Comparing the Performance of Different Classifier Data Mining Algorithms in Relation to the Size of the Training Set

Research Question: How does the volume of data affect the accuracy of predictions made using different classifier data mining algorithms?

Computer Science

Word Count = 3,994

Table of Contents

1. TITLE

Comparing the Performance of Different Classifier Data Mining Algorithms in Relation to the Size of the Training Set

2. RESEARCH QUESTION:

How does the volume of data affect the accuracy of predictions made using different classifier data mining algorithms?

3. INTRODUCTION:

As technology and the world as we know it progresses, the amount of information available and on demand increases exponentially along with it. The vastness of information and the ever increasing demand for it calls for more efficient methods of not only retrieving computer users' specific needs but also to equip governments and organizations with the tools needed to succeed.

The rapid growth of data has led to the creation of the term 'big data' which "consists of extensive dataset primarily in the characteristics of volume, variety, velocity, and/or variability that require a scalable architecture for efficient storage, manipulation, and analysis" ("NIST Big Data Interoperability Framework"). In real life, many patterns emerge from our day to day activities. It is therefore not uncommon that pattern seeking is a natural activity performed by the brain to simplify the process of decision making. Alongside big data therefore, emerges the need for a knowledge discovery process termed KDD (Knowledge Discovery in Databases) which analyses and extracts previously unforeseen patterns from big data through the use of a crucial process: data mining.

The advent of data mining algorithms is a great feat in the world of computer science. However, they have certain limitations in performing their duties, namely the size of the datasets and the quality of the data. Moreover, the process of data collection for research purposes can be cumbersome and costly. As a result of this problem it is essential that when performing research with data mining tools in a particular field, we know roughly how much data is required for the algorithms to function accurately.

Generally, it is established as the size of data increases, the accuracy of models increases. Firstly in the research paper titled "Comparisons between Data Clustering Algorithms", Osama Abu Abba evaluates the performance of different clustering data mining algorithms using factors such as the "size of the dataset, number of clusters, type of dataset and the type of software used." (Abba 1) This was done by using the algorithms to make clustering models and the evaluating them based on, the accuracy of the predictions of algorithms. From this research, it was discovered that the accuracy of the models built by the algorithms increased as the data size increased. From this research, I gained a fair understanding of the factors that affected the models present in data and in doing so, gained a valid premise for this research question. This research was however limited in the fact that it did not explore the type of relationship in depth and did not test for a wide range of data instances. It only tested for two data sizes: for 36,000 instances and for 4,000 instances which does not provide enough data for exploring the relationship. In my research, I intend to delve deeper into the nature of the relationship between the data size and the accuracy of the models.

It is therefore of interest to answer the research question: How does the volume of data affect the accuracy of predictions made using different classifier data mining algorithms specifically: the Linear Regression Algorithm, the Multilayer Perceptron Algorithm, the Random Tree Algorithm and finally, the IBk Lazy Algorithm.

In summary, findings from this research will:

1. Investigate the relationship between the size of the training dataset and the error rate.
2. Reveal the best algorithm (out of the ones being tested) at making accurate predictions.
3. And as an extension, reveal the minimum size of data required to make accurate predictions, to prevent the wastage of time and resources in collecting data.

4. BACKGROUND INFORMATION:

4.1 Data Mining:

Data mining is a novel data analysis technique that goes a step further than traditional mathematical analysis techniques in handling larger multivariate datasets and handling noise in data such as missing values or outliers. Data mining is a form of machine learning that enables a computer to learn the patterns underlying certain datasets so it can predict the outcomes of situations given certain variables. For this research, I will be using WEKA software (Waikato Environment for Knowledge Analysis) which is a "collection of machine learning algorithms and data pre-processing tools" (Witten et al, 7).

4.2 Types of Data Mining Algorithms

Data mining algorithms generally fall under the following categories: Classification, Association and Clustering. These categories may either adopt supervised or an unsupervised learning processes. In supervised learning, we know the variables as well as the outcomes and the job of the data mining technique is to build a model that can accurately relate the variables to the outcomes to be used for prediction and classification. However, in unsupervised learning, we are not aware of the outcomes of the events but we hope to find a model to help understand patterns in the data.

Classification data mining techniques are supervised forms of learning that can be used to create models that either classify data into groups such as “high” or “low” or predict numerical values by creating models based on training data.

Association data mining algorithms on the other hand, explore frequencies with which groups of data items, called ‘item sets’, occur together. In a supermarket, association data mining techniques can be used to identify items that customers often purchase together, in order to help stores to better arrange items in the stores or in designing targeted consumer ads.

Lastly, clustering is an unsupervised form of learning, used in sorting instances or items into clusters or groups which share similar properties. For my research, I will focus on classification data mining algorithms.

4.3 DATA MINING ALGORITHMS IN STUDY

Algorithm 1: Linear Regression Algorithm

This data mining algorithm works by finding a relationship that best describes the relationship between a set of known variables (x_1, x_2, x_3 etc.) and the target value (y), what we want to predict. Each of the variables is given a coefficient value and a measure of error is added to the model. This shown in the equation below:

$y = \text{Intercept} + \text{Sum}(b_1x_1, b_2x_2\ldots.) + E$ (Sabala) , where:

y: target value

x_n: Independent variables (predictor values)

Intercept: the y intercept on the graph of the model function

b_n = coefficients

E = error

Sum: method that finds the sum of the products of the coefficients and the predictor values.

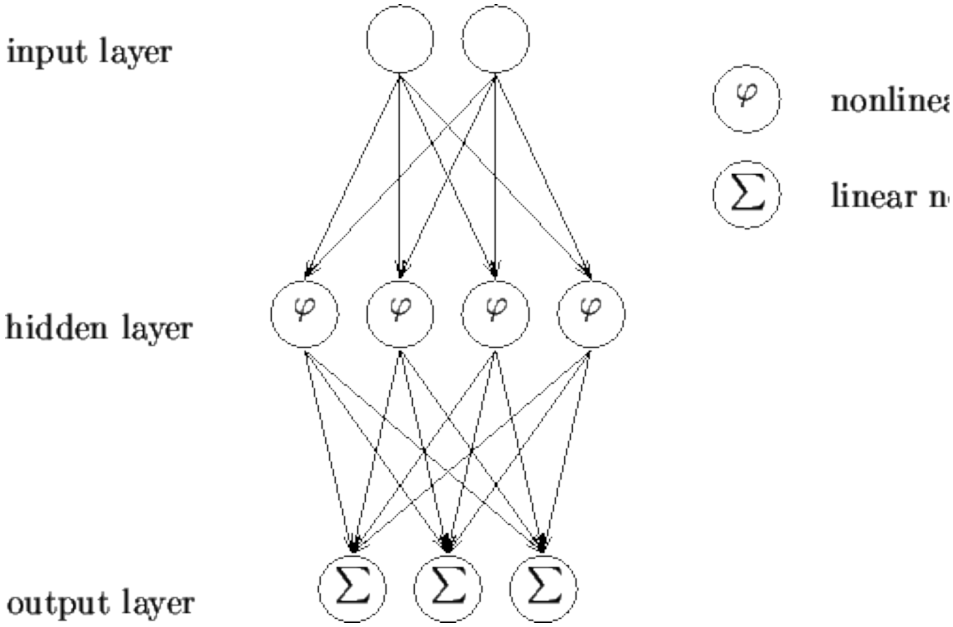
This algorithm which is an extension of traditional linear regression processes was chosen because of its reputation in making models in mathematics. I believe that it would be interesting to tests its performance in comparison to other techniques.

Algorithm 2: Multilayer Perceptron

The Multilayer Perceptron data mining algorithm is a neural network that makes uses layers, made up of nodes, to create models. There are three types of layers used in this data mining algorithm. They are:

- Input layer/Visible Layer: this is where the independent variables to be used in building the model are inputted.
- Hidden layer: This is the layer that connects the input and the output layer. In this layer, a special function called an activation function is applied to the data to discover the relationships between the variables and the target value. Since the activation function is a non-linear one, the relationships here are non-linearized relationships.
- Output layer: In this layer, the relationships are linearized with the use of a summation function and the final model is built.

Below is a diagram of the different layers:



(Ahonkela)

The formula for the multilayer perceptron algorithm can be seen below:

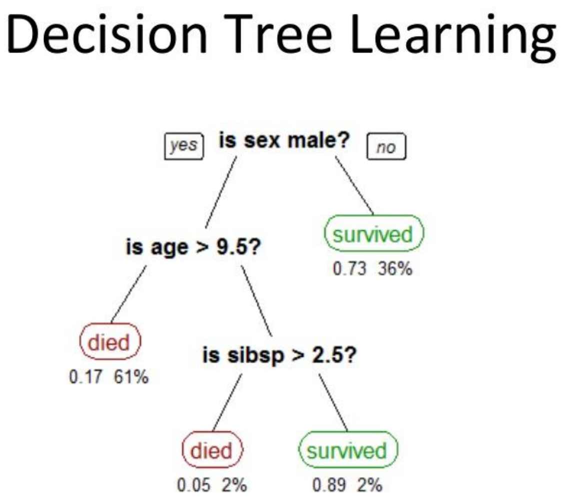
“$\mathbf{x} = \mathbf{f}(\mathbf{s}) = \mathbf{B}\boldsymbol{\varphi}(\mathbf{A}\mathbf{s} + \mathbf{a}) + \mathbf{b}$

where $\mathbf{s}$ is a vector of inputs and $\mathbf{x}$ a vector of outputs. $\mathbf{A}$ is the matrix of weights of the first layer, $\mathbf{a}$ is the bias vector of the first layer. $\mathbf{B}$ and $\mathbf{b}$ are, respectively, the weight matrix and the bias vector of the second layer. The function $\boldsymbol{\varphi}$ denotes an element wise nonlinearity. The generalisation of the model to more hidden layers is obvious.” (Ahonkela)

This algorithm is a relatively novel is designed to incorporate a back propagation process which works by recursively calculating the minimum of the error function (through gradient descent) at the output layer of the algorithm and then using this information to correct the weights assigned in previous nodes minimize the output error. (Rojas, 156)

Algorithm 3: Random Tree Algorithm

This algorithm works by using tree-like structures to model data to be able to predict a final outcome. The different paths to take are determined through the use of probabilities. After this a decision is made and the data is split into two different nodes or outcomes. This process of decision making and synthesis (break down), is continued until we reach the final outcome of the model, that is the target value. An example of a decision tree algorithm which predicts the survivors of the Titanic Ship is seen below:



Decision tree showing survival of passengers on the *Titanic* ('sibsp' is the number of spouses or siblings aboard). The fig. under the leaves show the probability of survival and the percentage of observations in the leaf.

(Pandey)

Algorithm 4: IBk Algorithm

This algorithm makes use of the k-nearest neighbour technology in order to classify data into groups or to make predictions. This algorithm falls under the broader category of lazy learning classifier algorithms. Instead of going through all the data instances and learning from them all together, it learns on an instance by instance basis. The class of the next instance is predicted based on the instances in the training set by finding the new instance's proximity to instances whose class is already known (Korting). The value of k determines the region around a new instance that is used in determining the class or group the new instance belongs to. For a k value of 3, the instance will be plotted on a portion on the grid based on the variables and then a using certain distance functions, the 3 nearest data points to the algorithm are found and then based on these data points the identity or value of the new instance is predicted (Korting). "For regression this might be the mean output variable, in classification this might be the mode (or most common) class value" (Brownlee)

A visualization of a lazy learner functioning can be seen in the diagram below:

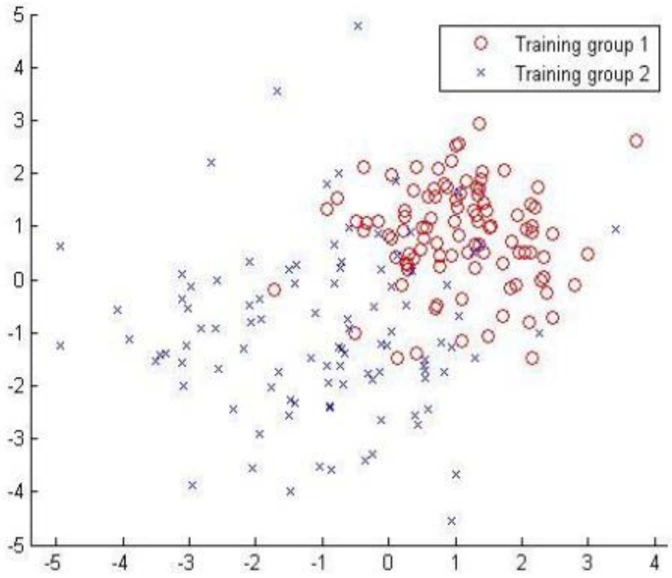


Fig.1 Distance used Euclidean and Rule Nearest (Kataria et al)

In the above diagram, we can see the visualization of the end of the training process for a kNN algorithm which has divided the training instances into two groups using a Euclidian and Rule Nearest distance functions.

In this research k is set to the default, 1, and a Euclidian search algorithm is used to determine the distance between instances.

The Euclidean formula for determining the distance between two points x and u can be calculated as:

$$d(\mathbf{x}, \mathbf{u}) = \sqrt{\sum_{i=1}^{n}(x_i - u_i)^2}.$$

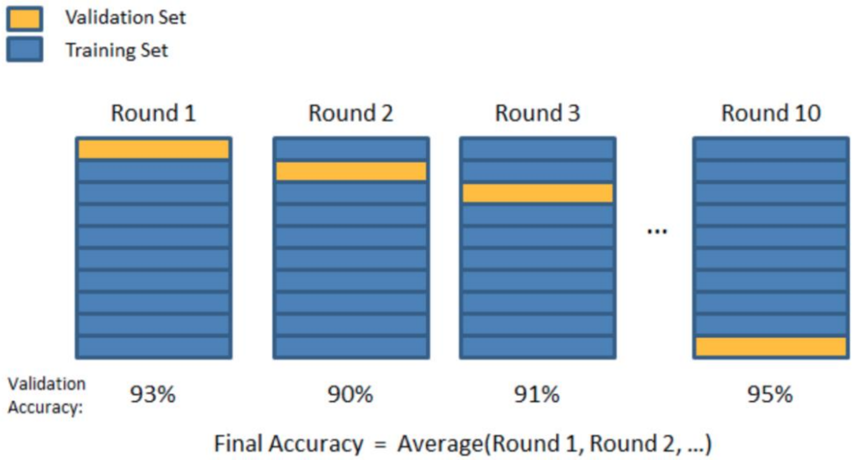
(Leung, 5)

This algorithm is one of the simplest and most intuitive machine-learning algorithms however it has high computational costs. (Kavuri, 22)

4.1 MODEL VALIDATION

The models built from the varying dataset sizes are validated using an validation method called the K-fold cross validation.

According to Stefanowski, in the K-fold cross evaluation process, the training set is divided into k-subsets of equal size (for example if k=5, the training set is divided into groups of 5). Each subset is used for testing while the rest is used for training. (E.g. for 5 subsets, 4 subsets out of the 5 are used for training and the remaining subset is used for testing the model). After this the average error from the cross validation is calculated. (Stefanowski, 24) This process of model building and evaluation is repeated for each training sample size and the average error is measured. An example of this process is seen in the diagram below:



(Bronshtein)

The above image shows a summary of the cross validation process; however, unlike in the photo, in this research, the error for each round is calculated and then an average error is found at the end of the rounds.

5. INVESTIGATION

5.1 DATASET

The dataset being tested on in this experiment is called the Boston House-Price Data collected by Harrison and Rubinfeld. The dataset was obtained from a data library called StatLib Datasets Archive (source: lib.stat.cmu.edu). This dataset which was collected as part of a research titled Hedonic Housing Prices and Demand for Clean Air (Rubinfeld et al, 96), contains information collected by the US Census Service concerning housing in the area of Boston Massachusetts. This dataset has 506 instances. This dataset is strictly numerical, that is, it contains only numbers and not categories in the form of text. The dataset contains the following column headings:

"CRIM: per capita crime rate by town

ZN: proportion of residential land zoned for lots over 25,000 square feet

INDUS: proportion of non-retail business acres per town

CHAS: Charles River dummy variable (1 if tract bounds river; 0 otherwise)

NOX: nitric oxides concentration (parts per 10 million)

RM: average number of rooms per dwelling

AGE: proportion of owner-occupied units built prior to 1940

DIS: weighted distances to five Boston employment centres

RAD: index of accessibility to radial highways

TAX: full-value property-tax rate per $10,000

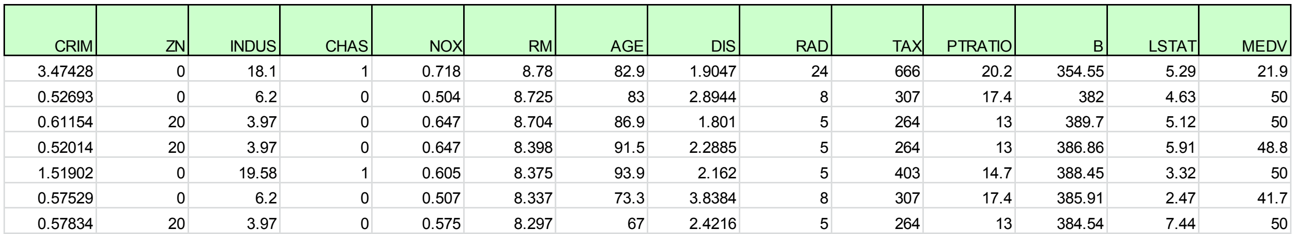
PTRATIO: pupil-teacher ratio by town

B: 1000(Bk -0.63)^2 where Bk is the proportion of blacks by town

LSTAT: % lower status of the population

MEDV: Median value of owner-occupied homes in $1,000" (Shmueli et al, 21)

A sample of this dataset can be seen below:



CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
3.47428	0	18.1	1	0.718	8.78	82.9	1.9047	24	666	20.2	354.55	5.29	21.9
0.52693	0	6.2	0	0.504	8.725	83	2.8944	8	307	17.4	382	4.63	50
0.61154	20	3.97	0	0.647	8.704	86.9	1.801	5	264	13	389.7	5.12	50
0.52014	20	3.97	0	0.647	8.398	91.5	2.2885	5	264	13	386.86	5.91	48.8
1.51902	0	19.58	1	0.605	8.375	93.9	2.162	5	403	14.7	388.45	3.32	50
0.57529	0	6.2	0	0.507	8.337	73.3	3.8384	8	307	17.4	385.91	2.47	41.7
0.57834	20	3.97	0	0.575	8.297	67	2.4216	5	264	13	384.54	7.44	50

The attributes listed above were determined to be factors that affected the value of houses in the Boston state, based on extensive research and investigation. The strength of the correlation between the independent variables and the target variable will determine the model. Since this data was obtained from real life, a few anomalous values within the data that would also affect the accuracy of the models. As a result as part of the method, data pre-processing was required in order to clean the data before building models on them.

5.2 PREPROCESSING OF DATA

Data pre-processing is the process of preparing data before performing mining on the data. The accuracy of the models that can be built from any dataset depends on the quality of the data. The quality of the data may be affected by the presence of:

I. Outliers and Extreme Values:

   Outliers in the data were removed using special filters in the WEKA package: weka.filters.unsupervised.attribute.InterquartileRange. This filter operates using statistical measures of outliers. It works by finding the interquartile range for the instances of each attribute, and then determines the maximum and minimum threshold for flagging outliers by finding 3 times the IQR and 6 times the IQR to identify the threshold for extreme values. Removal of these outliers will improve the efficiency of the models built with the algorithms.

II. Irrelevant attributes:

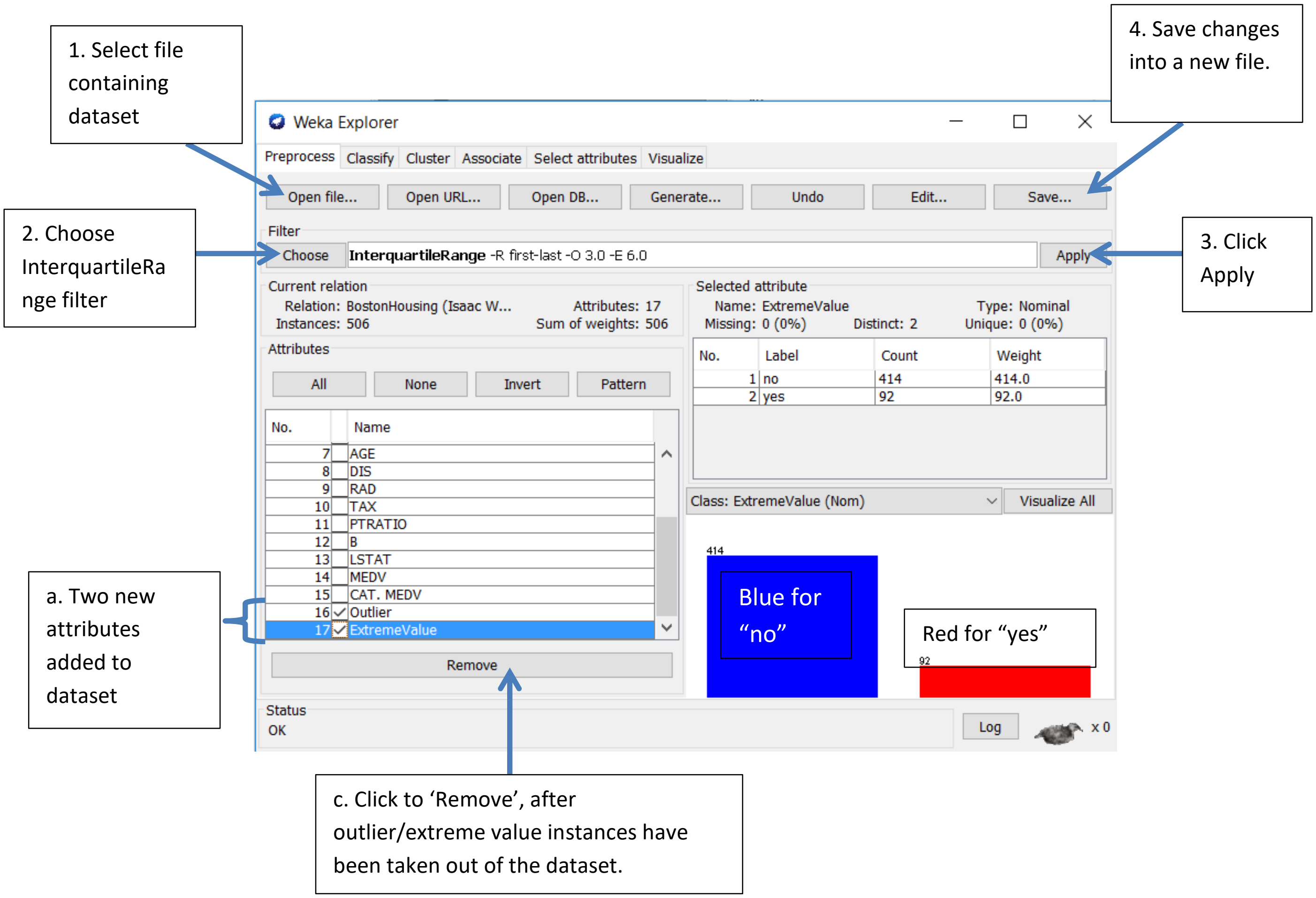
   No irrelevant attributes were identified in the dataset. To determine this, graphs of each instance against the target attribute (what out model aims to predict): median value. The strength of the correlation between these two variables is then used to determine whether or not it is relevant in building the model.

III. Missing values:

There were no missing values in this dataset.

After pre-processing the data, a total of 164 instances were taken out, leaving 342 instances (out of the initial 506 instances) for experimentation.

5.2.3 Diagram and Steps for Data Pre-processing:



1. Select file containing dataset

4. Save changes into a new file.

2. Choose InterquartileRange filter

3. Click Apply

Filter: InterquartileRange -R first-last -O 3.0 -E 6.0

Current relation
Relation: BostonHousing (Isaac W... Attributes: 17
Instances: 506 Sum of weights: 506

Selected attribute
Name: ExtremeValue Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	no	414	414.0
2	yes	92	92.0

Attributes

No.		Name
7	□	AGE
8	□	DIS
9	□	RAD
10	□	TAX
11	□	PTRATIO
12	□	B
13	□	LSTAT
14	□	MEDV
15	□	CAT. MEDV
16	☑	Outlier
17	☑	ExtremeValue

Class: ExtremeValue (Nom)

Status
OK

a. Two new attributes added to dataset

Blue for "no"

Red for "yes"

c. Click to 'Remove', after outlier/extreme value instances have been taken out of the dataset.

a. After doing this, two new attributes, Outlier and ExtremeValue are added to the dataset to indicate whether a particular instance is an outlier/extreme value. If value of either attribute is 'yes' it means that there is either an outlier or extreme value in the data.

b. Open the new file in a text editor and delete all instances where either attributes Outlier or ‘ExtremeValue’ say “yes”.

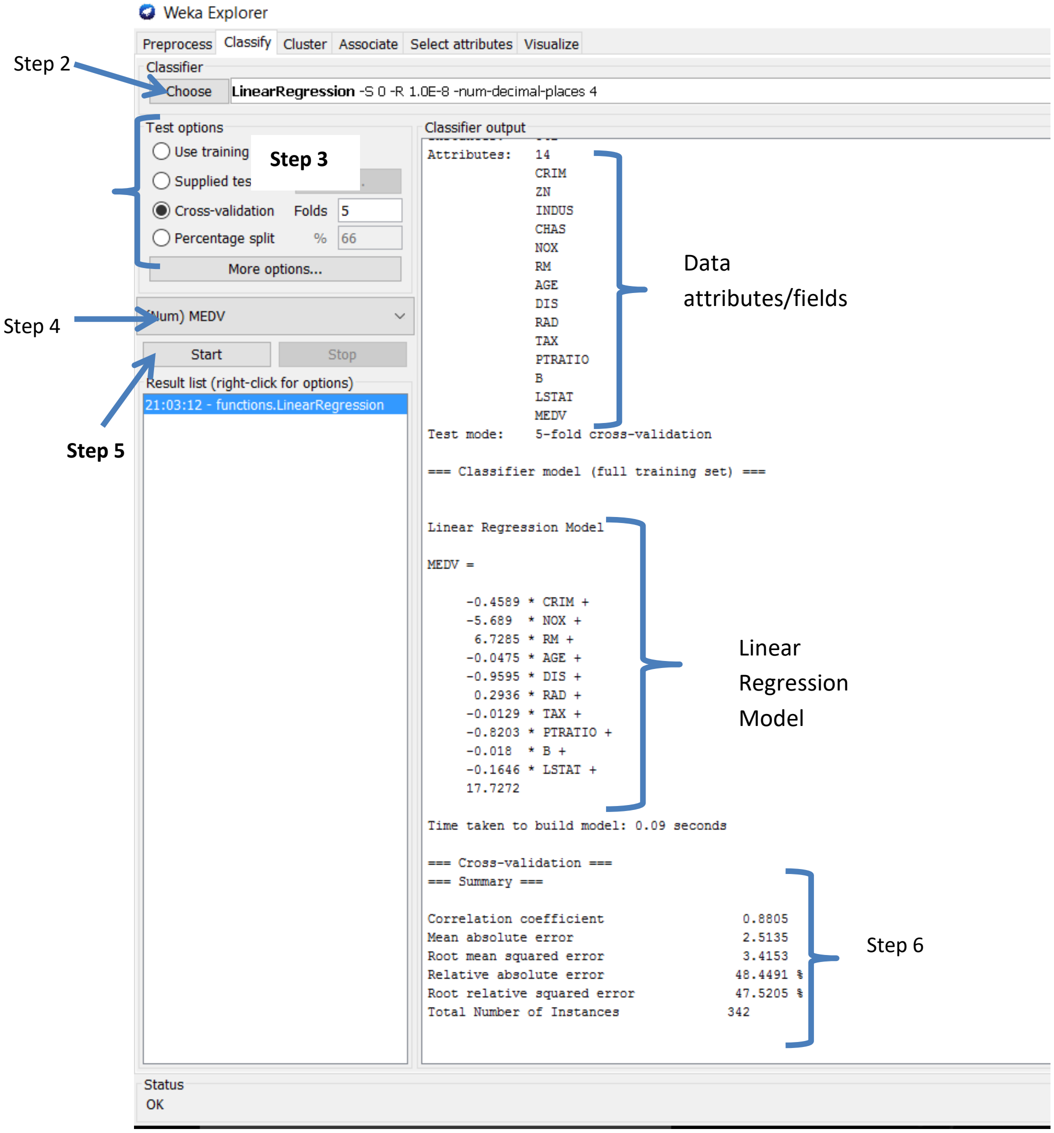
c. Save file again, open in the WEKA platform then select and remove the two attributes Outlier and ‘ExtremeValue’ from the dataset.

5.3 TESTING

The following processes were carried out in testing the data:

1. Select first file containing a particular number of instances of the dataset e.g. 342 instances.
2. Select data mining algorithm.
3. Select cross evaluation and set k to a 5 fold cross evaluation.
4. Set target variable to MEDV.
5. Click ‘Start’ to create a model based on that dataset.
6. Record the correlation coefficient, the mean average error and the root mean squared error of the algorithm from the output panel of the interface.
7. Repeat the above steps for each of the number of instances of the training samples.
8. Repeat steps 1-6 for the next algorithm.
9. Record the mean average error and the number of instances of the training set for each of the four algorithms.
10. Plot a graph of the mean average error against the number of instances of each of the four algorithms

5.4 Diagram for Testing



Classifier: Choose LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

Test options

○ Use training

○ Supplied tes

● Cross-validation Folds 5

○ Percentage split % 66

(Num) MEDV

Result list (right-click for options)

21:03:12 - functions.LinearRegression

Classifier output

```
Attributes:   14
              CRIM
              ZN
              INDUS
              CHAS
              NOX
              RM
              AGE
              DIS
              RAD
              TAX
              PTRATIO
              B
              LSTAT
              MEDV
Test mode:    5-fold cross-validation

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


Linear Regression Model

MEDV =

     -0.4589 * CRIM +
     -5.689  * NOX +
      6.7285 * RM +
     -0.0475 * AGE +
     -0.9595 * DIS +
      0.2936 * RAD +
     -0.0129 * TAX +
     -0.8203 * PTRATIO +
     -0.018  * B +
     -0.1646 * LSTAT +
     17.7272

Time taken to build model: 0.09 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.8805
Mean absolute error                      2.5135
Root mean squared error                  3.4153
Relative absolute error                 48.4491 %
Root relative squared error             47.5205 %
Total Number of Instances              342
```

Status

OK

6 RESULTS AND ANALYSIS OF RESULTS

This section contains the tables of results from the experiment and graphical analysis of the data.

6.1 TABLE OF RESULTS FOR EACH ALGORITHM

6.1.1 Descriptions of columns in table:

Number of Instances: This shows the number of instances present in each of the training samples.

Correlation Coefficient: This column shows the strength of the correlation between the target value (house median value) and the model.

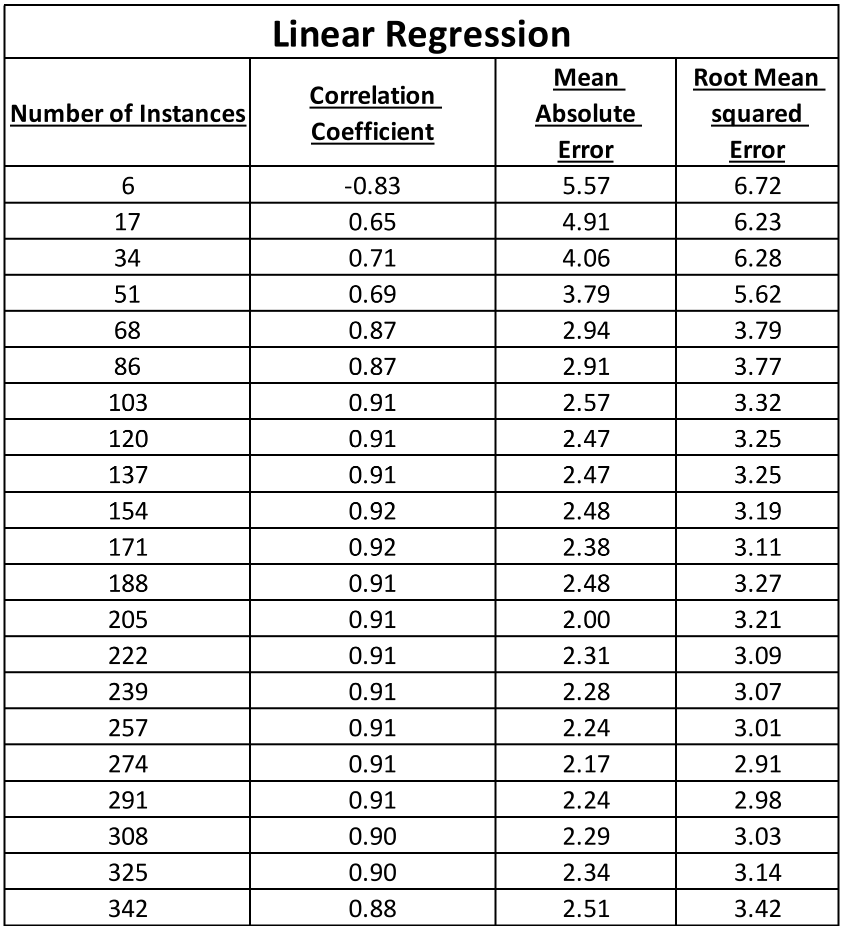
Mean Absolute Error (MAE): this metric measure the average of the absolute difference between the predicted values and the true values of all instances in the training set (springer.com). It shows us the average error of all the predictions made by the model.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}_j|$$

(“MAE and RMSE - Which Metric Is Better?”)

Root Mean Squared Error (RSME): This metric is similar to the MAE however it is more sensitive to the difference between a model’s predictions and the actual value. In this instance, I will use solely the MAE since the nature of our data (housing prices) does not require great sensitivity.

Table 1.0: Table of Results for Linear Regression Algorithm



Linear Regression			
Number of Instances	**Correlation Coefficient**	**Mean Absolute Error**	**Root Mean squared Error**
6	-0.83	5.57	6.72
17	0.65	4.91	6.23
34	0.71	4.06	6.28
51	0.69	3.79	5.62
68	0.87	2.94	3.79
86	0.87	2.91	3.77
103	0.91	2.57	3.32
120	0.91	2.47	3.25
137	0.91	2.47	3.25
154	0.92	2.48	3.19
171	0.92	2.38	3.11
188	0.91	2.48	3.27
205	0.91	2.00	3.21
222	0.91	2.31	3.09
239	0.91	2.28	3.07
257	0.91	2.24	3.01
274	0.91	2.17	2.91
291	0.91	2.24	2.98
308	0.90	2.29	3.03
325	0.90	2.34	3.14
342	0.88	2.51	3.42

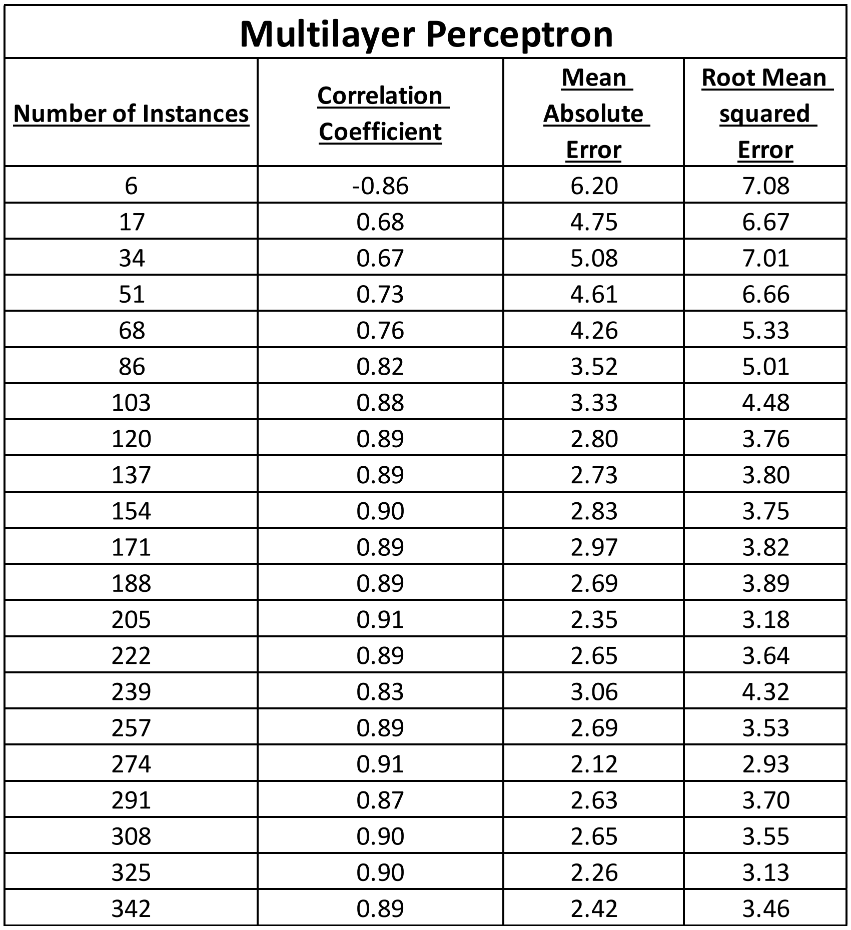
Table 2.0: Table of Results for Random Tree Algorithm

Random Tree			
Number of Instances	**Correlation Coefficient**	**Mean Absolute Error**	**Root Mean squared Error**
6	-0.73	5.60	6.29
17	-0.18	8.80	13.26
34	0.67	4.06	5.04
51	0.69	4.26	5.14
68	0.63	4.97	6.75
86	0.62	4.66	6.16
103	0.74	4.14	5.53
120	0.81	3.60	4.94
137	0.77	4.08	5.66
154	0.80	3.51	4.78
171	0.74	4.19	6.02
188	0.80	3.64	4.84
205	0.81	3.28	4.63
222	0.80	3.21	4.66
239	0.77	3.91	5.25
257	0.79	3.41	4.69
274	0.81	3.13	4.37
291	0.81	3.04	4.21
308	0.81	3.13	4.33
325	0.82	2.87	4.14
342	0.83	2.97	4.27

Table 3.0: Table of Results for IBk Algorithm

IBK			
Number of Instances	Correlation Coefficient	Mean Absolute Error	Root Mean squared Error
6	-0.31	4.00	4.42
17	0.34	5.44	7.36
34	0.65	4.44	5.61
51	0.56	4.35	6.00
68	0.76	4.26	5.33
86	0.71	4.21	5.58
103	0.81	3.57	4.72
120	0.77	3.67	5.41
137	0.78	3.60	5.28
154	0.82	3.24	4.54
171	0.79	3.51	5.07
188	0.81	3.29	4.70
205	0.82	3.04	4.42
222	0.81	3.08	4.41
239	0.80	3.26	4.60
257	0.82	3.02	4.21
274	0.82	2.89	4.14
291	0.83	2.74	3.95
308	0.81	2.89	4.16
325	0.82	2.79	4.09
342	0.80	3.08	4.44

Table 4.0: Table of Results for Multilayer Perceptron Algorithm



Multilayer Perceptron			
Number of Instances	**Correlation Coefficient**	**Mean Absolute Error**	**Root Mean squared Error**
6	-0.86	6.20	7.08
17	0.68	4.75	6.67
34	0.67	5.08	7.01
51	0.73	4.61	6.66
68	0.76	4.26	5.33
86	0.82	3.52	5.01
103	0.88	3.33	4.48
120	0.89	2.80	3.76
137	0.89	2.73	3.80
154	0.90	2.83	3.75
171	0.89	2.97	3.82
188	0.89	2.69	3.89
205	0.91	2.35	3.18
222	0.89	2.65	3.64
239	0.83	3.06	4.32
257	0.89	2.69	3.53
274	0.91	2.12	2.93
291	0.87	2.63	3.70
308	0.90	2.65	3.55
325	0.90	2.26	3.13
342	0.89	2.42	3.46

7.0 GRAPHS AND ANALYSIS OF GRAPHS

For each of the curves below, the Mean Average Error has been plotted on the y axis and the Number of instances of the training sample on the x axis.

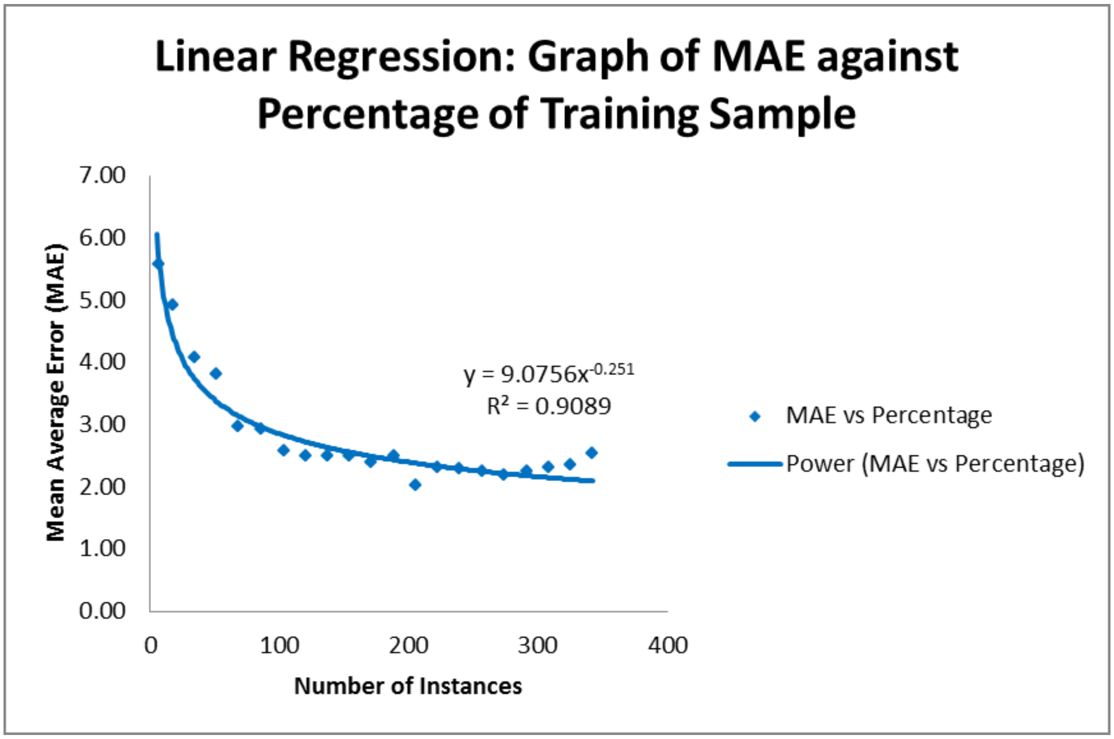
From each one of the graphs, we can identify the following variables from the best fit line in order to give us a sense of the performance of the algorithms. We can identify:

1. Decay Rate

2. Error Coefficient (Maximum Achievable Error)

3. Fail rate (ratio of decay rate to the error coefficient) = Maximum achievable Error/Decay Rate.

7.1 Linear Regression Algorithm

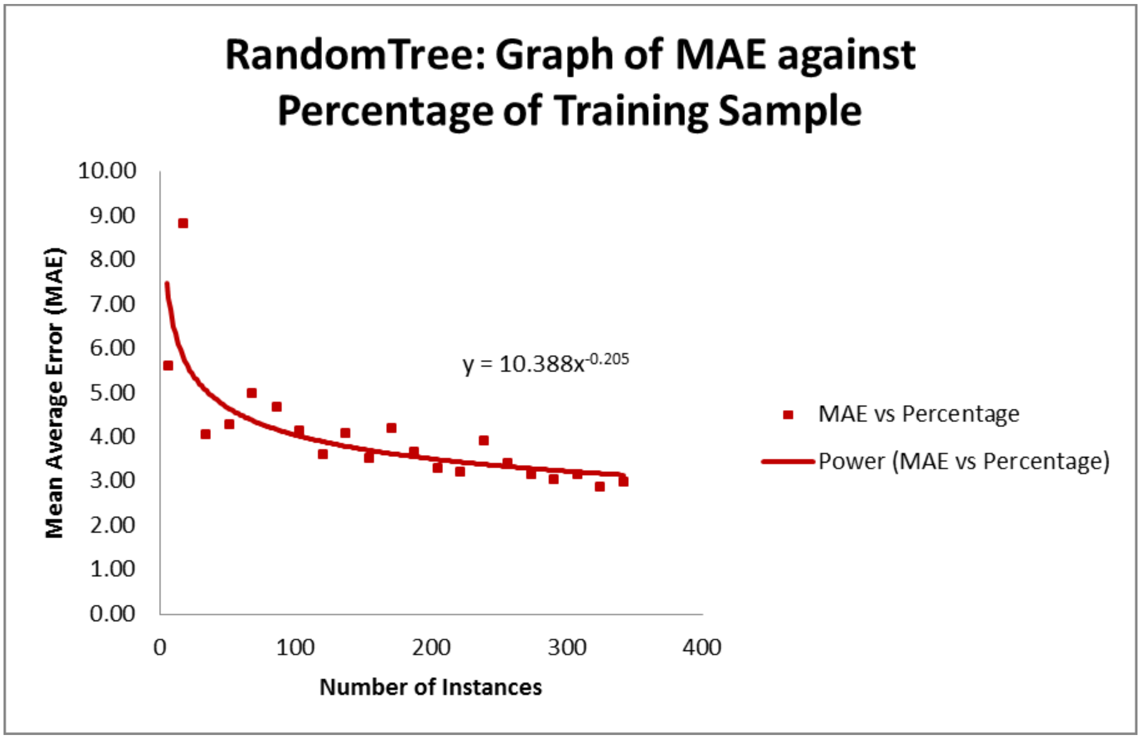


Graph 1.0

The graph above is plotted using data from Table 1.0. It shows the learning curve for the linear regression data mining algorithms. The graph line shows an inverse power relationship between the MAE and the number of instances of the training sample; as the size of the training sample increases, the average number of errors made by the model decreases. From the graph the relationship $y= ax^b$, where '*a'* is the error multiplier which is determined by the maximum achievable error, and *b* is the decay rate. (Figueroa et al, 2). This value determines the extent to which the mean average error reduces as the number of instances of the training sample increases. The maximum achievable error for this algorithm has been determined to be 9.0756 while the decay rate for this graph is -0.251. The performance ratio for this algorithm is found by dividing the maximum achievable error by the absolute value of the decay rate. This is shown below:

9.0756/0.251 = 36.15. This algorithm therefore has an approximate fail rate of 36.

7.2 Random Tree Algorithm



Graph 2.0

The graph above is plotted using data from Table 2.0 for the Random Tree Algorithm.

Summary of information from Graph 2.0:

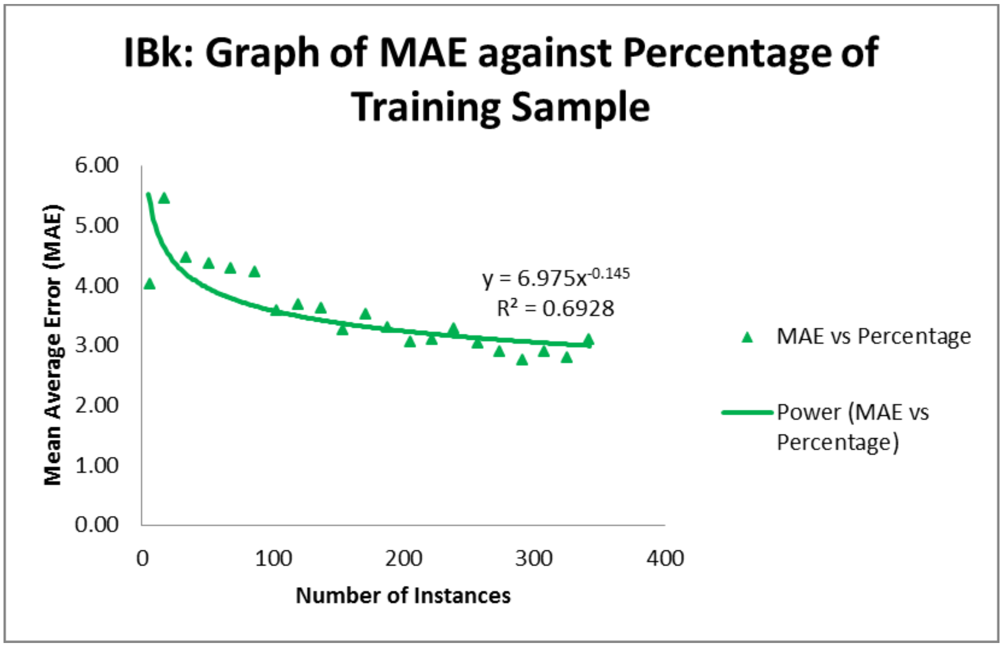
This graph is described by the equation MAE= $10.388x^{-0.205}$ where x: number of instances.

Decay rate= -0.205

Maximum achievable error= 10.388

Fail rate= 10.388/0.205= 50.67. This algorithm therefore has an approximate fail rate of 51.

7.3 IBk Algorithm



Graph 3.0:

The graph above is plotted using data from Table 3.0 for the Lazy IBk data mining algorithm.

Summary of information from Graph 3.0:

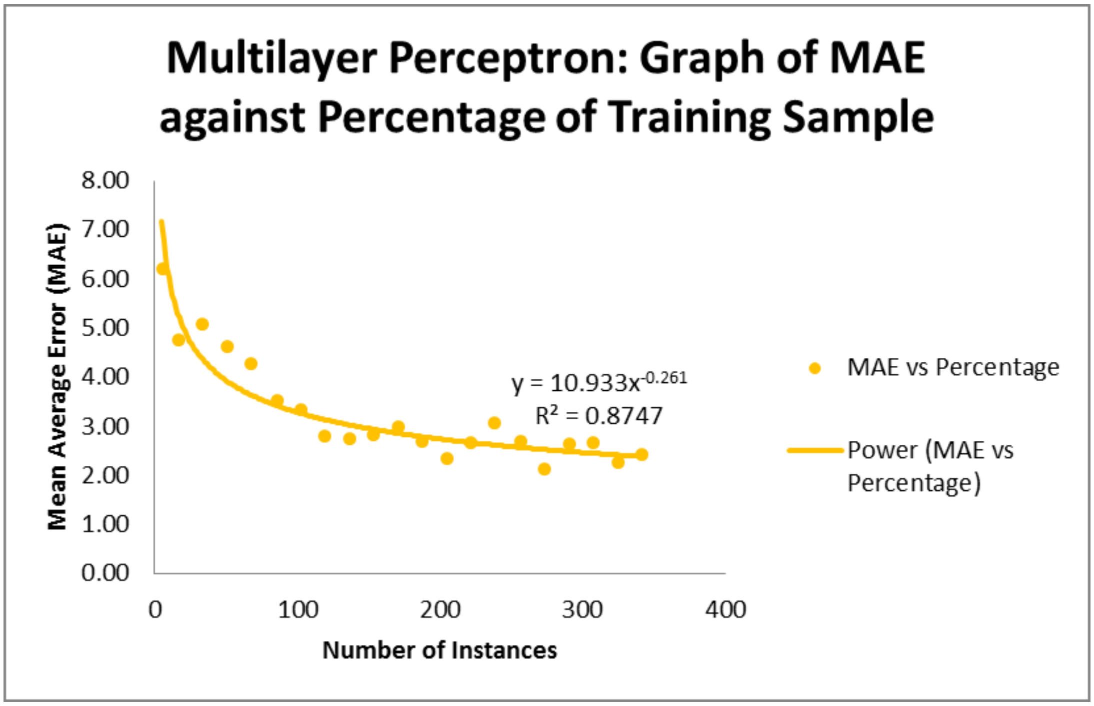
This graph is described by the equation MAE= 6.975x $^{-0.145}$ where x: number of instances.

Decay rate= -0.145

Maximum achievable error= 6.975

Fail rate= 6.975 /0.145 = 48.103 which is approximately equal to 48.

7.4 Multilayer Perceptron Algorithm



Graph 4.0:

The graph above is plotted using data from Table 4.0. It shows the curve for the Multilayer Perceptron Algorithm.

Summary of information from Graph 4.0:

The graph is described by the equation MAE = $10.933x^{-0.261}$ where x: number of instances.

Decay rate= -0.261

Maximum achievable error= 10.933

10.933/0.261 = 41.888, which is approximately equal to 42.

By comparing the fail ratings of the different algorithms together:

LR: 36

RT: 51

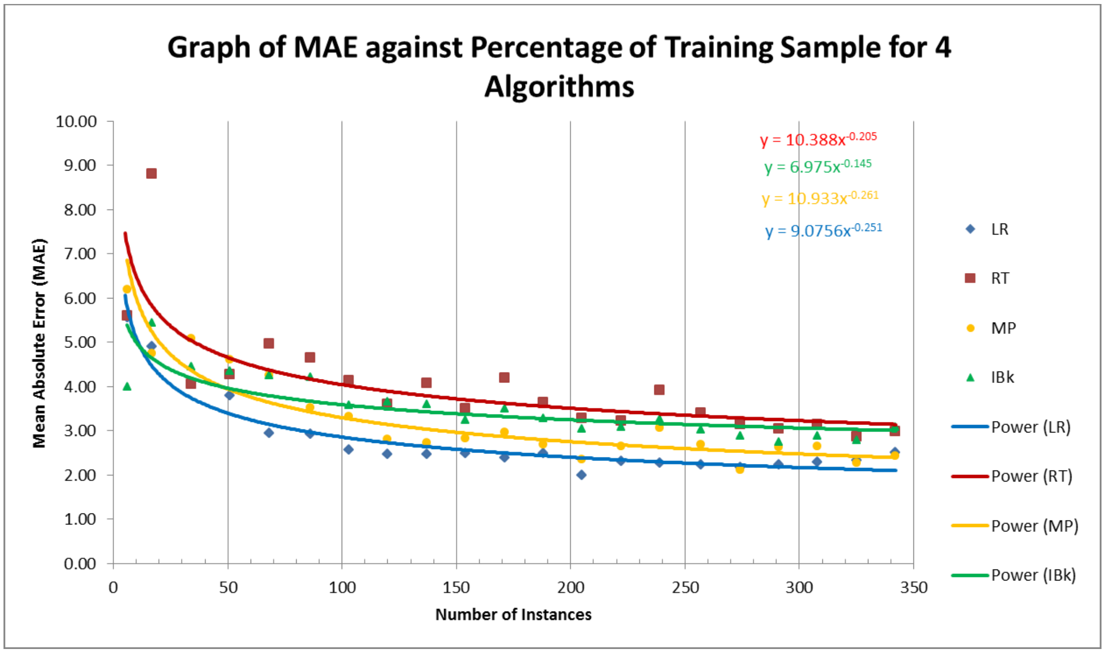
IBk: 48

MP: 42

We expect that when the curves are plotted on the same graph, they should be in the following order, from the worst to the best algorithm:

1st RT 2nd IBk 3rd MP 4th LR

7.5 All Algorithms Together



From the above graph, the curves are arranged in increasing order of accuracy from the top to the bottom. As predicted from the failure ratings, the algorithm that performs most poorly is the Random Tree Algorithm, followed by the IBk Algorithm, then by the Multilayer Perceptron Algorithm and finally by the Linear Algorithm which is the most accurate of the 4 algorithms.

Moreover, we see that for each of the algorithms, after a certain number of instances, the change in the mean average error is very minimal. According to Delmaster and Hancock, a rule of thumb for determining the amount of data required for successful classifier performance is to have 6 x m x p instances, where m is the number of target classes and p is the number of variables being investigated (Delmaster, 68). In this research, there are 14 variables in the dataset and 1 target variable. Therefore according to this suggested rule, there should be at least 6 x 1 x 14 = 84 instances, for successful classifier performance. From the graph, for each algorithm we notice that as the number of instances increases, there is a large change in the mean average error, however after a certain number of instances, there are only slight reductions in the mean average error. For the Linear regression algorithm, this value can be estimated to be roughly around 90 instances. This appears to agree with the number suggested by Delmaster and Hancock. Though this estimate may vary from algorithm to algorithm, it gives us a fair value of roughly how much data we would require in using classification data mining algorithms to build fairly accurate models for this dataset.

8.0 CHALLENGES AND RECCOMENDATIONS:

This research was conducted using only one dataset which contained numerical data. This means that the findings on the performance of the classifier data mining algorithms evaluated will only apply to numerical data. As an extension, some of these algorithms may be tested using categorical data as well to see if similar trends occur.

As a recommendation, from the findings of this research, we propose that the Linear Regression Algorithm is the most suitable algorithm for making accurate models from numerical data.

9.0 CONCLUSION

This research investigates the relationship between the volume of the data used in building models and the accuracy of these models. For each one of the algorithms tested, we see that as the size of the training sample increases, the mean average error of the predictions made by the model decreases. Among the 4 algorithms: Linear Regression, Random Tree and IBk and Multilayer Perceptron data mining algorithms, the Linear Regression Algorithm is the most accurate while the Random Tree Algorithm is the least accurate.

The method of analysis used in this research may serve as a framework for measuring and comparing the performance of different data mining algorithms.

9.0 Works Cited:

- Abba, Osama A. "Comparisons Between Data Clustering Algorithms." The International Arab Journal of Information Technology, vol. 5, no. 3, July 2008, pp. 320-25. Accessed 11 Oct. 2017.
- “MAE and RMSE - Which Metric Is Better?” Medium, Human in a Machine World, 23 Mar. 2016, medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d.
- Ahonkela, Antti. Multilayer Perceptron, https://www.hiit.fi/u/ahonkela/dippa/node41.html. Accessed 25 Oct. 2017.
- Bronshtein, Adi. Train/Test Split and Cross Validation in Python, 17 May 2017, https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6. Accessed 20 Oct. 2017.
- Brownlee, Jason. “K-Nearest Neighbors for Machine Learning.” Machine Learning Mastery, Machine Learning Mastery, 21 Sept. 2016, machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/. Accessed 24 Aug. 2017
- Delmaster, R., and Hancock, M. (2001), Data Mining Explained, Boston: Digital Press.
- *Encyclopedia of Machine Learning*. , 2010, p. 652, https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_525. Accessed 26 Oct. 2017.
- Figueroa, Rosa L., Qing Zeng-Treitler, Sasikiran Kandula, and Long H. Ngo. "Predicting sample size required for classification performance." *Research Article*, BMC Medical Informatics & Decision Making, 2012. Accessed 26 Oct. 2017. Path: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307431/.

- Grady, Nancy, and Wo Chang, editors. NIST Big Data Interoperability Framework. Vol. 1, NIST Special Publication, 2015, https://bigdatawg.nist.gov/_uploadfiles/NIST.SP.1500-1.pdf. Accessed 14 Oct. 2017.
- Gupta, Prashant. *Cross-Validation in Machine Learning*, 5 June 2017, https://medium.com/towards-data-science/cross-validation-in-machine-learning-72924a69872f. Accessed 24 Oct. 2017.
- Harrison, David, and Daniel Rubinfeld. "Hedonic Housing Prices and the Demand for Clean Air." *Journal of Environmental Economics and Management*, vol. 5, 1978, pp. 81-102, https://www.law.berkeley.edu/files/Hedonic.PDF. Accessed 24 Oct. 2017.
- Kataria, Aman, and M D. Singh. "A Review of Data Classification Using K-Nearest Neighbour Algorithm." International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 6, June 2013, pp. 354-60, citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.3893&rep=rep1&type=pdf. Accessed 22 Oct. 2017.
- Kavuri, Ajay. "K-Nearest Neighbour Algorithm." KNN, Slide Share, 22 May 2015, https://www.slideshare.net/ajaykrishnateja/knn-48499627. Accessed 10 Oct. 2017.
- Korting, Thales S, director. How KNN Algorithm Works. How KNN Algorithm Works, YouTube, 18 Feb. 2014, www.youtube.com/watch?v=UqYde-LULfs. Accessed 18 Aug. 2017
- Pandey, Akur. "The Science Behind Predictive Analytics: A Text Mining Perspective." Slide Share, 22 Mar. 2013, https://www.slideshare.net/ankurpandeyinfo/the-science-behind-predictive-analytics-a-text-mining-perspective-17498749. Accessed 28 Sept. 2017.
- Rojas, Raul. Neural Networks. Berlin, Springer-Verlag, 1996, pp. 155-56, https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf. Accessed 9 Nov. 2017.

- Sabala, Michal. “PMML 2.0--Regression.” Data Mining Group - Regression, Data Mining Group, dmg.org/pmml/v2-0/Regression.html.
- Shmueli, Galit, Nitin R. Patel, and Peter C. Bruce. Data Mining in Excel: Lecture Notes and Cases, Resampling Stats, Inc., 30 Dec. 2005. Accessed 26 Oct. 2017.
- Stefanowski, Jerzy. Data Mining Evaluation of Classifiers, Poznan University of Technology, 2010, www.cs.put.poznan.pl/jstefanowski/sed/DM-4-evaluatingclassifiersnew.pdf. Accessed 18 Oct. 2017.
- Vlachos, Pantelis. *StatLib---Datasets Archive*, 19 July 2005, lib.stat.cmu.edu/datasets/. Accessed 24 Oct. 2017.
- Witten, Ian H., Eibe Frank, Mark A. Hall, and Christopher J. Pall. The WEKA Workbench. Fouth Edition ed., Morgan Kauffman, 2016, https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf. Accessed 18 Aug. 2017