

**LAB 2**

**MOVIE DATABASE TEXT MINING USING WEKA J-48, RANDOM FOREST AND RANDOM TREE**

COURSE CODE : MANB1143

COURSE NAME : BUSINESS INTELLIGENCE

PROGRAMME : MSc (BIA)

PREPARED BY : AHMAD NIZAM BIN MOHD LAZIM - MAN171058

AHMAD SHAKIR BIN AHMAD ZULKIFLI - MAN171020

AMJAD ALASSEH - MAN171009

MUHAMMAD NIDZAM MASO’OD - MAN171043

SUBMITTED TO : DR AZIZUL AZIZAN

Contents

[1. Introduction 3](#_Toc503612632)

[2. Objective 3](#_Toc503612633)

[3. Scope 4](#_Toc503612634)

[4. Aim 4](#_Toc503612635)

[5. Tools 4](#_Toc503612636)

[6. Steps 4](#_Toc503612637)

[7. Result 6](#_Toc503612638)

[8. Conclusion 11](#_Toc503612639)

[9. References 11](#_Toc503612640)

1. Introduction

Business Intelligence industry has grown rapidly in the last couple of years. With the advent of latest technology Text Mining has taken Business Intelligence to the next level. Businesses or organisation would be able to capitalise or manipulate data at rapid scale to improve their decision-making process, hence improve their overall business performance specifically. Now, Text Mining not only been used in Business Intelligence area, but also has been widely used in different business area namely Risk Management, Knowledge Management, Cybercrime Prevention, Fraud Detection, Contextual Advertising and Social Media Data Analysis. [1]

In this paper, we will discuss how Text Mining can assist us doing analysis on negative and positive user responses to Movie Database.

1. Objective

Objective why you select that three (3) methods

1. Naive Bayes

Naïve Bayes is the most applicable algorithms for text mining. Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naive Bayes classifiers is based on Bayes’ theorem, and the adjective naive comes from the assumption that the features in a dataset are mutually independent. In practice, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well under this unrealistic assumption [2].

1. J-48

The J48 Decision tree classifier follows the following simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained [3].

1. Random Forest

Random forest is mainly used in the text mining area. This algorithm is used in text classification because it can performed well in highly dimensional feature spaces for example textual data. When building decision trees, it is important to consider also the classifier’s feature selection since the dimensionality of textual data will cause excessive detailing risk to increase [4].

1. Scope

The dataset consist of review that can be categorized into two types which are positive review and negative review. All the data in the dataset are in .txt format.

1. Aim

The aim of this study is to determine which algorithms is the best to predict negative or positive movie review from our dataset.

1. Tools

We are using WEKA to do our data mining process. WEKA is a program that contain various of algorithms of machine learning to do data mining operations. The application writing in java langue and build in with all algorithms that can use it on your dataset directly or embedded in your system. WEKA has a lot of tools for data pre-processing, classification, regression, clustering, association rules, and visualization. And also, well fit developer who develop a new machine learning schemes. Weka is open source software issued under the GNU General Public License. [2]

1. Steps

Install WEKA from the official website then prepare our sample data (unzip the data).

Then we run WEKA and choose “simple CLI”.

After the window opened, we run this command in simple CLI command line:

java weka.core.converters.TextDirectoryLoader -dir "Our data directory" > "Our intended data directory \movie.arff"

Next step we pre-process data and save the processed one by choosing explore from the main GUI and load the arff file using “open file” option then pick “stringToWordVector” filter and hit “apply”.

We then delete unwanted words using “pattern” and click “remove” and we would delete other unexpected words manually and hit save

For the final step we choose the classifiers what we wanted and run it.

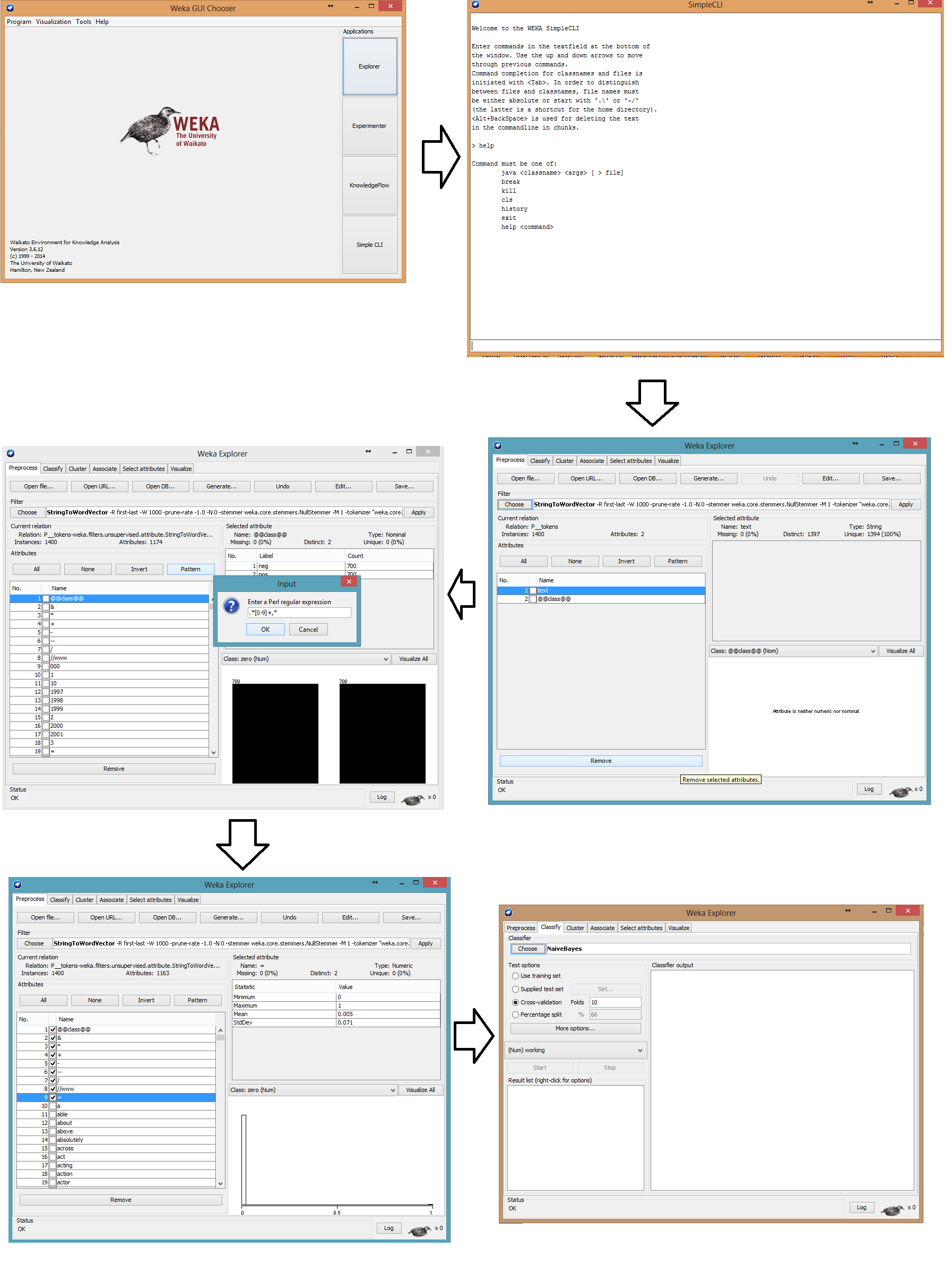


Figure 1 Pre-processing steps

1. Result

For the classification, three algorithms were used. The results of these algorithms are then compared to find out the best fit for the dataset. The algorithms used are:

1. Naive Bayes
2. J-48
3. Random Forest

The algorithms mentioned above are selected for implementation to find out the most accurate algorithm to predict negative or positive movie review from our dataset .The dataset consist of instances. The test option used for all the algorithm was using cross-validation with 10 folds.

1. **Naive Bayes**

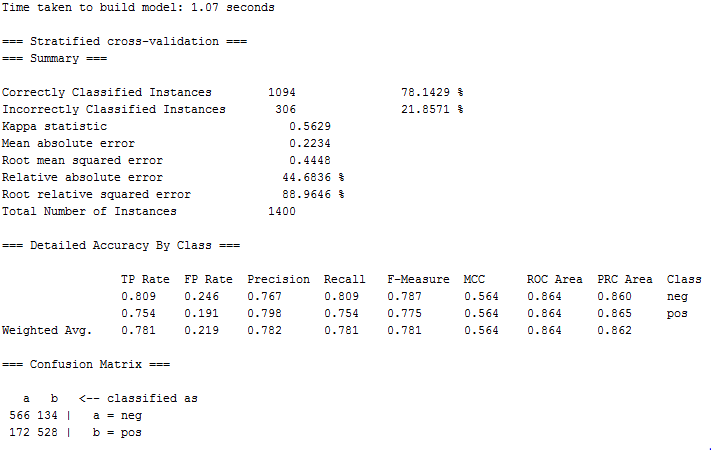


Figure 2 Naive Bayes result

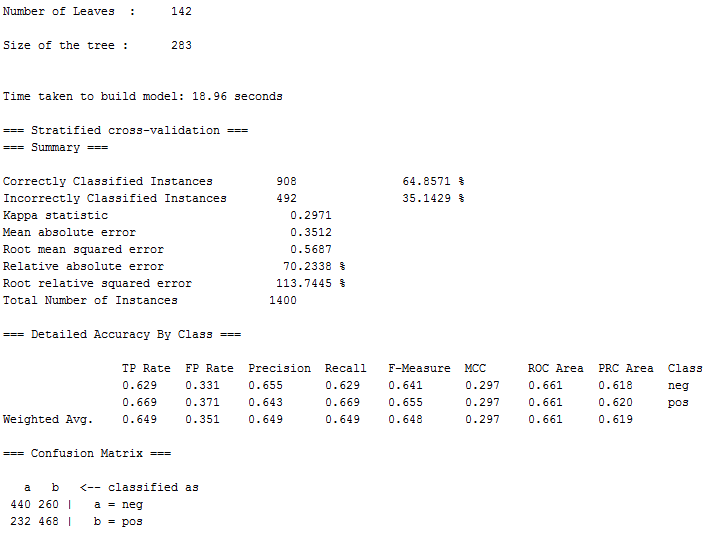
1. **WEKAJ-48**

Figure 3 Weka J-48 result

1. **Random Forest Tree**

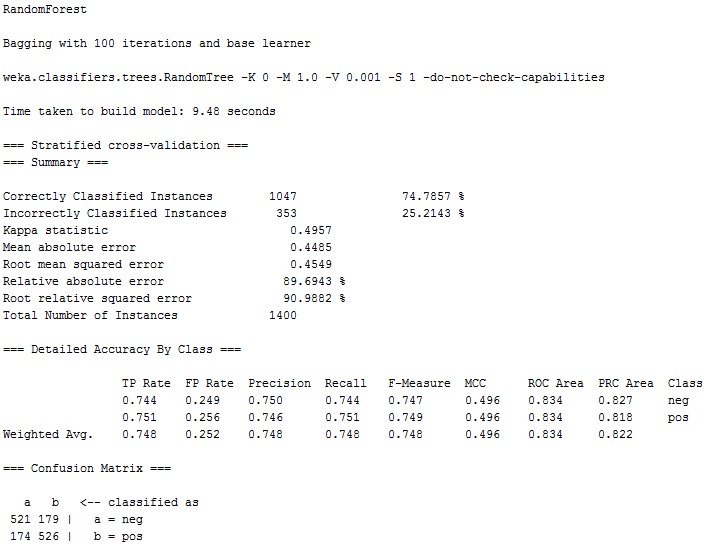


Figure 4. Random Forest Tree result

**Comparison between three algorithms:**

We will compare the three algorithms to find out the most accurate algorithm to classify movie review sentiment (negative or positive review) text analysis. We will compare the result using the confusion matrix, accuracy measures, error measures, and accuracy rate.

**Confusion Matrix**

Confusion matrix are obtained to find out the accuracy of the data and how many instances belong to which class. It also shows how many instances are correctly identified and which class they belong to. In this study, we have two classes hence we have confusion matrix of 2 by 2. The confusion matrix for each algorithm has been given below.

**Table 3: Comparison of confusion matrix between three algorithms.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Confusion Matrix** | | |
| **Naive Bayes** |  | a (Neg) | b (Pos) |
| a (Neg) | 566 | 134 |
| b (Pos) | 172 | 528 |
| **J-48** |  | a (Neg) | b (Pos) |
| a (Neg) | 440 | 260 |
| b (Pos) | 232 | 468 |
| **Random Forest** |  | a (Neg) | b (Pos) |
| a (Neg) | 521 | 179 |
| b (Pos) | 174 | 526 |

Based on the Table 3 above, we can conclude that Naïve Bayes algorithm shows the highest accuracy in identifying in which class the instance belongs. With 566 instances correctly classified to the negative (neg) class and 582 instances correctly classified to the positive (pos) class, it left only 306 instances incorrectly classified which is the lowest from all three algorithms.

**Accuracy Measures**

Accuracy measures table will be used to determine how accurate the model.

**Table 4: Comparison of accuracy measures between three algorithms.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **TP** | **FP** | **Precision** | **Recall** | **Class** |
| **Naive Bayes** | 0.809 | 0.246 | 0.767 | 0.809 | neg |
| 0.754 | 0.191 | 0.798 | 0.754 | pos |
| **J-48** | 0.629 | 0.331 | 0.655 | 0.629 | neg |
| 0.669 | 0.371 | 0.643 | 0.744 | pos |
| **Random Forest** | 0.744 | 0.249 | 0.750 | 0.816 | neg |
| 0.751 | 0.256 | 0.746 | 0.751 | pos |

From the Table 4 above, we can conclude that Naive Bayes has the highest accuracy among the three algorithms.

**Error Measures**

Error measure is to shows the degree of mistakes done by each algorithm. The smaller the error, the more accurate it will be.

**Table 5: Comparison of error measures between three algorithms.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Criteria** | **Classifiers** | | |
| **Naive Bayes** | **J-48** | **Random Forest** |
| Kappa Statistics | 0.5629 | 0.2971 | 0.4957 |
| Mean absolute error | 0.2234 | 0.3512 | 0.4485 |
| Root mean squared error | 0.4448 | 0.5687 | 0.4549 |
| Relative absolute error | 44.6836% | 70.2338% | 89.6943% |
| Root relative squared error | 88.9646% | 113.7445% | 90.9882 % |

From the Table 5 above, we can conclude that Naïve Bayes algorithm has the lowest error measure followed by Random Forest tree algorithm, while Weka J-48 algorithm has the highest error measure among three algorithms.

**Accuracy Rate**

This table shows the overall result of which algorithm is the best and achieved the highest accuracy to accurately classify movie review whether it is negative (neg) or positive (pos) from the dataset.

**Table 6: Comparison of accuracy rate between three algorithms.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Criteria** | **Classifiers** | | |
| **Naive Bayes** | **J-48** | **Random Forest** |
| Time to build model (in seconds) | 1.07 | 18.96 | 9.48 |
| Correctly classified instance | 1094 | 908 | 1047 |
| Incorrectly classified instance | 306 | 492 | 353 |
| Accuracy (%) | 78.1429% | 64.8571% | 74.7857% |

Based on Table 6 above, we can conclude that among the three algorithms, Naïve Bayes tree algorithm has the highest accuracy at 78.1429% .From above analysis that we have done, we can say that Naive Bayes is the best algorithm to be used in classify movie review whether it is negative (neg) or positive (pos) from the dataset.

1. Conclusion
2. References

[1] <http://www.expertsystem.com/10-text-mining-examples/>

[2] <http://sebastianraschka.com/Articles/2014_naive_bayes_1.html>

[3] <http://www.d.umn.edu/~padhy005/Chapter5.html>

[4] Segnini, Armando., & Motchoffo,J.J.T. C. . Random Forests and Text Mining. http://www.academia.edu