tag) this had no noticeable effect.
- Case conversion to uppercase, no effect.
- Use of tag stripper node, no noticeable effect.
- Stanford tagger removed, accuracy dropped to 46%, added back to the workflow. Stanford tagger set to English caseless this had no effect. Changed back to English left 3 words.
- Pruning of the decision tree attempted, this dropped the accuracy to 48%
- Changed number filter to filter terms containing numbers went to 50%, this was changed back to terms representing numbers.
- Second number filter added to filter by both metrics = no effect
- The number of characters used within the N chars node was changed to values between 2 and 7, the optimum value was determined to be 5 characters. This had a noticeable effect increasing the accuracy to 64.3.
- The Frequency filter threshold was changed to filtering by threshold which increased the accuracy to 66.4% and then various values between 0.01 and 1.6 were tested, raising the lower bound decreased the accuracy so this was kept at 0.01, however lowering the upper

bound to 0.6 increased the accuracy to 68.6%, these were kept as the final frequency settings used.

- The Document column was removed by column filter node = 70%
- The model was run 5 times, the best-case accuracy was 71.333%

The best achieved accuracy for the decision tree classification model was 71.333% as displayed in the confusion matrix below.



Confusion Matrix - 2:46 - Scorer

Document ...	0	1
0	118	34
1	52	96

Correct classified: 214

Wrong classified: 86

Accuracy: 71.333 %

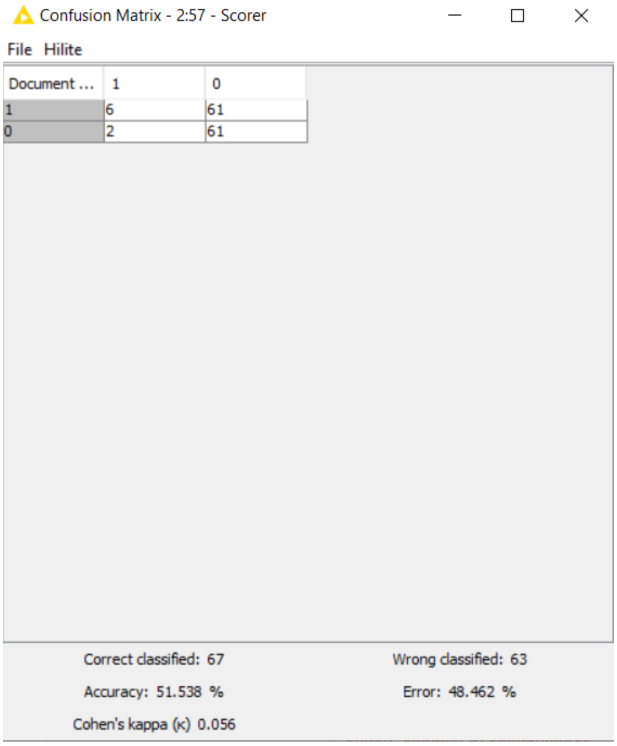
Error: 28.667 %

Cohen's kappa (κ) 0.426

Figure 4.2, Confusion matrix displaying new accuracy.

K-nearest neighbour

The base accuracy of this model after the question one pre-processing was 51.538%.



Document ...	1	0
1	6	61
0	2	61

Correct classified: 67 Wrong classified: 63

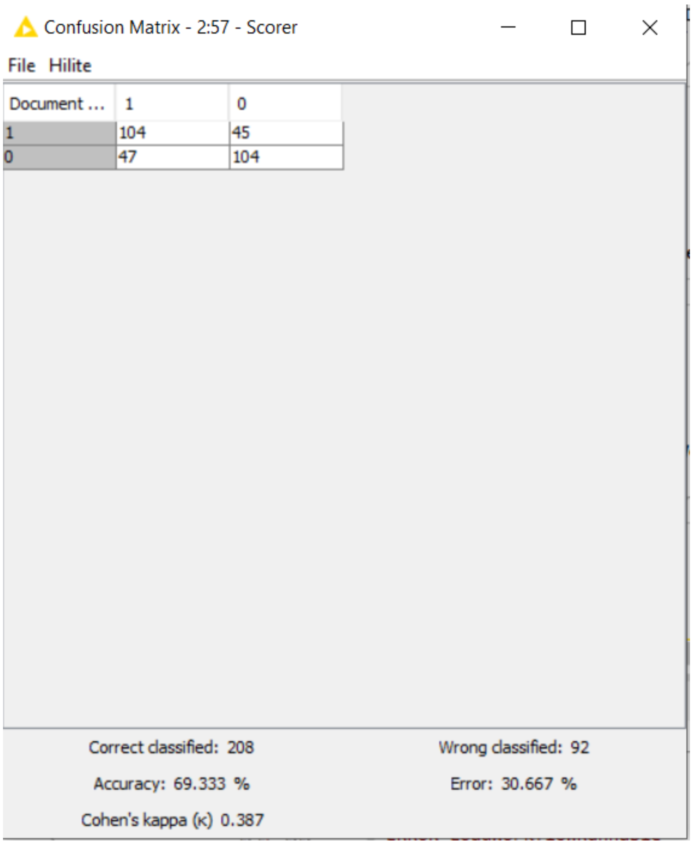
Accuracy: 51.538 % Error: 48.462 %

Cohen's kappa (κ) 0.056

Figure 4.4, Confusion matrix displaying accuracy.

- The Stanford tagger node and Stanford lemmatizer node were removed, the accuracy dropped to 48%, these nodes were reintroduced to the workflow.
- Case conversion to uppercase, no effect.
- The Number filter node was changed to terms containing numbers, this increased the accuracy to 58.4%
- A Second number filter was added to allow filtering by both terms representing and containing numbers this reduced the accuracy and so was removed, and the previous step implemented.
- Various values for the number of character node were tested with the final optimised solution using a value of seven, this improved the accuracy to 62.3%.
- The frequency filter was set to filter by threshold improving accuracy to 68.85% and various values between a lower bound of 0.01 and 1.6 were tested, the optimised rage was between 0.01 and 1.
- The document column was removed increasing the accuracy to 68.2%.
- The model was run five times ending in a best-case accuracy of 69.333%.

The confusion matrix below shows the best achieved accuracy of the k-nearest neighbour model.



Document ...	1	0
1	104	45
0	47	104

Correct classified: 208

Wrong classified: 92

Accuracy: 69.333 %

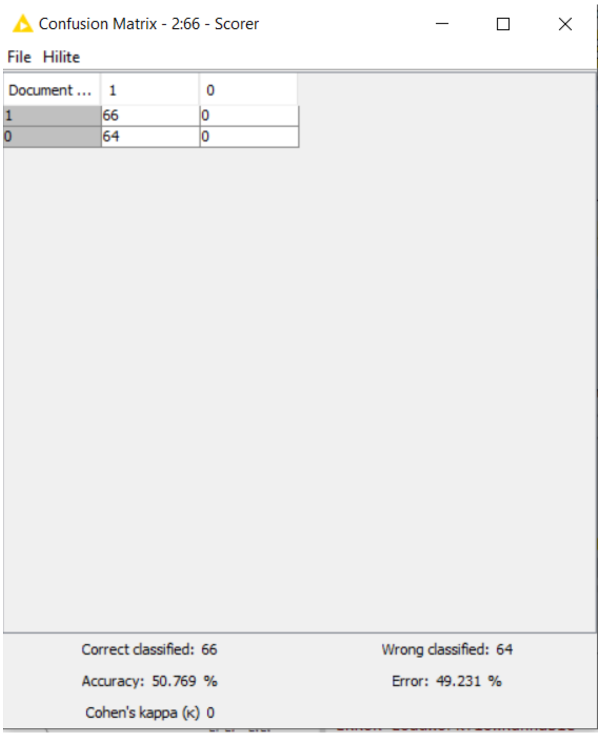
Error: 30.667 %

Cohen's kappa (κ) 0.387

Figure 4.5, Confusion matrix displaying new accuracy.

SVM

The base accuracy of this model after the question one pre-processing was 44.8%.



Document ...	1	0
1	66	0
0	64	0

Correct classified: 66 Wrong classified: 64

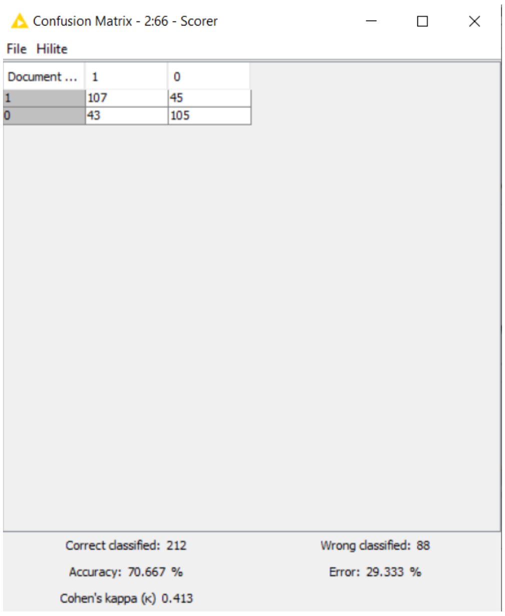
Accuracy: 50.769 % Error: 49.231 %

Cohen's kappa (κ) 0

Figure 4.6, Confusion matrix displaying accuracy.

- The document column was filtered out with the column filter node.
- Removal of Stanford tagger and lemmatizer made no significant difference.
- Addition of Tag stripper and/or tag filter nodes did nothing.
- Changed the numeric type tags to string type, this increased the accuracy to 66.667% using the number to string node.
- Various values were tested for the frequency filter between a lower bound of 0.01 and 1. 6, the final optimal range was between 0.01 and 1.
- Number filter tested with both metrics, filtering by terms representing numbers was more accurate and was used for the model.
- Second number filter node used to filter by both metrics, accuracy dropped so second node removed.
- The number of characters was tested with values between 3 and 7 with the final chosen value being 5, this gave an accuracy of 68.33%.
- The model was run 5 times ending in a best-case accuracy of 70.667%

The confusion matrix below shows the best achieved accuracy with SVG classification.



Document …	1	0
1	107	45
0	43	105

Correct classified: 212 Wrong classified: 88

Accuracy: 70.667 % Error: 29.333 %

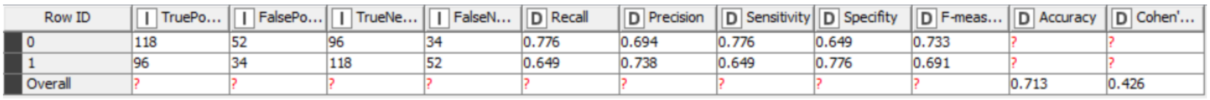
Cohen's kappa (κ) 0.413

Figure 4.7, Confusion matrix displaying new accuracy.

Conclusion

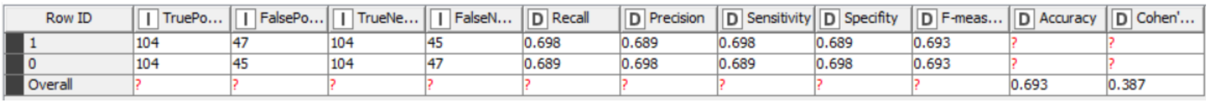
The tables below show a breakdown of the performance of the 3 methods presented.

Table 4.1, Decision tree classification



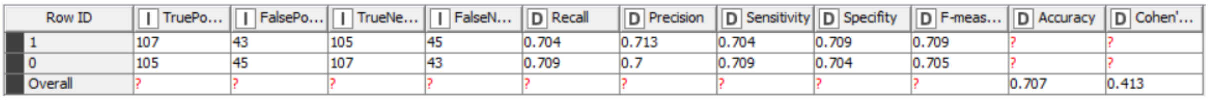
Row ID	TruePo…	FalsePo…	TrueNe…	FalseN…	Recall	Precision	Sensitivity	Specifity	F-meas…	Accuracy	Cohen'…
0	118	52	96	34	0.776	0.694	0.776	0.649	0.733	?	?
1	96	34	118	52	0.649	0.738	0.649	0.776	0.691	?	?
Overall	?	?	?	?	?	?	?	?	?	0.713	0.426

Table 4.2, K-nearest neighbour classification



Row ID	TruePo…	FalsePo…	TrueNe…	FalseN…	Recall	Precision	Sensitivity	Specifity	F-meas…	Accuracy	Cohen'…
1	104	47	104	45	0.698	0.689	0.698	0.689	0.693	?	?
0	104	45	104	47	0.689	0.698	0.689	0.698	0.693	?	?
Overall	?	?	?	?	?	?	?	?	?	0.693	0.387

Table 4.3, SVM classification



Row ID	I TruePo...	I FalsePo...	I TrueNe...	I FalseN...	D Recall	D Precision	D Sensitivity	D Specifity	D F-meas...	D Accuracy	D Cohen'...
1	107	43	105	45	0.704	0.713	0.704	0.709	0.709	?	?
0	105	45	107	43	0.709	0.7	0.709	0.704	0.705	?	?
Overall	?	?	?	?	?	?	?	?	?	0.707	0.413

Comparison

All three models achieved very similar results for metrics such as recall, precision, sensitivity and specifity as well as accuracy once some time was spent testing and adjusting parameters although decision tree classifier was the most accurate. It would have been better to have gotten a higher accuracy ideally higher than 85% however, no matter what was attempted ~70% was the best that could be achieved with the data available and classifiers used. There was very little difference between the three classifiers in terms of performance with all running in less than 30 seconds, however the fact that the K-nearest neighbour model is set to only recognise 7 or more characters means this model may not be of much real use in the real world as lots of words with context are not considered.