CSCanada
Management Science and Engineering
Vol. 10, No. 4, 2016, pp. 36-46
DOI:10.3968/8997

ISSN 1913-0341 [Print]
ISSN 1913-035X [Online]
www.cscanada.net
www.cscanada.org

The Data Mining Application Based on WEKA: Geographical Original of Music

WU Yuchen[a],*

[a]School of Economics and Management, Nanjing University of Science and Technology, Nanjing, China.
*Corresponding author.

Received 5 September 2016; accepted 2 November 2016
Published online 26 December 2016

Abstract

In this article, we use Weka as our tool for data mining. In the first step, we retrieve the dataset from the UCI database. At the same time, we get the purpose of analysis. Then we classify the datasets and found class attributes. We classify the datasets into two attributes: latitude and longitude. The second step, we finish the data cleaning. The tools used for data cleansing are Microsoft Excel, Google Maps, and Weka. The next is aggregation and Skewed Data. Then the appropriate attributes of dataset are selected. The third step, it's the experiment design. We choose three classifiers: Naive Bayes, J48 and IBk. The fourth step, we finally get the experimental results through the appropriate classifiers, and the results are summarized. The fifth step, to make ROC Curve. The sixth step, the analysis of final results. Of three aspects: classifier analysis, attribute analysis, noise analysis. In the seventh step, we get the final conclusion that IB1 is the most successful model for our dataset.

Key words: Data mining; Weka; Naïve Bayes; J48; IBk; IB1; ROC curve

Wu, Y. C. (2016). The Data Mining Application Based on WEKA: Geographical Original of Music. *Management Science and Engineering*, *10*(4), 36-46. Available from: URL: http://www.cscanada.net/index.php/mse/article/view/8997
DOI: http://dx.doi.org/10.3968/8997

1. DATA BACKGROUND

1.1 Origin of the Data

The data set we used for this project can be found in the UCI Machine Learning Repository webpage. The donor of the data builds the data set from his personal collection of music. These music tracks were then ran through a program called MARSYAS which extracts timbal information from the entire length of the track and produces 68 variables that describe the music track. The geographical location was manually collected from the CD sleeve notes and when that information was not adequate, additional research was conducted to determine the country of origin for the music tracks.

Figure 1
UCI Database

1.2 Purpose of Analysis

The purpose of this analysis is to try to see if we are able to predict the country of origin based upon the timbal information from the music tracks. We want to see if there is enough geographical influence in the aspects of the music tracks so that the origin can be predicted.

Figure 2
World Map

1.3 Dataset & Attributes

A total of 1,059 different music tracks were used and the tracks are from 33 different countries. Each track has 68 numerical attributes describing the music within the track.

Since each of our attributes is numeric, we cannot apply to any domain knowledge to the attributes to attempt to determine which ones may be most useful or which ones could possibly be a false predictor. Below is a short snap shot of some of the data we were working with, you can see that it is very difficult to make any decisions about which attribute could be more useful since to the naked eye, they all appear to be similar.

60	61	62	63	64	65	66	67	68
-0.96078	-0.81934	-0.64179	-0.85715	-1.15952	-0.7399	-0.76741	-0.64977	-0.80107
-0.01847	-0.37545	0.011739	-0.30319	-0.78865	-0.78479	-0.57886	-0.82015	-0.67505
0.214693	0.090279	-1.03806	-1.07665	-0.39777	-1.05544	-1.07083	-0.596	-0.55327
-1.18288	-0.65913	-1.13421	-1.05795	-1.20566	-1.12599	-0.99384	-0.62636	-1.06299
-0.60507	0.013224	-0.29201	-0.08119	-0.91929	-0.70733	-0.58923	-0.52113	-0.4518
-0.34957	-0.15827	-0.0476	-0.76129	-0.3009	-0.73087	-0.48464	-0.22976	-0.55256
0.675759	-0.58059	0.21062	-0.6701	-0.37668	-1.00109	-0.60123	-0.75073	-0.84815
-0.72329	0.456281	-0.27062	-0.14264	-0.02193	-0.62615	-0.53767	-0.84146	-0.61708
0.182293	0.091129	-0.84859	-1.03617	-0.60443	-1.05014	-0.88894	-0.63	-0.76629

Figure 3
68 Attributes

1.4 Class Attribute

The class attribute, what we are attempting to predict with the musical information from the data set, is the country of origin. This is presented in the data set as two different numerical attributes, the latitude and longitude which tell you the country of origin. This presented a problem and was something that we addressed in our Data Cleaning stage.

1.4.1 Data Cleaning

> Data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Used mainly in databases, the term refers to identifying incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and then replacing, modifying, or deleting this dirty data or coarse data.[1]

1.4.2 Data Cleaning Tools

We used three tools to help us do the cleaning to our data set:

We used three tools to clean the data:

a) Microsoft Excel
b) Google Maps
c) Weka

1.5 Aggregation

Aggregation is combining two or more attributes into a single attribute for the purpose of data reduction, change of scale, and more "stable" data. Our first step in the data cleaning process was to combine the two original class attributes of "Latitude" and "Longitude" into the country in which that point was located.[2] This was done by entering the coordinates into Google Maps and recording the results to Excel.

[1] Data Cleansing. Wikipedia. 2016. Reteived from http://en.wikipedia.org/wiki/Data_cleansing.

[2] Sikora, Riyaz. (2016). *Data and Input Concepts*, Lecture 2 [PowerPoint slides]. Retrieved from elearn.uta.edu.

Figure 4
Three Tools

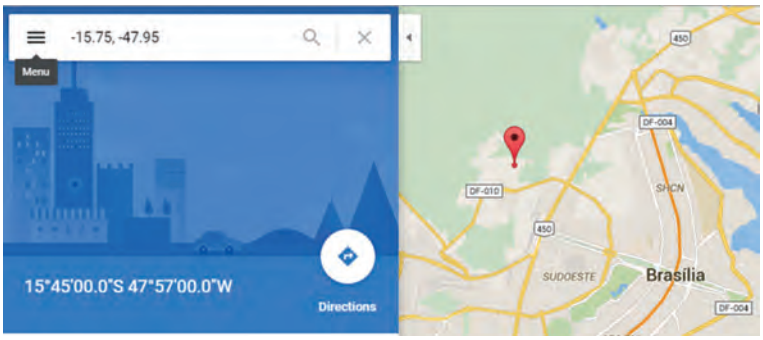


Figure 5
An Example for Using of Latitude and Longitude

The class attribute became "Country." The dataset now consisted of 69 attributes and 1,059 instances with 33 class values. We ran ZeroR to establish a baseline by which we could determine if our preprocessing was having a positive or negative impact on the output. The result was a prediction of India with **6.1321%** accuracy.

67	68	Latitud	Longitu
91477	-0.83625	-15.75	-47.95
.1236	1.391141	14.91	-23.51
75763	1.063847	12.65	-8
01233	-0.3922	9.03	38.74
71838	1.289783	34.03	-6.85
50619	-0.00847	12.65	-8
99507	-0.87273	12.65	-8
63644	-0.42621	14.66	-17.41

67	68	Country
-1.05768	-0.9035	Iran
-0.70411	-0.5318	Indonesia
-1.04845	-1.08268	Japan
-1.0302	-0.9412	Italy
-0.87307	-0.85509	India
-0.84035	-0.94644	Morocco
-0.74554	-0.70253	Senegal
-0.94057	-0.82586	Morocco

Figure 6
69 Attributes

1.6 Skewed Data

The original data was heavily skewed because of the large number of class variables and the uneven distribution of instance classification.

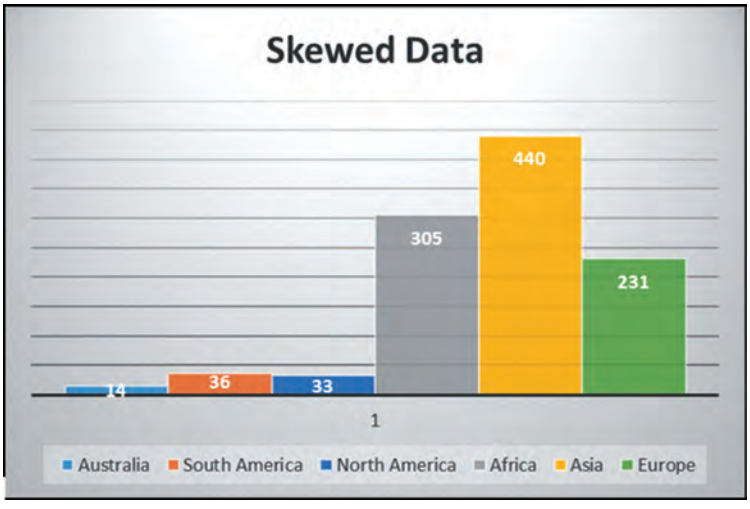


Figure 7
Skewed Data

We knew that reducing the number of class values would improve the accuracy rate, so we modified the class attribute to identify the continent from which the music originated. This left us with six class values, rather than the 33 we originally created. This improved our accuracy for ZeroR to **35.8491%**.

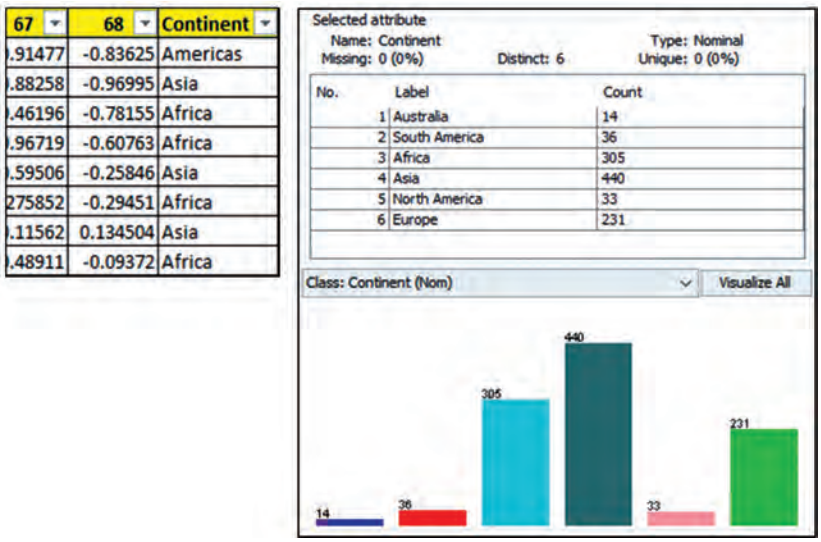


Figure 8
New 6 Class Values

Because the accuracy rate was still low, we continued reducing the number of class attributes. Very few instances were classified as Australia, so they were removed from the dataset. Similarly, North and South America accounted for a small number of instances, so they were combined into the class "Americas." We were left with 69 attributes and 1045 instances. The accuracy of ZeroR continued to improve to 43.25%.

Even though the results continued to improve by combining class variables, we did not have a music expert to consult regarding geographical similarities in music. So we continued to manipulate the data. Each attribute was also heavily skewed and/or had multiple outliers. This issue was addressed by placing minimum and maximum values on each attribute.

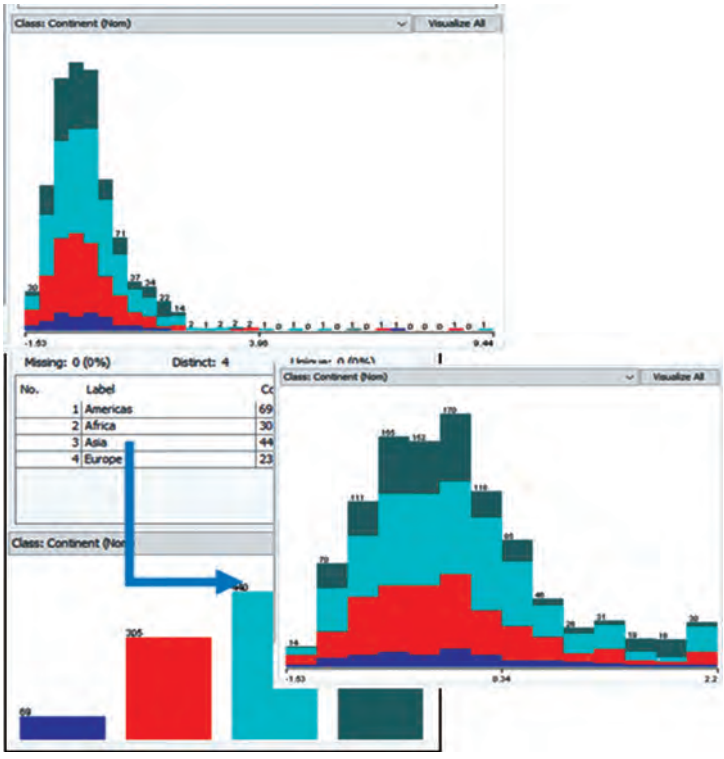
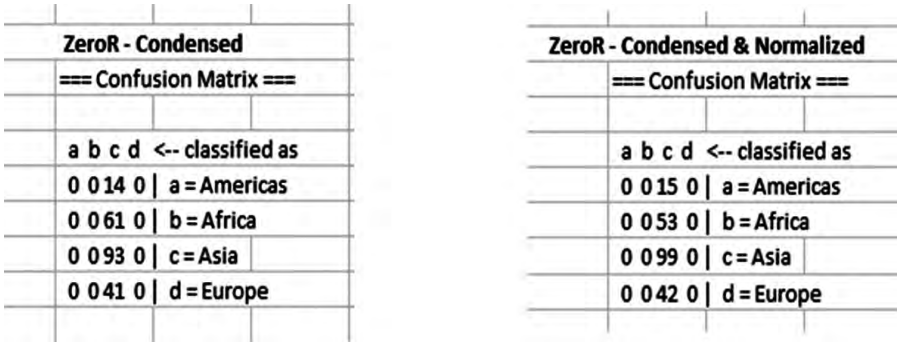


Figure 9
Normalized Data

The accuracy of ZeroR on this data was 43.21%, which was not an improvement over the previous version of the data. At this point, we had to compare the Confusion Matrix for each dataset to decide which one we would proceed with. ZeroR with the normalized data correctly predicted the majority class (Asia), as well as the minority class (Americas), more times than it did the previous version of the dataset, so we decided to continue with the normalized data.



ZeroR - Condensed

=== Confusion Matrix ===

a	b	c	d	<-- classified as
0	0	14	0	a = Americas
0	0	61	0	b = Africa
0	0	93	0	c = Asia
0	0	41	0	d = Europe

ZeroR - Condensed & Normalized

=== Confusion Matrix ===

a	b	c	d	<-- classified as
0	0	15	0	a = Americas
0	0	53	0	b = Africa
0	0	99	0	c = Asia
0	0	42	0	d = Europe

Figure 10
Comparison of the Confusion Matrix

At this point because the class values were still unequally distributed, we decided to add 100% SMOTE to the data. SMOTE stands for Synthetic Minority Oversampling Technique and is "used to adjust the relative frequency between minority and majority classes in the data."[3] This increased our number of instances from 1045 to 1114, but the ZeroR accuracy decreased to 40%.

We applied SMOTE again, bringing the total to 200%. This caused the accuracy to decrease even more to 39%. Since this was close to the accuracy rate for 100% SMOTE, we compared the Confusion Matrix again. 100% SMOTE correctly predicted the majority class more frequently than the 200% SMOTE.

1.7 Attribute Selection

Since the highest accuracy we had recorded to this point was only 43.25%, and the general characteristics of the data were not immediately apparent to a non-musical expert, we decided to use the wrapper method to try to determine the important attributes.

The attribute evaluators used were the CfsSubsetEval, Classifier SubsetEval Naïve Bayes, Classifier SubsetEval J48, and IB1.

2. EXPERIMENT DESIGN

2.1 Classifier Prediction

The classifiers was selected based on two criteria:

- High prediction accuracy rate
- High stability of prediction model – low deviation and relative high accuracy rate

We chose the following classifiers for our experiment design:

- **Naïve Bayes** classifier applies Bayes Theorem, a probabilistic framework to solve classification problems, and strongly (naïve) assumes the independence between the attributes. Naïve Bayes works quite well even though there is violation of independence assumption. The reason is because it can make the correct class prediction based on the maximum probability, not the exact probability. In addition, Naïve Bayes can produce quite stable prediction.
- **J48** is a decision tree rule classifier. It builds decision trees from a set of training data using information entropy concept. At each node, the tree will decide to split at the attribute with higher information gain (rich in data). That means if one attribute has an outstanding class, it stops splitting and predicts that class at that node. J48 is well-known for its high accurate model building.
- **IBk** is an instance-based learning algorithm, aka lazy algorithm because it stores the training instances and does not actually learn anything but base its prediction on the near neighborhood's class. IBk predicts class on k-nearest neighbors' classifier. If k=1, it is called the basic nearest-neighbor instance-based learning. The nearest neighbor is determined by the training instance closest in Euclidean distance to the given test instance. The testing instance, hence, is predicted the same class as this training instance. Normally, increasing k continuously will increase the accuracy rate. However, it is not.
- The case for our data set.

[3] Frank, Eibe, Mark A. Hall, Ian H. Witten.

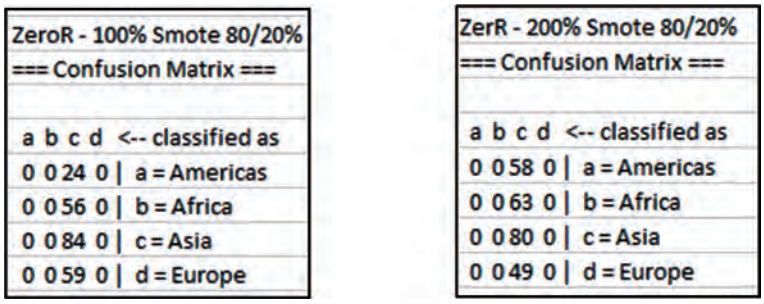


Figure 11
Comparison of Different SMOTE Parameters

For IBk, we tried to increase k while checking on the accuracy rate but it was not getting better as can be seen in the table below, therefore, we decided to select IB1 for the main experiment.

Table 1
Estimated Results of IB1

		IBk					
		C1	C2	C3	C4	C5	C6
Average values of 10 runs	IB1	65.25	64.08	62.62	62.47	56.41	53.34
	IB2	61.93	62.47	59.22	60.47	57.65	55.91
	IB3	62.91	63.63	60.00	61.59	53.44	52.37
	IB4	63.00	64.35	60.18	62.74	55.67	54.81
	IB5	63.32	64.44	60.52	63.30	56.12	53.30
	IB6	62.20	63.81	59.82	63.21	55.70	54.39

2.2 Classifier Selection

Our experiment design consists of 3 factors which make 6 cases in total as the table below.

Factor 1: Percentage split

Factor 2: Number of attributes

Factor 3: Noise

Factor design is used to determine which specific case will generate the highest accuracy rate. Note that "10% Noise" is only applied for "All attributes". Percentage split 80/20 means that classification result will be evaluated on a test set which is 20% of the original data. Sometimes, not all the attributes are important and support the prediction, they could even be false predictors. Therefore, removing them of the attributes will help improve the prediction accuracy. Lastly, noise:

C1 = All 69 attributes + Percentage split 80% training set-20% testing set
C2 = 28 selected attributes + Percentage split 80% training set-20% testing set
C3 = All 69 attributes + Percentage split 60% training set-40% testing set
C4 = 28 selected attributes + Percentage split 80% training set-20% testing set
C5 = All 69 attributes + Percentage split 80% training set-20% testing set + 10% Noise
C6 = All 69 attributes + Percentage split 60% training set-40% testing set + 10% Noise

Table 2
Parameter C Settings

	All attributes	Selected attributes	All attributes+10%noise
Percentage split	C1	C2	C5
Percentage split (60%/40%)	C3	C4	C6

3. EXPERIMENT RESULTS

3.1 Results for Each Classifier

Table shows the total of 18 experiments for 6 cases using 3 classifiers. Note that we run each experiment ten times and take the average values in order to avoid the sampling errors.

The result of the test runs are shown below for each classifier:

Table 3
Results for Each Classifier

Results for each classifier
E1=Performance for Naïve Bayes when,All 68 Attributes+Percentage Split of 80%:20%
E2=Performance for Naïve Bayes when,All 68 Attributes+Percentage Split of 60%:40%
E3=Performance for Naïve Bayes when,Selected 28 Attributes+Percentage Split of 80%:20%
E4=Performance for Naïve Bayes when,Selected 28 Attributes+Percentage Split of 60%:40%
E5=Performance for Naïve Bayes when, All 68 Attributes+Percentage Split of 80%:20%+10% Noise
E6=Performance for Naïve Bayes when, All 68 Attributes+Percentage Split of 60%:40%+10% Noise
E7=Performance for J48when,All 68 Attributes+Percentage Split of 80%:20%
E8=Performance for J48when,All 68 Attributes+Percentage Split of 60%:40%
E9=Performance for J48when,All Selcted 28 Attributes+Percentage Split of 80%:20%
E10=Performance for J48when,All Selcted 28 Attributes+Percentage Split of 60%:40%
E11=Performance for J48when,All 68 Attributes+Percentage Split of 80%:20%+10% Noise
E12=Performance for J48when,All 68 Attributes+Percentage Split of 60%:40%+10% Noise
E13=Performance for IB1when,All 68 Attributes+Percentage Split of 80%:20%
E14=Performance for IB1when,All 68 Attributes+Percentage Split of 60%:40%
E15=Performance for IB1when,Selected 28 Attributes+Percentage Split of 80%:20%
E16=Performance for IB1when,Selected 28 Attributes+Percentage Split of 60%:40%
E17=Performance for IB1when, All 68 Attributes+Percentage Split of 80%:20%+10% Noise
E18=Performance for IB1when, All 68 Attributes+Percentage Split of 60%:40%+10% Noise

Table 4
Results of Naïve Bayes, J48 and IB1

			Naïve Bayes			
Seed	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
1	39.91	49.78	41.26	46.86	35.87	37.67
2	39.01	40.36	39.91	47.31	33.18	36.55
3	46.64	54.26	43.50	52.02	41.26	39.46
4	40.81	47.09	39.69	48.65	38.57	37.67
5	47.09	49.78	43.27	48.65	41.70	40.81
6	46.64	49.78	45.74	49.78	37.67	39.01
7	44.39	48.88	44.62	50.45	39.46	41.48
8	38.12	51.57	39.01	48.21	36.77	36.77
9	43.50	47.98	41.93	47.31	38.12	36.55
10	38.57	46.64	39.24	45.74	35.87	35.65
Average	42.47	48.61	41.82	48.50	37.85	38.16
Stand Dev	3.41	3.46	2.26	1.76	2.46	1.85
			J48			
Seed	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
1	50.67	49.33	46.41	46.41	44.39	43.27
2	55.16	56.50	51.12	50.67	37.67	45.52
3	48.43	52.02	48.21	48.21	39.01	46.64
4	52.02	58.74	44.39	48.21	42.60	44.62
5	47.98	58.74	46.86	52.24	37.67	46.41
6	51.12	52.02	53.59	51.35	39.46	47.98
7	48.43	50.67	48.21	45.96	45.74	43.27
8	49.33	50.67	49.55	50.22	44.39	43.50
9	51.12	50.67	47.98	51.79	42.15	41.70
10	51.57	46.19	48.65	50.00	47.53	39.46
Average	50.58	52.56	48.50	49.51	42.06	44.24
Stand Dev	2.05	3.93	2.41	2.09	3.31	2.41

To be continued

Continued

	IB1					
Seed	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
1	65.02	67.26	61.88	65.70	55.16	54.04
2	65.47	64.57	61.21	59.42	56.05	53.81
3	65.02	60.54	62.11	59.64	56.05	54.26
4	65.92	67.26	63.00	64.13	56.50	53.59
5	68.61	65.47	65.70	65.70	59.64	56.73
6	66.82	65.02	63.45	61.66	57.85	52.91
7	58.74	61.88	61.44	61.44	51.57	53.36
8	70.40	65.02	65.92	64.80	59.19	50.22
9	59.19	56.95	58.74	57.62	52.91	50.22
10	67.26	66.82	62.78	64.57	59.19	54.26
Average	65.25	64.08	62.62	62.47	56.41	53.34
Stand Dev	3.52	3.16	2.01	2.76	2.56	1.83

3.2 Summary of Results

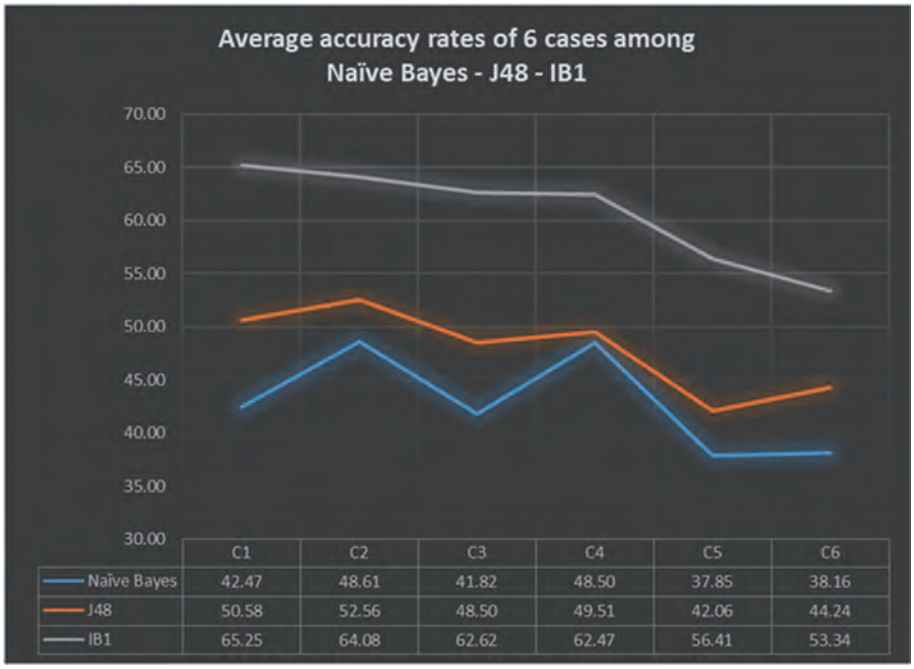


Figure 12
Comparison for Each Classifier of Average Accuracy Rates

As can be seen in the accuracy graph, IB1 outperformed Naïve Bayes and J48 in all cases. The best result it can reach is 65.25 with "all attributes" tested on 80% training sets and 20% of testing set. Meanwhile, the result for "Selected attributes" cases (C2, C4) got slightly worse because there were very few attributes that were identified as important across all of the algorithms we used. The CfsSubsetEval chose only 28 out of 69 attributes as important. We think that all attributes have the relatively same importance level. Next, logically, using 60% training set and 40% testing set resulted in lower rate. As noted in Data mining book, adding noise is very sensitive for Instance-based learning algorithm. That might explain the significant decrease in case 5 and 6 where we added 10% Noise.

At the same time, Naïve Bayes and J48 prediction rate behaved quite similarly. Naïve Bayes ranged from 37.85% to 48.61% while it was 42.06 to 52.56 for J48. Across the experiments, "Selected attributes" clearly improved the estimation rate for both. Adding Noise, again, not a good factor to leverage the accuracy at all.

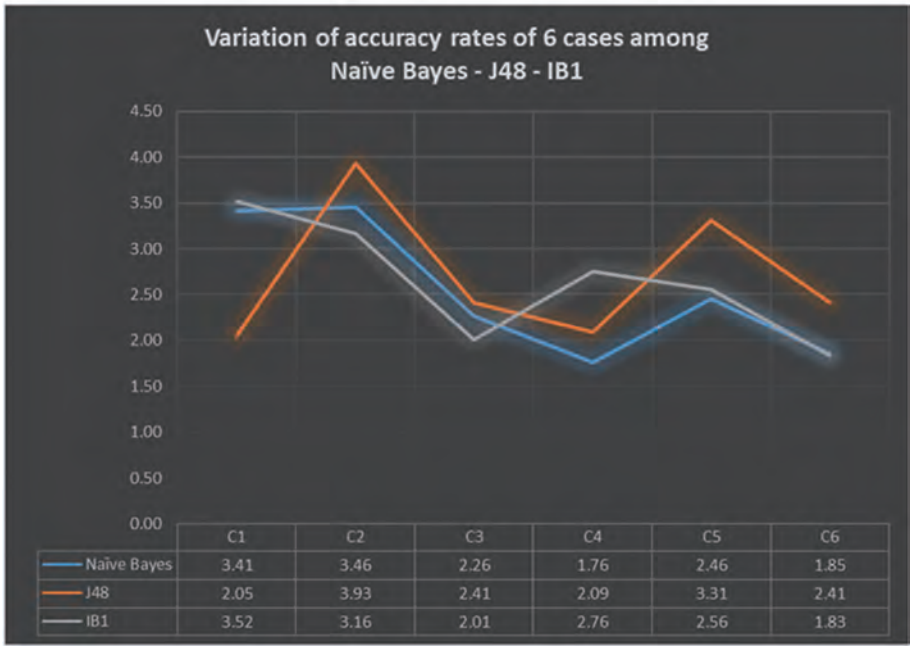


Figure 13
Comparison for Each Classifier of Variation of Accuracy Rates

The deviation between the cases fluctuated remarkably at high figures. That means the factors play vital role in their performance. J48, as known as highly accurate model, definitely scarified its stability with the highest differences among cases. In the meantime, Naïve Bayes and IB1 kept closed to each other with overall lower deviation.

3.3 Confusion Matrix

Since the accuracy rates for C1, C2, C3 and C4 are similar, we analyze the confusion matrix to evaluate each case.

E13

```
=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 29  1  1  1 |  a = Americas
  4 35 15 10 |  b = Africa
  5 11 55  9 |  c = Asia
  2  9  8 23 |  d = Europe
```

E15

```
=== Confusion Matrix ===

  a   b   c   d   <-- classified as
 52   2   1   1 |   a = Americas
 16  62  30  14 |   b = Africa
 15  26 111  20 |   c = Asia
  4  12  24  46 |   d = Europe
```

E14

```
=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 19  1  0  1 |  a = Americas
  6 30 18  9 |  b = Africa
  3 15 63  8 |  c = Asia
  2  5 10 28 |  d = Europe
```

E16

```
=== Confusion Matrix ===

  a   b   c   d   <-- classified as
 48   4   9   2 |   a = Americas
  7  69  36  13 |   b = Africa
 11  21 108  20 |   c = Asia
  2  15  24  47 |   d = Europe
```

IB1					
		C1	C2	C3	C4
	Accuracy rate	65.25	64.08	62.62	62.47
Confusion	Majority accuracy rate	69.6	70.8	64.5	67.5
Matrix	Minority accuracy rate	90.6	90.5	92.8	76.2

Figure 14
Final Results of IB1

IB1 case 1 still reasonably scored the best among all.

4. ROC CURVE

Asia - All Attributes - Smote & Randomize - Naïve Bayes (Blue and orange) AND IB1 (orange only) - 80%/20%

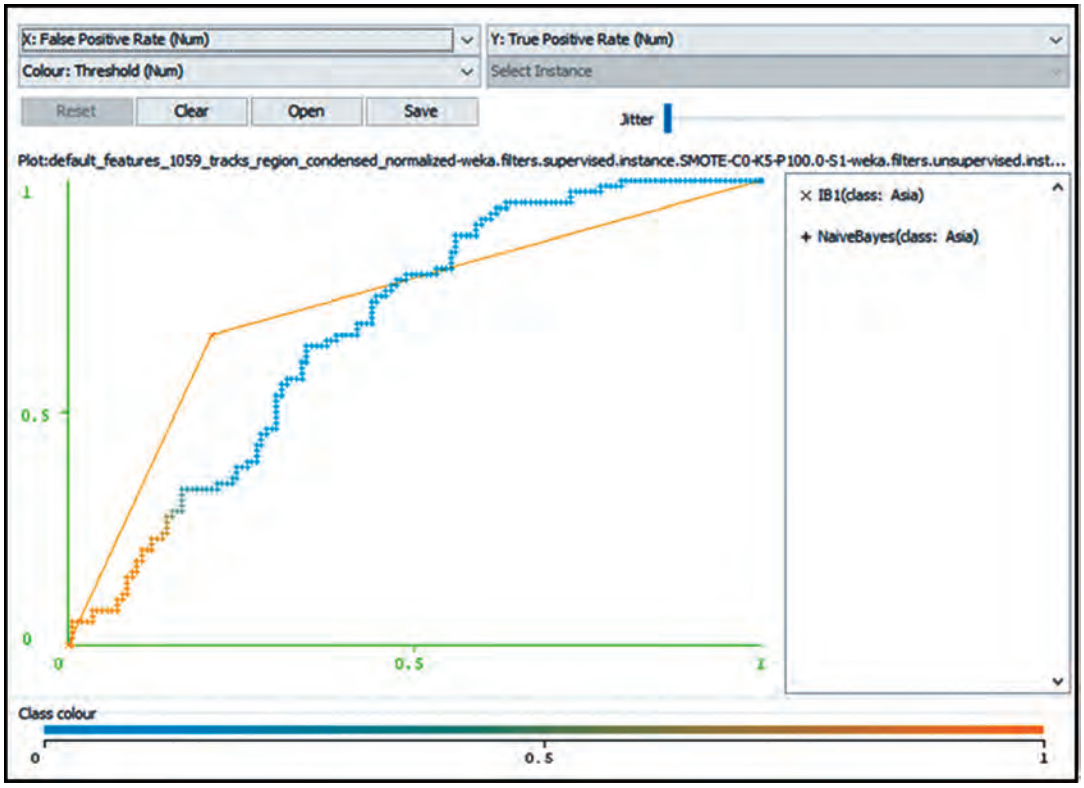


Figure 15
ROC Curve

As can be observed, the area under the ROC Curve is larger for IB1 than Naïve Baye.

5. ANALYSIS

5.1 Classifier Analysis

From section 4.1, we picked the highest accuracy rate and the lowest variance for each of three classifiers to present it in the table below. It is clearly seen that IB1 got the best accuracy rate 65.25%. The variance, meanwhile, was quite similar among three of them. In fact, IB1's was not as stable as Naïve Bayes' but maintained a reasonable rate of 2.01 which was still better than J48's.

Therefore, based on classifier analysis, IB1 builds the best models with the highest accuracy and relatively stable variance.

Table 5
Comparison of Final Results for Each Classifier

Classifier	Highest accuracy	Lowest variance
Naïve Bayes	48.61	1.76
J48	52.56	2.05
IB1	65.25	2.01

5.2 Attribute Analysis

Our second factor is number of attributes. We need to identify if selected attributes help to improve the accuracy rate. Testing only the selected attributes did improve prediction capability for Naïve Bayes noticeably, 6%-7%. It applied the same effect on J48, however, at merely 1%-2%. Surprisingly, it did not work well for IB1 as the accuracy slightly decreased by 1%-2%. Overall, selected attributes help to improve accuracy rate for Naïve Bayes and J48 but they are still much lower than IB1.

5.3 Noise Analysis

Concerning noise factor, to avoid the complication of cases combination, from the beginning, we only added 10% Noise to All Attributes, not Selected Attributes. Noise is more effective for Naïve Bayes and J48 rather than IB1. Instance-based learning, as in theory, is very sensitive to noise. That might be the reason why the accuracy rate for IB1 dropped slightly. Similarly to the impact of number of attributes on the three algorithms, adding noise did not significantly improve prediction performance for Naïve Bayes and J48 to outperform IB1.

CONCLUSION

The following is conclusion we can get results. The average accuracy for IB1—C1 is 65.25%, which is much higher than other algorithms. And Naive Bayes has the least Standard deviation in the condition of C1, C4 and C5, but lower accuracy rate than IB1. We also test the influence of noise. Adding noise will affect accuracy rate, and IB1 is the most influenced algorism compared with others, but still higher than other algorisms. 80/20 percentage split in training/test data will improve the performance of most algorithms including IB1, J48, and Naïve Bayes. And about the lowest variance, we can get that Naïve Bayes has the least number. The number of IB1 is 2.01, it's the second least number. In the end, IB1 was the most successful model for our dataset.

REFERENCES

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861-874. doi:10.1016/j.patrec.2005.10.010

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). Prentice Hall.

Swets, J. A. (1996). *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*. Mahwah, NJ: Lawrence Erlbaum Associates.

Wu, S. (2013). A review on coarse warranty data and analysis. *Reliability Engineering and System, 114,* 1-11.