RMIT UNIVERSITY MELBOURNE
COSC2111 - DATA MINING

Assignment 2

Udeshika Dissanayake (S3400652)

Udeshika.dissanayake@student.rmit.edu.au

Table of Contents

Part 1: Classification with Neural Networks

Q 1.

The data preprocessing task has been performed using weka. Firstly, the numeric attribute of "Age" is analyzed, and the impossible values (there were only one entry with an impossible value) have been removed. Regarding the missing values, the nominal attribute of "Sex" has 150 missing entries. This is less than 4% of the data set, hence without loss of information generality of the data set, these 150 entries with missing "Sex" value have been removed from the assessment. As far as the numerical attributes are concerned, the missing values have been replaced by the respective mean values using the `ReplaceMissingValues` function. All the nominal attributes have been encoded with dummy encoding using `NominalToBinary` function. In order to get two columns crated for attribute "Sex" (i.e. one for male and other for female), the `(Indices=2, transformAllValues=True)` parameters have been set. For the rest of other nominal attributes, (`No class`) option was used in nominal to binary encoding. All the attributes have been normalized to between 0 – 1 using (`Normalize -S 1.0-T 0.0`) in order to achieve efficient learning and quicker convergence of the neural network. Once the preprocessing is done, the neural network model that will be developed under this question will eventually has 34 inputs & 4 outputs (see Appendix *Figure 2*).

Q 2.

The training, validation, and testing data sets have been generated using `resample` function in weka. The proportionate division was `60-20-20`. That is 60% (2172 instances) of data is for training, and 20% (724 instances) for validation and remaining 20% for testing (725 instances). These generated data sets in weka is in `.arff` format and used a sh script to convert them in to `.pat` files in order to be used in for JavaNNS.

The neural network training should be stopped when the validation error starts to increase. It is important to determine the right time to stop training the neural network, otherwise it will end up being over-trained for the training data, hence called an overfitting model (see Appendix *Figure 3*).

Q 3.

In JavaNNS, the test data set needs to be pointed in the control panel after the neural network is trained. Subsequently, when the data file is saved as `.res` the test data will be used to determine the model. This will also provide the results for the test data. Further, using the ssh command `analyze -s -i test.res`, the statistical information on the test set for `40-20-40` (see Appendix *Figure 5*) can be obtained. Additionally, the `.res` file can be open and check the predicted class labels for each test instances. An example below:

```
0.47312 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0.00037 0
0.16588 0 0.21495 0 0.28502 0 0.27735 0 0 1
0 0 0 0
1 0 0 0
0.99971 0.00055 0.00003 0.00002
```

1Input and output for a test instance for .res file

Expected class label –1 0 0 0 → negative
Predicted class label using 40-20-40– 1 0 0 0 → negative

Q 4.

Run No	Architecture	Parameters	Train Set		Epochs	Test Set	
			MSE	classification Error %		MSE	classification Error %
1	40-20-40	η = 0.2, d_{max} = 0.1	0.088	3.41	100	0.085	3.45
2			0.082	2.90	1000	0.081	2.76
3			0.072	1.52	3000	0.073	1.66
4			0.065	0.51	5000	0.067	0.69
5			0.063	0.32	10000	0.065	0.55

It is evident that the "Train MSE" figure does not considerably decrease as number of Epochs is increasing. This is due to the fact that single-layer "perceptron" models are limited to solve only the linearly separable problems with a binary target (1, 0). This particular example could be a non-linear separable type of problem, hence results (Train MSE & Test MSE) show non convergence (see Appendix *Figure 4*) even if number of Epochs are increased to very high value of 10,000. The parameters used in these trains are `H=0.2` and `dmax = 0.1` and architecture used in result-analyzing is `40-20-40` as shown the result table. (Each run has been initialized with different initial starting weights).

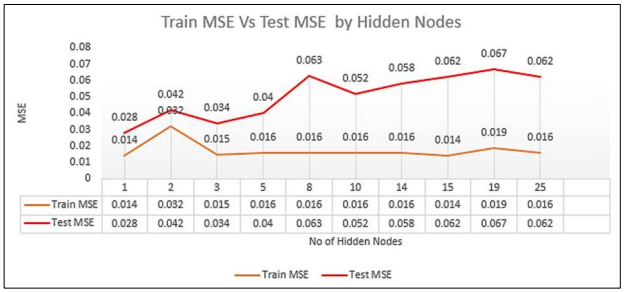
Q 5.

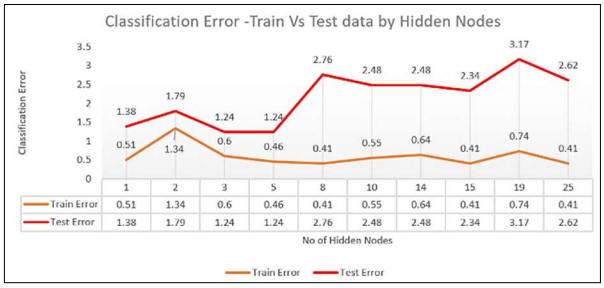
Run No	Architecture	Parameters	Train Set		Epochs	Test Set	
			MSE	classification Error %		MSE	classification Error %
1	40-20-40	η = 0.2, d_{max}=0.1	0.045	1.84	100	0.049	2.21
2			0.042	1.66	1000	0.047	2.07
3			0.028	1.06	4000	0.034	1.38
4			0.031	0.41	6000	0.052	0.69
5			0.014	0.51	20000	0.028	1.38

As can be seen in the result table the "Train MSE" figure is decreasing as the number of Epochs is increasing. Also, it can be observed that the "Test MSE" figure shows an overall downward trend through the initial runs (Run 1 to 3) and then shows a slight upward trend (Run 4). This could be due to overtraining of the model. Generally, the over training is identified when Test/Validation MSE shows upward trend and it is important to stop the model training before that in order to eliminate any overfitting of the model. According to the results, it can be concluded that `Epochs = 4000` is the right level of training for this particular example. Further,

it is evident that the one hidden layer model implemented & tested here shows superior performance results (low Train MSE & Test MSE figures) compared to that of single-layer perceptron model assessed in Q4.

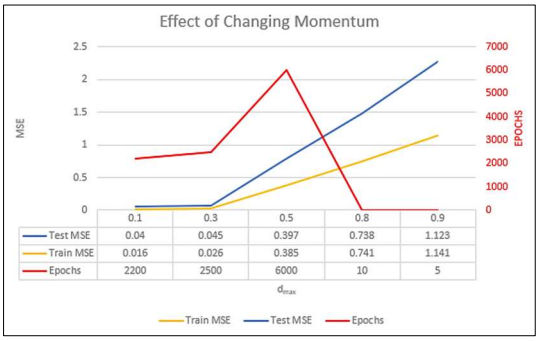
Q 6.

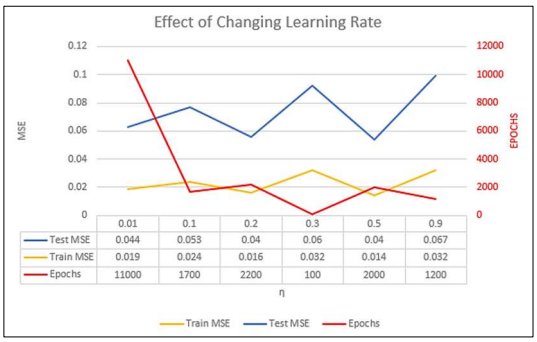




Experiments were conducted ranging the number of hidden layers from 1 to 25 as can be seen in the results figures. Each experiment run was stopped at the respective optimal Epochs number in order to eliminate any overfitting due to Epochs (training cycles). This allows to find the correct number of hidden nodes from the experiment results. Referring to the first figure, it is evident that the "Train MSE" and "Test MSE" curves starts to diverge after `Hidden Nodes = 3`. This means the model starts to overfit beyond 3 hidden nodes. The second figure with classification error rate does show the similar patters too. It is worth noting that results for `Hidden Nodes = 1` is also shows equally good performance (compared to `Hidden Nodes = 3`), however, 1 hidden node model needed a very high level of Epochs number. Considering both the number of Epochs taken and divergence of MSEs, `Hidden Nodes = 3` has been chosen as the right number of hidden nodes for this example. MSE error curves can be seen in *Figure 6* in the Appendix.

Q 7.





The impact of the model parameters of "Momentum" and "Learning Rate" has been investigated here by varying these parameters for `Hidden Nodes = 5` model. It can be observed from the first figure that both Train and Test MSE increases with the Momentum. Also, with the increase of momentum, the number of Epochs needed shows an upward trend initially before dropping to very small number. As far as the "Learning Rate" is concerned, the MSE figures shows an overall flat behavior with some fluctuations. However, the number of Epochs needed is significantly reducing with the Learning Rate. By observing both the figures, it can be concluded that increase of Momentum and Learning Rate contributes for acceleration of training process while compromising the accuracy for some extend.

Q 8.

For this example, the classification accuracy of weka J48 is 99.76% with no overfitting (see Appendix *Figure 7*) while the classification accuracy of JavaNSS (with `Hidden Nodes = 3`) is 98.39% with some low amount of over-fitting (2%). Therefore, it can be stated as Weka J48 works better for this data set.

Comparing these two approaches, the JavaNSS requires a considerably high amount of data preprocessing compared to that of weka J48 where most of the preprocessing is handled by itself (e.g Replace Missing Values, normalization). It is also worth mentioning that weka J48 can be used only for data sets with nominal classes or numerical classes need to be discretized. In comparison, the JavaNSS can be used for any type of data set after completing standards preprocessing steps (data encoding, normalization). In comparison, weka J48 has better explainability for the model.

Q 9.

The comparison between two software programmers is tabulated below:

Weka Multilayer-Perceptron	JavaNNS
Handle all the pre-processing	Lot of pre-processing
No idea what's going on the background	User to handle almost all the tasks
Unable to see validation and error graphs and weka decides when to stop without giving much visibility	can see validation and error graphs and user can decide when to stop
Weka stop training Automatically	No Automatic method to stop training

Overall, the JavaNNS gives better control to the user for training the model while weka performs most of the tasks by itself without giving much visibility to the user. Considering this fact, JavaNNS has been chosen to perform this study.

Part 2: Numeric Prediction with Neural Networks

Q 1.

The data preprocessing task has been performed using weka. There were 14 missing value entries in the data set as far as "thal" and "ca" attributes are concerned. Those missing value entries have been removed without loss of information generality of the data set. Since the task of this exercise is to predict the "chol" attribute, it has been moved to the last column of the data set. All the nominal attributes have been encoded with dummy encoding using `NominalToBinary` function with `No class` option. It is observed that the attribute "ca" is an ordinal variable, hence it has been left as it is without doing any encoding. All the attributes have been normalized to between 0 – 1 using (`Normalize -S 1.0-T 0.0`) in order to achieve efficient learning and quicker convergence of the neural network. Once the preprocessing is done, the neural network model that will be developed under this question will eventually has 23 inputs and 1 outputs (see Appendix *Figure 8*)

Q 2.

The training, validation, and testing data sets have been generated using `resample` function in weka. The proportionate division was `60-20-20`. That is 60% (355 instances) of data is for training, and 20% (188 instances) for validation and remaining 20% for testing (119 instances). These generated data sets in weka is in `.arff` format and used a sh script to convert them in to `.pat` files in order to be used in for JavaNNS.

The "mean-absolute error" is the average of the magnitude (ignoring the +/- sign of the error) of each individual error.

In order to achieve efficient learning and quicker convergence of the neural network, it is recommended to scale the input attribute. This will bring all the input attributes to a same range [0 - 1] and removes the unnecessary dominance of certain attributes due to its value ranging.

The output results from the developed model needs to be "reverse scale" to bring it back to the original scale in order to compare the model output.

Q 3.

Run No	Architecture	Parameters	Train MSE	Epochs	Test MSE
1	40-20-40	H=0.2, d_{max}=0.1	0.0027	100	0.0034
2			0.0027	120	0.0034
3			0.0028	150	0.0035
4			0.0025	190	0.0030
5			0.0025	200	0.0032

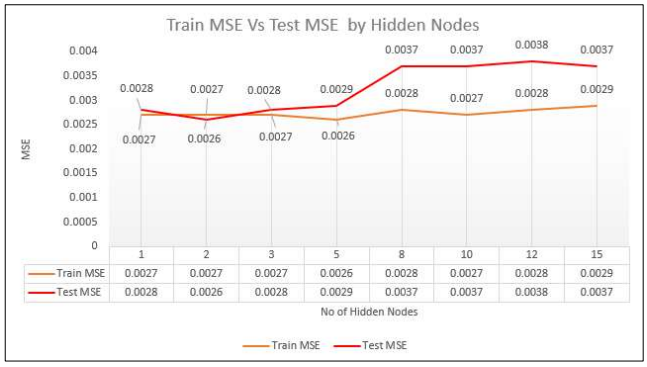
It is evident that the "Train MSE" values are quite low and comparatively similar across all the runs. Also, the "Test MSE" values are low and similar across all the runs. The single-layer "perceptron" models are limited to solve only the linearly separable problems with a binary target (1, 0). This particular example could be a linear separable type of problem, hence results (Train MSE & Test MSE) show convergence across all the runs even for relatively low Epochs numbers. It can be concluded that single single-layer "perceptron" models work fine for this example as Train and Test MSE values have been converged for all the training runs (see Appendix *Figure 9*). The parameters used in these trains are `H=0.2` and `dmax = 0.1` and architecture used in results-analyzing is `40-20-40` as shown the result table. (Each run has been initialized with different initial starting weights).

Q 4.

Run No	Architecture	Parameters	Train MSE	Epochs	Test MSE
1	40-20-40	H=0.2, d_{max}=0.1	0.0027	1000	0.0032
2			0.0027	1200	0.0033
3			0.0026	1800	0.0031
4			0.0026	2000	0.0032
5			0.0026	3000	0.0030

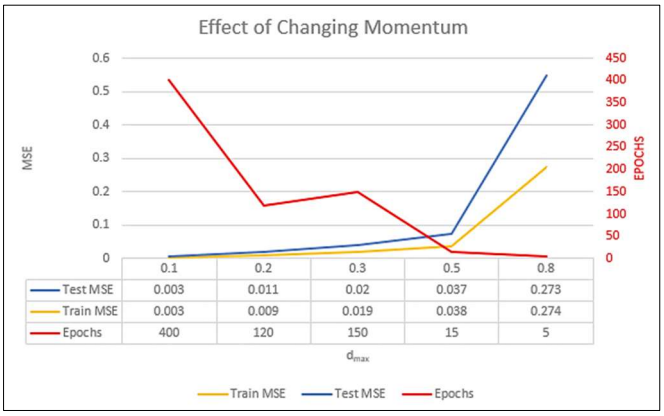
It can be observed that the results across all the train/test runs are almost similar: The "Train MSE" and "Test MSE" values are almost same and does seem to be converged for the respective Epochs numbers (see Appendix *Figure 10*). Also, it is worth mentioning for the provided Epochs numbers, there is no enough evidence to support the overtraining of the model. It can be seen that the Epochs numbers in each run here are relatively high in comparison to those of single-layer "perceptron" models in previous question. This suggests the single-layer "perceptron" models are quickly converging for this particular example.

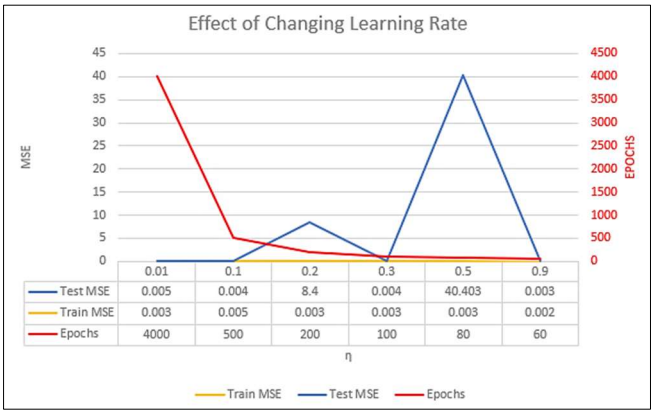
Q 5.



Experiments were conducted ranging the number of hidden layers from 1 to 15 as can be seen in the results figures. Each experiment run was stopped at the respective optimal Epochs number in order to eliminate any overfitting due to Epochs (training cycles). This allows to find the correct number of hidden nodes from the experiment results. Referring to the figure, it is evident that the "Train MSE" and "Test MSE" curves starts to diverge after `Hidden Nodes = 2`. This means the model starts to overfit beyond 3 hidden nodes. Considering both the number of Epochs taken and divergence of MSEs, `Hidden Nodes = 2` has been chosen as the right number of hidden nodes for this example. MSE error curves can be seen in Appendix *Figure 11*.

Q 6.





The impact of the model parameters of "Momentum" and "Learning Rate" has been investigated here by varying these parameters for `Hidden Nodes = 5` model. It can be observed from the first figure that both Train and Test MSE increases with the Momentum. The difference between Train and Test MSE grows as Momentum increases. This suggest that the overfitting is promoted with the increase of Momentum. Also, with the increase of momentum, the number of Epochs required for convergence decreases. As far as the "Learning Rate" is concerned, the MSE figures shows an overall flat behavior with significant fluctuations. However, the number of Epochs needed is significantly reducing with the Learning Rate. By observing both the figures, it can be concluded that increase of Momentum and Learning Rate contributes for acceleration of training process while compromising the accuracy for some extend.

Q 7.

A model training was conducted for `Hidden Nodes = 5` and without checking the validation error behavior to determine the time to stop the training. The model was trained until the "Training MSE" is no longer reducing – 0.003. The classification error at this point was 0% as can be seen in Appendix *Figure 12.* Subsequently, the test data was applied to the model and error rate was assessed. It was noted that the classification error was significantly high---10%---and the "Test MSE" was also relatively high 0.004. This suggests a significant amount of overfitting. Therefore, it is always recommended to decide when to stop the model training based on the validation error.

Q 8.

The "relative-absolute error" represents the sum of the magnitude of errors (ignoring the +/- sign of the error) across each instances. In other words, the sum of the absolute difference between the exact and the model predicted values. In contrast, the "mean-absolute error" is averaging out the "relative-absolute error" across number of instances. Therefore, the "mean absolute error" is more meaningful and preferred when comparing the models as it does not keep accumulating with number of instances, rather averaging out with number of instances.

$$\text{Mean Absolute Error} \quad \frac{\sum_{i=1}^{n} |actual(i) - predicted(i)|}{n}$$

Q 9.

The square of the "root mean square" error (RMS Error) for weka M5P is found to be 0.002 (see Appendix *Figure 13*) without overfitting. The "mean squared error" (MSE) for JavaNSS with 2 hidden nodes is found to be 0.0027 without overfitting. Since the mean-absolute error comparison is mathematically equivalent to the mean square errors, the mean square errors are used here for the comparison. This is also because the mean square errors are readily available in the respective results. It is evident that the weka M5P produces better model for this data set due to the fact that its error figure is lower that that of JavaNSS model.
When comparing JavaNSS Vs weka M5S for numerical classifier tasks, it is noted that JavaNSS requires considerably high amount of data preprocessing compared to weka M5P. Also, weka M5P has better explainability of the model compared to that of JavaNNS. Finally, it is worth noting that M5P is inherently quicker to implement than neural network implementation.

Part 3: Data Mining

1 Introduction

The intention of this study is to apply data mining techniques---classification, clustering, associate finding, and attribute selection---on a movie data set in order to build a model to determine the greatness/success of a movie before it is released in cinema. The movie data set used in this study is collected by IMDb website [1] and repository at kaggle [2]. The original data set has been modified ("genre" information) in order to make the data set more usable for the purpose. The name of the original data set is "IMDB 5000 Movie Dataset" and it contains useful information such as "Budget", "FB Likes" and "IMDb Score" for more than 5000 titles. The data set owner---IMDb---owned by Amazon is claimed to be the world's most popular and authoritative source for movie, TV and celebrity content. All the data mining implementation under this study were conducted in Weka.

2 Data Mining Modelling

The IMDb Score is derived through viewers ratings and votes obtained by IMDb [3]. This score reflects the success or the greatness of a movie. However, the IMDb score for a movie will only start to exist after some time from its release. This study intents to use data mining techniques (Classification, Clustering, and Associate Finding) on 5000+ movie data in order to find useful relation/model to predict IMDb Score of a movie before it releases.

2.1 Hypothesis

Using the features/characteristics on movies the IMDb Score can be predicted before it released in cinema.

2.2 Data Preparation, Preprocessing

The original data set consists of 37 descriptive features and 5043 observations of movies. Since the objective of this exercise is to build a model to predict the IMDb Score before the movie is released, the following attributes have been dropped assuming those have no fair influence on the results: color, director_name, num_critic_for_reviews, actor_2_name, gross, actor_1_name, movie_title, num_voted_users, cast_total_facebook_likes, actor_3_name, num_user_for_reviews, title_year movie_facebook_likes.

For nominal attributes, the missing values were treated by dropping the entire observation record, while for numerical attributes, the missing values were treated by replacing the country mean. For the simplicity of the implementation, the IMDb Score was discretized to three categories: Bad, Average, and Good. It is observed that there were very high number of levels for County (66 Levels) and Language (48 Levels) attributes. The least appearing Countries were replaced by "International" while keeping the most appearing 11 Counties as they are. Similarly, the least appearing Languages were replaced by "Other" while keeping the most appearing 6 Languages as they are (see Appendix *Figure 14*). The final data set contains 4740 attributes and 23 explanatory variable and one target variable with three levels (see Appendix *Figure 15*)

2.3 Data Mining Techniques

Classification

Multiple classifiers---J48, Random Forest, IBK and Naïve Bayes---were trained for the preprocessed data set in order to find the most optimal (highest accuracy with lowest overfitting) classification model. The 10-fold cross validation has been used to train the models. Subsequently, the attribute selection was conducted using `CfsSubsetEval-BestFirst` and `WrapperSubsetEval-BestFirst` methods. The models were re-trained for feature selected data set and assess the performance of the classification models

Attribute selection

The attribute selection has been conducted to optimize the classification models. The `CfsSubsetEval-BestFirst` and `WrapperSubsetEval-BestFirst` methods have been used for attribute selection. Then the classification models were compared for different attribute selected data sets.

Clustering

The K-Mean and EM clustering method have been implemented to find any hidden clusters in the data set. The elbow method was used to find the optimal cluster number (i.e. 5) for K-Mean method. Different parameters and seeds have been used for EM clustering algorithm and assess the performance of each by comparing the cluster results. Further, the different combination of attributes were used in cluster algorithms in order to obtain meaningful result.

Associate finding

All numerical attributes have been discretized to 4 levels using weka and then used `Apriori` package to find any interesting associations between attributes.

2.4 Results

Below table shows the results obtained for different classifiers models implemented on full data set and attribute-selected data sets. For the full data set, the Random Forest (RF) shows 100% train accuracy. However, the test accuracy (68%) suggest that the model is over-trained. When both accuracy and over-training are concerned, it is evident that J48 with `m=7` provides better classifier model for the data set with all attributes (see Appendix *Figure 16*). The results in the table also tells that equally good

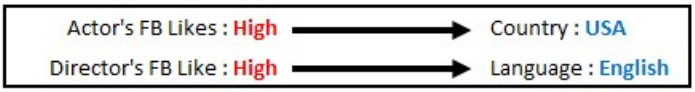
model results could be obtained for J48 with attribute selection. The `Cfs subset-Bestfirst` with `m=2` and `number of attributes = 5` shows `69.1% train accuracy and 66.8% test accuracy for J48. Similarly, Wrapper-Bestfirst with m=7 and number of attributes = 11`, shows 72.9% train accuracy and 68.3% test accuracy. While it is evident that the attribute selection does not show a drastic improve of the model accuracies, it could provide an implementation efficiency due to the smaller number of attributes.

Classifier	Full Data set (All Attributes)			CfsSubsetEval-BestFirst			WrapperSubsetEval-BestFirst		
	Parameters	Train Accuracy	Test Accuracy	Parameters	Train Accuracy	Test Accuracy	Parameters	Train Accuracy	Test Accuracy
J48	m=17	73.9 %	69.3 %	m=2	69.1 %	66.8 %	m=7	72.9 %	68.3 %
RF	iter=150	100 %	72.8 %	iter=100	93.6 %	64.5 %	iter=150	98.2 %	67.8 %
IBK	KNN=250	66.8 %	66.4 %	KNN=150	66.3 %	66.2 %	KNN=150	66.5 %	65.8 %
NB	Use supervise discretization=T	70 %	69.2 %	use supervise discretization=T	67.8 %	67.6 %	use supervise discretization=T	68.9 %	68 %

A meaningful cluster result could be obtained from EM clustering method to predict the goodness of a movie through the number of Facebook (FB) likes of Directors and Actors. It is worth mentioning that K-mean did not produce any meaningful results due to its hard-boundary methodology (see Appendix *Figure 17*). In contrast, the EM method with its soft boundary approach and probabilistic clustering technique, produced a meaningful result as can be seen in below table. (see Appendix *Figure 18*)

Cluster No	FB Likes				Budget	Origin	% Good Movie
	Director	Actor - 1	Actor - 2	Actor - 3			
0	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★	77% USA	38%
1	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★	42% USA	65%
2	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★	85% USA	60%
3	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★	85% USA	58%
4	★★★★★	★★★★★	★★★★★	★★★★★	★★★★★	80% USA	45%

It is evident that the amount of Director's FB likes significantly contributes to the success of a movie. Cluster 2 with `Director FB likes = High`, shows a relatively high (60%) chance of becoming a good movie, while Cluster 0 and 4 with `Director FB likes = Low` shows a low (38% and 45%, respectively) change of being a good movie. It is worth noting that the Cluster 1 shows a contrasting result with `Director FB likes = Low` and high rate (65%) of being a good movie. However, it is evident that most movies in Cluster 1 are non-USA (only 42% USA), therefore it could be assumed that international directors do not get same FB responses as USA/English directors. The Actor FB likes could also use as a measure to predict the goodness of a movie: Cluster 2 and 3 with `Actor-1 FB Likes = High` shows the highest chances of being a good movie. Again, the previous assumption to describe the contrast behavior of Cluster 1 prevails i.e. international Actors are not getting same FB responses as USA/English Actors. The Association finding has been conducted to test this assumption (see Appendix *Figure 19*). In summary, below association findings confirms that the amount of Director/Actor FB likes has clear association with their country and language. In other words, USA and English-speaking Directors/Actors get higher responses (FB likes) from FB compared to their international counterparts.



2.5 Conclusions

Developed a reasonably accurate (70%) classification model with J48 Decision Tree to predict the greatness of a movie before it released in cinema. Using EM Clustering, five meaningful clusters have been identified based on Director/Actor FB likes and greatness of the movie. Finally, through an association finding, the relationship between amount of Director/Actor FB likes with their country of origin is explored.

3 Overall Conclusions

Predicting the greatness or success of a movie based on its characteristics before it gets released in cinema will be important in many aspects in movie industry. Here in this study, different data mining techniques have been applied to explore 5000+ movie records from IMDb and developed a ~70% accurate classifier to predict the greatness of a movie. Also, for USA/English movies, a success predictor through FB likes was built using clustering techniques. An association finding was used to confirm the assumption used in clustering task.

References

[1] "IMDb," [Online]. Available: https://www.imdb.com/. [Accessed 27 Sep 2020].

[2] Yueming, "www.kaggle.com," 2017. [Online]. Available: https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset. [Accessed 27 Sep 2020].

[3] "IMDb," En.wikipedia.org, 2020. [Online]. Available: https://en.wikipedia.org/wiki/IMDb. [Accessed 08 Oct 2020].

[4] Konstantin, "Four Years Remaining » Machine learning," Fouryears.eu, 2020. [Online]. Available: http://fouryears.eu/tags/machine-learning/.

[5] J. Brownlee, "Understand the Impact of Learning Rate on Neural Network Performance," Machine Learning Mastery, 2020. [Online]. Available: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/. [Accessed 27 Sep 2020].

[6] "JavaNNS," Open Open University Wiki, 2020. [Online]. Available: https://openou.fandom.com/wiki/JavaNNS. [Accessed 2020].

Appendix

Part 1: Classification with Neural Networks

Q1:

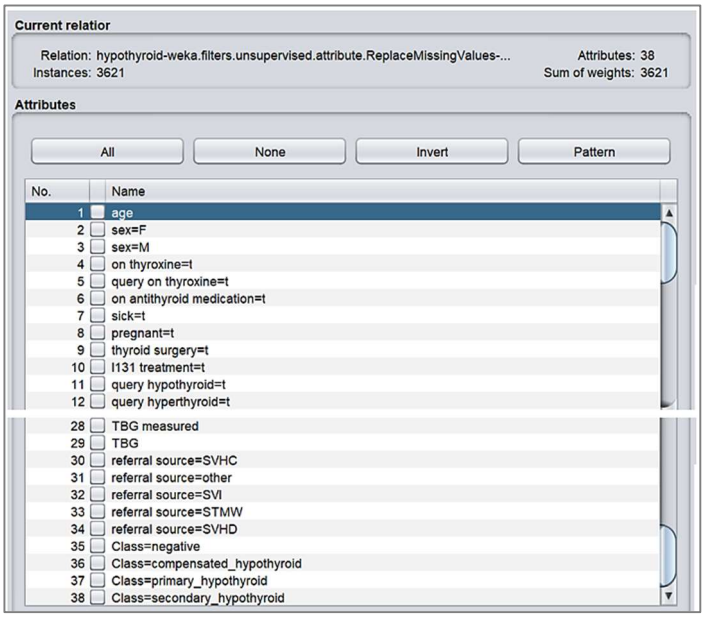


Figure 2: snapshot from preprocessed dataset (34 inputs and 4 outputs)

Q2:

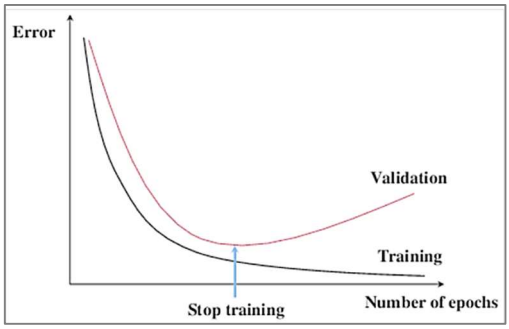
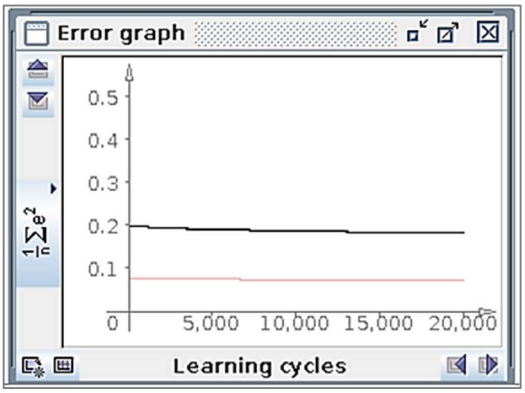


Figure 3: When to stop training a neural network [4]

Q3:

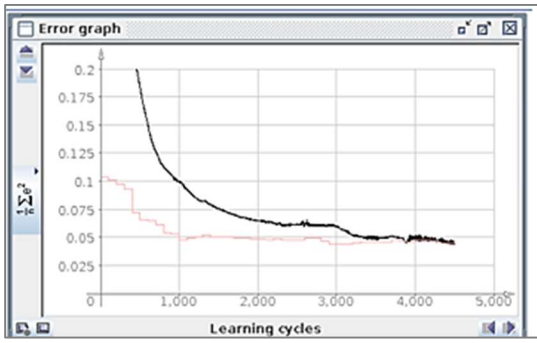



```
[s3400652@csitprdap01 part1Q1]$ analyze -s -i 5000-test.res
STATISTICS ( 725 patterns )
wrong   :  0.69 %  ( 5 pattern(s) )
right   : 94.21 %  ( 683 pattern(s) )
unknown :  5.10 %  ( 37 pattern(s) )
error   : 48.446785
[s3400652@csitprdap01 part1Q1]$ analyze -s -i 5000-train.res
STATISTICS ( 2172 patterns )
wrong   :  0.51 %  ( 11 pattern(s) )
right   : 93.55 %  ( 2032 pattern(s) )
unknown :  5.94 %  ( 129 pattern(s) )
error   : 141.584198
[s3400652@csitprdap01 part1Q1]$
```

Figure 4: Analyzing results using "analyze -s -i test.res"

Figure 5: Not converging to zero when there are no hidden layers

Q6:




```
[s3400652@csitprdap01 part1Q6]$ analyze -s -i 3-4500-train.res
STATISTICS ( 2172 patterns )
wrong   :  0.60 %  ( 13 pattern(s) )
right   : 98.39 %  ( 2137 pattern(s) )
unknown :  1.01 %  ( 22 pattern(s) )
error   : 33.398926
[s3400652@csitprdap01 part1Q6]$ analyze -s -i 3-4500-test.res
STATISTICS ( 725 patterns )
wrong   :  1.24 %  ( 9 pattern(s) )
right   : 97.38 %  ( 706 pattern(s) )
unknown :  1.38 %  ( 10 pattern(s) )
error   : 24.915689
```

Figure 6: Results for JavaNNS for 3 hidden nodes

Q8:

```
Correctly Classified Instances        1228               99.7563 %
Incorrectly Classified Instances         3                0.2437 %
Kappa statistic                          0.9833
Mean absolute error                      0.0019
Root mean squared error                  0.0348
Relative absolute error                  2.566  %
Root relative squared error             18.3093 %
Total Number of Instances             1231

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.998     0.000     1.000      0.998     0.999     0.989     0.999     1.000     negative
               1.000     0.001     0.986      1.000     0.993     0.993     1.000     0.986     compensated_hypothyroid
               0.960     0.002     0.923      0.960     0.941     0.940     0.998     0.896     primary_hypothyroid
               ?         0.000     ?          ?         ?         ?         ?         ?         secondary_hypothyroid
Weighted Avg.  0.998     0.000     0.998      0.998     0.998     0.988     0.999     0.997

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1134    0    2    0 |    a = negative
    0   70    0    0 |    b = compensated_hypothyroid
    0    1   24    0 |    c = primary_hypothyroid
    0    0    0    0 |    d = secondary_hypothyroid
```

Figure 7: Results for Weka J48 with 10-fold cross validation

Part 2: Numeric Prediction with Neural Networks

Q1:

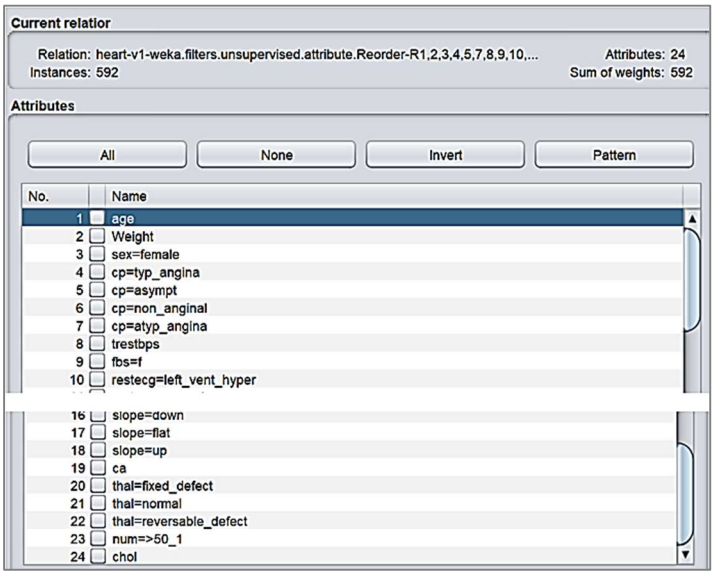


Figure 8: snapshot from preprocessed dataset (23 inputs and 1 output)

Q3:

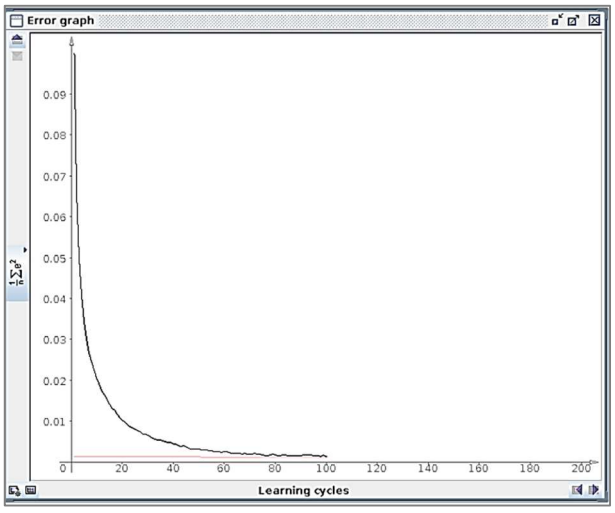


Figure 9: Train and Test MSE values have been converged - No hidden layers

Q4:

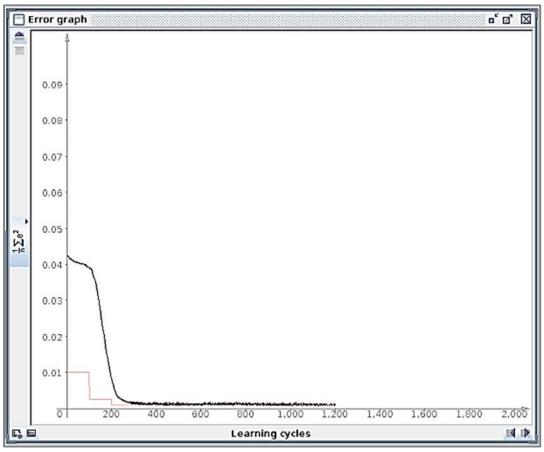


Figure 10: Train and Test MSE values have been converged - One hidden layers

Q5:

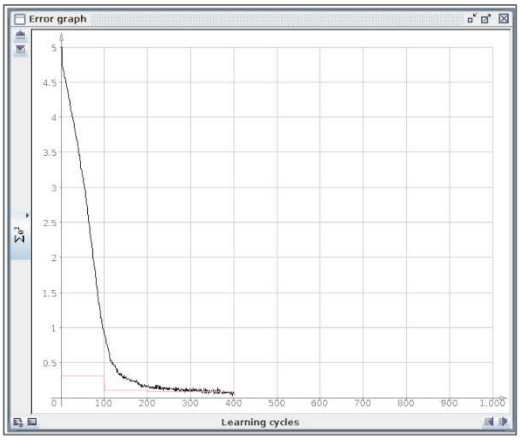


Figure 11: MSE error curves - 2 hidden nodes

Q7:

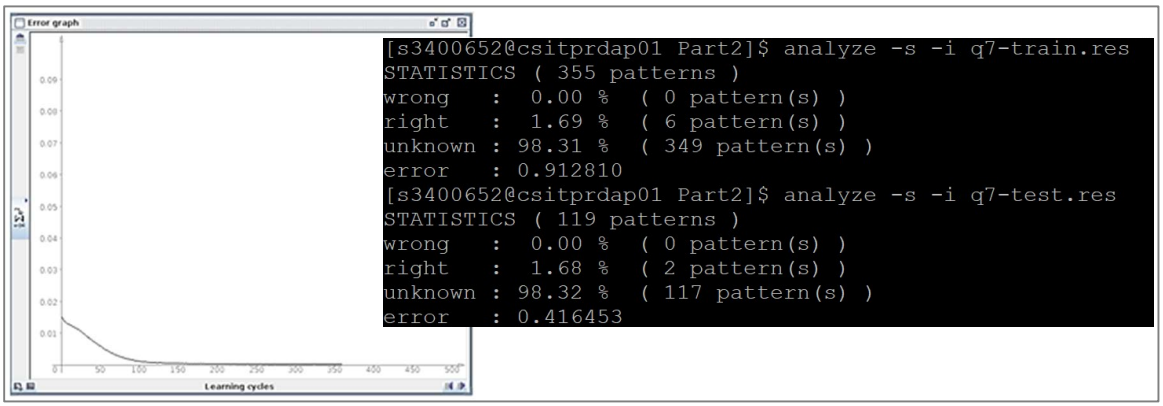


Figure 12: stopping training when the MSE is no longer changing

Q9:

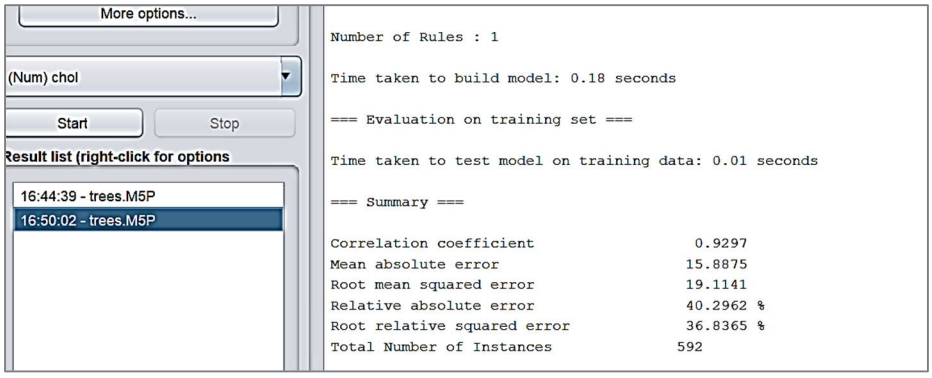
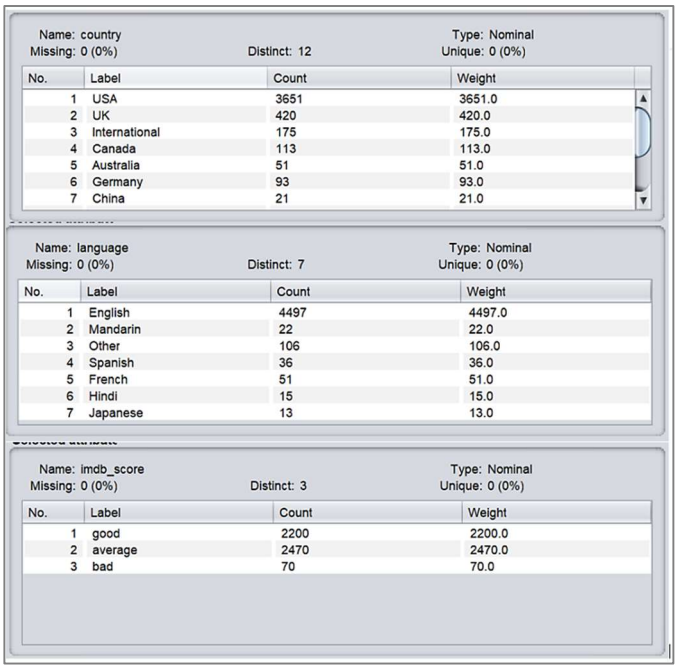


Figure 13: Results from weka M5P

Part 3: Data Mining

2.2 Data Preparation, Preprocessing



Name: country
Missing: 0 (0%) Distinct: 12 Type: Nominal Unique: 0 (0%)

No.	Label	Count	Weight
1	USA	3651	3651.0
2	UK	420	420.0
3	International	175	175.0
4	Canada	113	113.0
5	Australia	51	51.0
6	Germany	93	93.0
7	China	21	21.0

Name: language
Missing: 0 (0%) Distinct: 7 Type: Nominal Unique: 0 (0%)

No.	Label	Count	Weight
1	English	4497	4497.0
2	Mandarin	22	22.0
3	Other	106	106.0
4	Spanish	36	36.0
5	French	51	51.0
6	Hindi	15	15.0
7	Japanese	13	13.0

Name: imdb_score
Missing: 0 (0%) Distinct: 3 Type: Nominal Unique: 0 (0%)

No.	Label	Count	Weight
1	good	2200	2200.0
2	average	2470	2470.0
3	bad	70	70.0

Figure 14: Data Preprocessing - Discretization

Descriptive Features

Descriptive Features	Type
duration	Numeric
director_facebook_likes	Numeric
actor_3_facebook_likes	Numeric
actor_1_facebook_likes	Numeric
Action	Nominal
Adventure	Nominal
Comedy	Nominal
Crime	Nominal
Documentary	Nominal
Drama	Nominal
Family	Nominal
Horror	Nominal
Mystery	Nominal
Romance	Nominal
Sci-Fi	Nominal
Thriller	Nominal
facenumber_in_poster	Numeric
language	Nominal
country	Nominal
content_rating	Nominal
budget	Numeric
actor_2_facebook_likes	Numeric
aspect_ratio	Nominal

Target/Class Feature

$$imdb\ Scor \begin{cases} Bad & \rightarrow\ IMDb\ Score \leq 3.3 \\ Average & \rightarrow\ 3.3 < IMDb\ Score < 6.6 \\ Good & \rightarrow\ IMDb\ Score \geq 6.6 \end{cases}$$

Figure 15: Preprocessed data set

2.4 Results

```
J48 pruned tree
------------------

duration <= 110
|   duration <= 69: good (80.0/7.0)
|   duration > 69
|   |   Documentary = f
|   |   |   Drama = f
|   |   |   |   language = English
|   |   |   |   |   director_facebook_likes <= 3000
|   |   |   |   |   |   budget <= 127500000: average (1515.0/361.0)
|   |   |   |   |   |   budget > 127500000
|   |   |   |   |   |   |   Action = t: average (28.0/11.0)
|   |   |   |   |   |   |   Action = f: good (34.0/13.0)
|   |   |   |   |   director_facebook_likes > 3000: good (38.0/11.0)
|   |   |   |   language = Mandarin: good (3.0/1.0)
|   |   |   |   language = Other: average (19.0/7.0)
|   |   |   |   language = Spanish: average (6.0/2.0)
|   |   |   |   language = French: good (8.0/2.0)
|   |   |   |   language = Hindi: average (1.0)
|   |   |   |   language = Japanese: good (3.0/1.0)
|   |   |   Drama = t
|   |   |   |   language = English
|   |   |   |   |   Horror = f
|   |   |   |   |   |   Action = t: average (134.0/35.0)
|   |   |   |   |   |   Action = f
|   |   |   |   |   |   |   country = USA
|   |   |   |   |   |   |   |   actor_1_facebook_likes <= 15000
|   |   |   |   |   |   |   |   |   Sci-Fi = t: good (26.0/7.0)
|   |   |   |   |   |   |   |   |   Sci-Fi = f
|   |   |   |   |   |   |   |   |   |   budget <= 11500000
|   |   |   |   |   |   |   |   |   |   |   content_rating = PG-13
|   |   |   |   |   |   |   |   |   |   |   |   director_facebook_likes <= 53: average (40.0/16.0)
|   |   |   |   |   |   |   |   |   |   |   |   director_facebook_likes > 53: good (18.0/4.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = PG: average (27.0/10.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = G: good (3.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = R
|   |   |   |   |   |   |   |   |   |   |   |   duration <= 99
|   |   |   |   |   |   |   |   |   |   |   |   |   Thriller = f
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Romance = f: average (67.0/27.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Romance = t
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   actor_3_facebook_likes <= 224: average (1
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   actor_3_facebook_likes > 224: good (17.0/
|   |   |   |   |   |   |   |   |   |   |   |   |   Thriller = t: average (25.0/7.0)
|   |   |   |   |   |   |   |   |   |   |   |   duration > 99: good (75.0/23.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = TV-14: good (0.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = TV-PG: average (3.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = TV-MA: good (0.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = TV-G: average (1.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = Unrated: good (17.0/7.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = Approved: good (5.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = TV-Y: good (0.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = NC-17: good (1.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = X: average (3.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = TV-Y7: good (0.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = GP: good (0.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = Passed: good (1.0)
|   |   |   |   |   |   |   |   |   |   |   content_rating = M: good (1.0)
|   |   |   |   |   |   |   |   |   |   budget > 11500000
|   |   |   |   |   |   |   |   |   |   |   facenumber_in_poster <= 4
|   |   |   |   |   |   |   |   |   |   |   |   Family = f
|   |   |   |   |   |   |   |   |   |   |   |   |   Romance = f
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Thriller = f
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   duration <= 106: average (64.0/23.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   duration > 106: good (22.0/7.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Thriller = t: average (51.0/12.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   Romance = t: average (96.0/24.0)
|   |   |   |   |   |   |   |   |   |   |   |   Family = t
|   |   |   |   |   |   |   |   |   |   |   |   |   Comedy = f: good (21.0/6.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   Comedy = t: average (24.0/7.0)
|   |   |   |   |   |   |   |   |   |   |   facenumber_in_poster > 4: average (17.0)
|   |   |   |   |   |   |   |   |   actor_1_facebook_likes > 15000: good (103.0/33.0)
|   |   |   |   |   |   |   country = UK: good (110.0/29.0)
|   |   |   |   |   |   |   country = International: good (15.0/6.0)
|   |   |   |   |   |   |   country = Canada: good (26.0/13.0)
|   |   |   |   |   |   |   country = Australia: good (10.0/5.0)
|   |   |   |   |   |   |   country = Germany: average (10.0/3.0)
|   |   |   |   |   |   |   country = China: average (1.0)
|   |   |   |   |   |   |   country = France: good (14.0/5.0)
|   |   |   |   |   |   |   country = Japan: good (0.0)
|   |   |   |   |   |   |   country = Spain: average (9.0/4.0)
|   |   |   |   |   |   |   country = India: good (0.0)
|   |   |   |   |   |   |   country = Italy: average (3.0/1.0)
|   |   |   |   |   Horror = t: average (71.0/12.0)
|   |   |   |   language = Mandarin: average (7.0/1.0)
|   |   |   |   language = Other: good (30.0/4.0)
|   |   |   |   language = Spanish: good (15.0/2.0)
|   |   |   |   language = French: good (16.0/7.0)
|   |   |   |   language = Hindi: good (2.0)
|   |   |   |   language = Japanese: average (2.0)
|   |   Documentary = t: good (64.0/12.0)
```

```
duration > 110
|   language = English
|   |   duration <= 132
|   |   |   budget <= 31115000
|   |   |   |   facenumber_in_poster <= 3: good (508.0/143.0)
|   |   |   |   facenumber_in_poster > 3
|   |   |   |   |   content_rating = PG-13: average (23.0/6.0)
|   |   |   |   |   content_rating = PG: good (6.0/2.0)
|   |   |   |   |   content_rating = G: average (0.0)
|   |   |   |   |   content_rating = R: good (35.0/15.0)
|   |   |   |   |   content_rating = TV-14: average (0.0)
|   |   |   |   |   content_rating = TV-PG: average (0.0)
|   |   |   |   |   content_rating = TV-MA: average (0.0)
|   |   |   |   |   content_rating = TV-G: average (1.0)
|   |   |   |   |   content_rating = Unrated: good (3.0/1.0)
|   |   |   |   |   content_rating = Approved: good (2.0)
|   |   |   |   |   content_rating = TV-Y: average (0.0)
|   |   |   |   |   content_rating = NC-17: average (0.0)
|   |   |   |   |   content_rating = X: average (0.0)
|   |   |   |   |   content_rating = TV-Y7: average (0.0)
|   |   |   |   |   content_rating = GP: average (0.0)
|   |   |   |   |   content_rating = Passed: good (1.0)
|   |   |   |   |   content_rating = M: average (0.0)
|   |   |   budget > 31115000
|   |   |   |   actor_2_facebook_likes <= 366
|   |   |   |   |   facenumber_in_poster <= 0: good (27.0/8.0)
|   |   |   |   |   facenumber_in_poster > 0
|   |   |   |   |   |   actor_1_facebook_likes <= 550: average (21.0/7.0)
|   |   |   |   |   |   actor_1_facebook_likes > 550: good (20.0/4.0)
|   |   |   |   actor_2_facebook_likes > 366
|   |   |   |   |   director_facebook_likes <= 800
|   |   |   |   |   |   Drama = f
|   |   |   |   |   |   |   Mystery = f: average (180.0/50.0)
|   |   |   |   |   |   |   Mystery = t
|   |   |   |   |   |   |   |   actor_1_facebook_likes <= 10000: average (18.0/4.0)
|   |   |   |   |   |   |   |   actor_1_facebook_likes > 10000: good (17.0/4.0)
|   |   |   |   |   |   Drama = t
|   |   |   |   |   |   |   Action = t: average (53.0/18.0)
|   |   |   |   |   |   |   Action = f
|   |   |   |   |   |   |   |   Romance = f
|   |   |   |   |   |   |   |   |   Mystery = f: good (87.0/31.0)
|   |   |   |   |   |   |   |   |   Mystery = t: average (18.0/8.0)
|   |   |   |   |   |   |   |   Romance = t
|   |   |   |   |   |   |   |   |   facenumber_in_poster <= 1
|   |   |   |   |   |   |   |   |   |   actor_3_facebook_likes <= 651: good (19.0/6.0)
|   |   |   |   |   |   |   |   |   |   actor_3_facebook_likes > 651: average (18.0/6.0)
|   |   |   |   |   |   |   |   |   facenumber_in_poster > 1: average (22.0/5.0)
|   |   |   |   |   director_facebook_likes > 800: good (57.0/17.0)
|   |   duration > 132: good (467.0/100.0)
|   language = Mandarin: good (12.0/2.0)
|   language = Other: good (48.0/5.0)
|   language = Spanish: good (14.0/3.0)
|   language = French: good (24.0)
|   language = Hindi: good (12.0/5.0)
|   language = Japanese: good (8.0/2.0)

Number of Leaves  :     100

Size of the tree :      144
```

Figure 16: Results for J48 - Full data set

```
kMeans
======

Number of iterations: 16
Within cluster sum of squared errors: 1322.0647832602133

Initial starting points (random):

Cluster 0: 26,266,838,English,USA,15000000,328,average
Cluster 1: 125,11,60,Other,Italy,5000000,29,average
Cluster 2: 0,687,11000,English,USA,32000000,1000,good
Cluster 3: 134,488,936,English,USA,1500000,935,average
Cluster 4: 2,137,591,English,USA,22000000,489,average

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute                     Full Data         0             1             2             3             4
                               (4740.0)      (42.0)        (236.0)       (1988.0)      (41.0)        (2433.0)
=============================================================================================================
director_facebook_likes         718.2141    501.3571      153.1017      1256.2782     13365.8537      123.9889
actor_3_facebook_likes          673.6907   10517.8095      108.9153       791.912      1120.5366      454.4098
actor_1_facebook_likes         6774.1757   27904.7619     1631.6314     8181.0599     12175.0976     5667.6539
language                         English      English         Other       English       English        English
country                              USA          USA  International          USA           USA            USA
budget                     41671447.8749 78048044.6667 136154565.4915  37303792.499  53060975.6098  35255539.7069
actor_2_facebook_likes          1729.8778    16857.1429      245.6737     2181.5101     3147.0244      1219.7994
imdb_score                        average      average          good          good       average        average



Time taken to build model (full training data) : 0.04 seconds

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

Clustered Instances

0        42 (  1%)
1       236 (  5%)
2      1988 ( 42%)
3        41 (  1%)
4      2433 ( 51%)
```

Figure 17: Results - kMeans Clustering

```
Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     IMDB-movie-data - test- classification-weka.filters.unsupervised.attribute.Remove-R1,5-17,20,23-weka.filters.unsupervised.attribute.Discretize-B4-M-1.0-Rfirst-last-precision4
Instances:    4740
Attributes:   8
              director_facebook_likes
              actor_3_facebook_likes
              actor_1_facebook_likes
              language
              country
              budget
              actor_2_facebook_likes
              imdb_score
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.95 (4503 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 1

Generated sets of large itemsets:

Size of set of large itemsets L(1): 5

Size of set of large itemsets L(2): 9

Size of set of large itemsets L(3): 7

Size of set of large itemsets L(4): 2

Best rules found:

 1. actor_1_facebook_likes='(-inf-160000]' 4736 ==> actor_2_facebook_likes='(-inf-34250]' 4736    <conf:(1)> lift:(1) lev:(0) [0] conv:(1)
 2. actor_1_facebook_likes='(-inf-160000]' budget='(-inf-3053875163.5]' 4734 ==> actor_2_facebook_likes='(-inf-34250]' 4734    <conf:(1)> lift:(1) lev:(0) [0] conv:(1)
 3. actor_3_facebook_likes='(-inf-5750]' actor_1_facebook_likes='(-inf-160000]' 4638 ==> actor_2_facebook_likes='(-inf-34250]' 4638    <conf:(1)> lift:(1) lev:(0) [0] conv:(0.98)
 4. actor_3_facebook_likes='(-inf-5750]' actor_1_facebook_likes='(-inf-160000]' budget='(-inf-3053875163.5]' 4636 ==> actor_2_facebook_likes='(-inf-34250]' 4636    <conf:(1)> lift:(1) lev:(0) [0] conv:(0.98)
 5. director_facebook_likes='(-inf-5750]' actor_1_facebook_likes='(-inf-160000]' 4552 ==> actor_2_facebook_likes='(-inf-34250]' 4552    <conf:(1)> lift:(1) lev:(0) [0] conv:(0.96)
 6. director_facebook_likes='(-inf-5750]' actor_1_facebook_likes='(-inf-160000]' budget='(-inf-3053875163.5]' 4550 ==> actor_2_facebook_likes='(-inf-34250]' 4550    <conf:(1)> lift:(1) lev:(0) [0] conv:(0.96)
 7. budget='(-inf-3053875163.5]' 4738 ==> actor_2_facebook_likes='(-inf-34250]' 4737    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.5)
 8. actor_3_facebook_likes='(-inf-5750]' 4641 ==> actor_2_facebook_likes='(-inf-34250]' 4640    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.49)
 9. actor_3_facebook_likes='(-inf-5750]' budget='(-inf-3053875163.5]' 4639 ==> actor_2_facebook_likes='(-inf-34250]' 4638    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.49)
10. director_facebook_likes='(-inf-5750]' 4556 ==> actor_2_facebook_likes='(-inf-34250]' 4555    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.48)
```

Figure 18: Results - Associate Finding using Apriori

```
=== Run information ===

Scheme:       weka.clusterers.EM -I 100 -N -1 -X 10 -max 5 -ll-cv 0.001 -ll-iter 0.001 -M 0.001 -K 10 -num-slots 1 -S 100
Relation:     IMDB-movie-data - test- classification-weka.filters.unsupervised.attribute.Remove-R1,5-17,20,23
Instances:    4740
Attributes:   8
              director_facebook_likes
              actor_3_facebook_likes
              actor_1_facebook_likes
              language
              country
              budget
              actor_2_facebook_likes
              imdb_score
Test mode:    evaluate on training data


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


EM
==

Number of clusters selected by cross validation: 5
Number of iterations performed: 24


                                Cluster
Attribute                            0            1            2            3            4
                                  (0.44)       (0.09)       (0.12)       (0.04)       (0.31)
=============================================================================================
director_facebook_likes
  mean                           131.8726     163.4472     4582.163     1705.7661     137.8704
  std. dev.                      206.2567     371.8445    6627.0062    4769.6887     189.2028

actor_3_facebook_likes
  mean                           331.2289       35.425     662.3092    6646.8596      521.011
  std. dev.                      221.0459      49.2537     433.3938     5301.735     269.1723

actor_1_facebook_likes
  mean                           741.6272     194.1412     19480.119    22661.6336   10338.5549
  std. dev.                       233.534     194.3726    33034.2622      14943.4    8126.5454

language
  English                       2028.0336     281.7147      542.0198     203.6895    1446.5424
  Mandarin                         7.4357       9.5645        1.0179       2.0134       6.9686
  Other                           16.2228      83.5748        2.0234       1.0021       8.1769
  Spanish                           6.842      28.1447        1.0003            1        4.013
  French                         25.9775      22.9967        1.0006       1.0001       5.0252
  Hindi                           2.0182       9.8933        1.9431       2.0578       4.0875
  Japanese                        1.0008       9.9992        4.9949       1.0051            1
  [total]                       2087.5306      445.888           554     211.7679    1475.8135
country
  USA                           1617.7267     190.6815      474.5678     183.2483    1189.7758
  UK                             211.7551      34.3072       28.8555      10.2566     139.8255
  International                   36.3545      94.7703       10.0557       2.0623      36.7572
  Canada                          74.4778      21.3276        2.9948       1.0116      18.1881
  Australia                       28.7666      12.9934        8.0994       1.9299       4.2108
  Germany                         34.9332      13.8409        11.373       3.4837      34.3692
  China                            6.4603       9.5366        1.4855       2.5489       5.9687
  France                          58.1135      32.5287        8.8792       7.1451      31.3336
  Japan                            4.0899       9.9992        4.0064        1.005       3.8995
  Spain                           11.6928      12.1744        5.0802       1.0126         7.04
  India                            1.0193       9.8933        1.9431       2.0578       4.0865
  Italy                             7.141       8.8348        1.6595       1.0061       5.3586
  [total]                       2092.5306      450.888           559     216.7679    1480.8135
budget
  mean                      19936818.1587  89339504.5953  46202275.7392  102471021.585  48051062.4321
  std. dev.                 17891216.952  647600170.2711  39112977.7164  89365523.5379  46017604.3855

actor_2_facebook_likes
  mean                           494.5049       65.9468     5862.0003     13441.7731     805.3356
  std. dev.                      253.7794       74.4218     5460.8974     10214.1266     383.2497

imdb_score
  good                           800.7568     286.9994      331.1694      121.4558     664.6186
  average                       1225.0507      146.879      217.8303       85.3121      799.928
  bad                             57.7232        8.0096        1.0003            1       7.2669
  [total]                       2083.5306      441.888           550     207.7679    1471.8135


Time taken to build model (full training data) : 8.92 seconds

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

Clustered Instances

0      2107 ( 44%)
1       443 (  9%)
2       555 ( 12%)
3       192 (  4%)
4      1443 ( 30%)


Log likelihood: -52.46592
```

Figure 19: Results - EM Clustering