Data Mining Final Project:

Bank Marketing Campaigns

Zhixin Liu

Professor Dejing Dou

CIS 453

March 18, 2016

TABLE OF CONTENTS:

Abstract

The purpose of the study is to figure out the potential strong association rules hidden in the Bank Marketing dataset, as well as discover the relationship between the successful marketing product and certain specific customer information. The data set used for research is based on a marketing campaigns data record of a Portuguese banking institution. Basic data mining process would be utilized in this project. The project firstly filter the meaningful and related data attributes and data instance from the original research data set by proceeding data preprocess including data clean, data transformation and data reduction. Later on classification method would be used for the filtered data instance throughout data mining process. By testing multiple classification method on the data set, the project would compare the data mining classification results obtained via various methods, summarize the difference and similarity among these results and analyze the accuracy of different classification method used for the data classification process.

Problem

How to promote a business marketing product at a lowest possible overhead is always viewed as the main problem by the manager. The most significant part of the problem is to identify the promising and potential customers of the marketing product with restricted information. Knowing that specific information which decides the promising and potential customer would allow manager to put more resource on positive part towards the product and reduce the cost spent on non-promising customer, so that to eliminate bottlenecks and create a more efficient promoting path.

Data Set Description

- ***Data Information***

The data set used in the project is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Basically the data set consist of 20 different attributes and at most 45211 different instances for research. Note that more than one contact to the same client was required, in order to access if the product like bank term deposit would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe a term deposit.

- ***Data Attributes Description***

Data Attributes viewed as Input variables is showed as follow:

- *Bank client data:*

1. age, which means the customers' age;
2. job type, including: blue collar, entrepreneur, housemaid, management, retired, etc.
3. marital status, including: divorced, married, single, unknown;
4. education level, including: high school, university degree, illiterate, unknown, etc.
5. default, which means whether the customer has credit in default.
6. housing, which means whether the customer has housing loan.
7. loan, which means whether the customer has personal loan.

- *Related with the last contact of the current campaign:*

8. contact, contact communication type including: cellular, telephone;
9. month, which means last contact month of year
10. day_of_week, which means last contact day of the week

11. duration, which means last contact duration, in seconds; Notice that this attribute highly affects the output target.

- *Other attributes:*

12. campaign, which is the number of contacts performed during campaign for this client;
13. pdays, which is the number of days that passed by after the client was last contacted from a previous campaign;
14. previous, which is the number of contacts performed before campaign for this client;
15. poutcome, which is the outcome of the previous marketing campaign;

- *Social and economic context attributes:*

16. emp.var.rate, which means employment variation rate, quarterly indicator;
17. cons.price.idx, which means consumer price index, monthly indicator;
18. cons.conf.idx, which means consumer confidence index, monthly indicator;
19. euribor3m, which means euribor 3 month rate, daily indicator;
20. nr.employed, which means the number of employees, quarterly indicator;

- *Output variable (desired target):*

21. result y, which means whether the client subscribed a term deposit.

Data Preprocessing

- ***Mining Tool used***

In this project, we use the Weka application tool for data processing and data mining.

- ***Preprocessing***

Considering the raw data of attributes who are sparse data, Discretize method should be utilized to combine those sparse data into an acceptable one. Those attributes including:

age, duration, campaign, plays, previous, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed. During this Discretize process, we merge the sparse data of one specific attribute into a five bins data with equal frequency. For example, the age data is showed as Fig-1

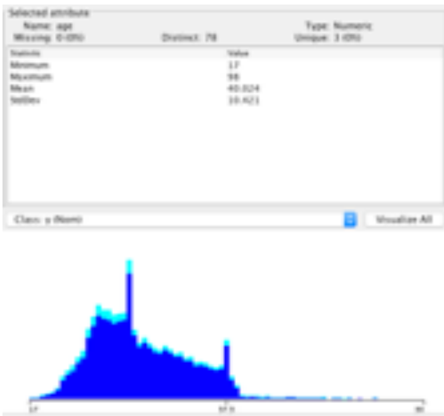

Fig-1

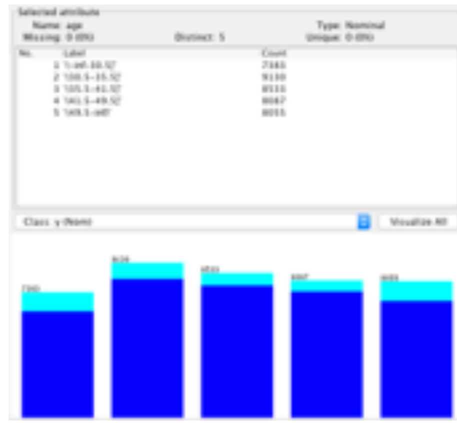

Fig-2

Here we do not care about each situation of specific age person, what we need to consider about is a group of people who are in specific age range. Under this circumstance, we apply Discretize method and get the acceptable result as Fig-2.

After that, in order to get most valuable and related attributes, we consider applying Select Attributes of Information-Gain-Attributes-Evaluation to filter those attributes who are not related to or have weak relationship towards the event of the client subscribes a term deposit. Under this context, we get the information-gain table of attributes result as Fig-3.

Based on the Fig-3 result table and real situation, we remove five attributes with lowest Information-Gain. These removed attributes contain: day_of_week, marital, loan, housing, education. Then we obtain the new selected attributes used for data classification, which are showed as the followings: {duration, euribor3m, cons.price.idx, cons.conf.idx, nr.employed, emp.var.rate, pdays, poutcome, month, previous, age, contact, job, default, campaign}.

InfoGainAttributeEval	
0.1094127 11 duration	0.0184393 1 age
0.1025698 19 euribor3m	0.0168012 8 contact
0.0980036 17 cons.price.idx	0.0142231 2 job
0.0976284 18 cons.conf.idx	0.0083306 5 default
0.0896296 20 nr.employed	0.0042851 12 campaign
0.0785864 16 emp.var.rate	0.0034476 4 education
0.0444837 13 pdays	0.0020686 3 marital
0.0438343 15 poutcome	0.0004645 10 day_of_week
0.0380966 9 month	0.0000997 6 housing
0.0277329 14 previous	0.0000193 7 loan

Fig-3

After applying above actions, we now have removed meaningless attributes and picking out specific potential useful attributes, then we consider applying the classification processes.

Classification Processing:

In the classification processing part, we would apply the filtered data obtained in data preprocessing into three different classification algorithms. By comparing the difference and similarity among those results based on the three algorithms, we then find the convinced result of the most meaningful attributes that decide the promising customer subscribing a term deposit. The three algorithm are respectively: J48 algorithm, Naive Bayesian Classification, and Neural Network Algorithm.

- ***Decision Tree Algorithm: J48***

Let's start with the decision tree algorithm J48. J48 is Java version implement of C4.5 algorithm for generating decision tree. It follows the ID3's way on building the decision tree from a set of training data through using information entropy. During the tree generation process of selecting node, J48 chooses the specific attribute of the data with the highest normalized information gain. Different from ID3, J48 has multiple improvements towards ID3, which includes:

- By creating a threshold to divide the list into multiple sublist according to whether its attribute value is larger or less than the threshold, J48 could handle both continuous and discrete attributes.
- By allowing attribute values to be marked as missing, J48 could handle training data with missing attribute values, so that missing attribute values are simply not used in gain and entropy calculations.
- Handling attributes with differing costs, as well as pruning trees after creation.

Under this context, we apply the previous filtered attributes with values into the J48 algorithm. Here, we split the all data into two part, where one part containing 70% data is used for generating decision tree, while another part containing 30% data is used for testing decision tree. Therefore, we then get the result as follows:

Correctly Classified Instances	Incorrectly Classified Instances	Time taken to build model
90.7575% (11214)	9.2425% (1142)	0.25s

For detailed accuracy by class under the weighted average, the result is showed as follow:

TP Rate	FP Rate	Precision	Recall	F-Measure
0.908	0.536	0.896	0.908	0.896

According the result table, we can clearly see the decision tree generated by J48 algorithm has a high accuracy on correctly classified instances. In detailed, the whole process of running J48 algorithm to generate decision tree totally takes 0.25 seconds. And in the 30%

testing data which has totally 12356 instance, there are 11214 instances which takes 90.7575% of all testing data can correctly classified based on the decision tree, while there are 1142 instances which takes 9.2425% of all testing data cannot correctly classified based on the decision tree. The high accuracy of correctly classified instances indicates the decision tree works well on classify the whole data value, thus it could provide us with good classification rules which helps on deciding the promising and potential customer. We modify the decision tree to show those meaningful rules which has higher accuracy and large number of instances(more than 50 instance applied), as well as return label class as “yes”:

```
pdays = '(-inf-513]'
|   nr.employed = '(-inf-5087.65]'
|   |   duration = '(145.5-221.5]'
|   |   |   poutcome = success: yes (267.0/74.0)                       72.2%
|   |   duration = '(221.5-367.5]': yes (392.0/71.0)                   81.8%
|   |   duration = '(367.5-inf)': yes (328.0/51.0)                     84.4%
pdays = '(513-inf)'
|   duration = '(221.5-367.5]'
|   |   nr.employed = '(-inf-5087.65]'
|   |   |   contact = cellular
|   |   |   |   previous = '(-inf-0.5]': yes (529.0/234.0)             55.7%
|   |   nr.employed = '(5087.65-5183.65]'
|   |   |   month = mar
|   |   |   |   default = no: yes (50.0/12.0)                          76%
|   duration = '(367.5-inf)'
|   |   nr.employed = '(-inf-5087.65]': yes (820.0/307.0)              62.5%
|   |   nr.employed = '(5087.65-5183.65]'
|   |   |   |   previous = '(-inf-0.5]'
|   |   |   |   |   month = apr
|   |   |   |   |   |   age = '(-inf-30.5]': yes (79.0/32.0)           59.4%
```

According the above modified decision tree, it clearly shows that those attributes that are the most influential ones that decides whether a customer subscribe a term deposit. Here we list all of those meaningful attributes: {pdays, nr.employed, duration, poutcome, contact, previous, month, default, age}.

Moreover, we highlight those classification rules which has the accuracy higher than 70% , and applied more than 200 instances. Totally there is three most influential rules:

- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(145.5-221.5]' **AND** poutcome = success **THEN** y = yes; - ***(accuracy = 72.2%)***
- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(221.5-367.5]' **THEN** y = yes; - ***(accuracy = 81.8%)***
- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(367.5-inf)' **THEN** y = yes; - ***(accuracy = 84.4%)***

J48 algorithm generates such a decision tree with high accuracy on classification rules. As showing us a comprehensive view of the filtered data with values, the decision tree indicates those candidates for the most influential attributes that decide the promising and potential customers of subscribing the marketing product.

- ***Naive Bayesian Classification***

Now we consider apply the filtered attributes with values into the Naive Bayesian Classification. The Naive Bayesian classifiers “assumes that the effect of an attribute value on a given class is independent of the values of the other attributes” (350), which is called class-conditional independence. That means it is made to simplify the computations involved. Technically, Naive Bayesian classification could have the minimum error rate in comparison to others. In practice, it has the disadvantages of assuming independence of features, namely it cannot achieve that ideal case owing to inaccuracies in the assumptions made for its use. However, Naive Bayesian Classification’s advantages makes it helpful on finding the most meaningful and influential classification rules and attributes. Naive Bayesian Classification has following advantages:

- It is fast to train, as well as is fast to classify.
- It is not sensitive to irrelevant features.
- It can handle real and discrete data.
- It can handle streaming data well.

Under this context, we apply the previous filtered attributes with their data values into the Naive Bayesian Classification algorithm. We split the all data into two part, where one part containing 70% data is used for generating classification rules, while another part containing 30% data is used for testing. Therefore, we then get the result as follows:

Correctly Classified Instances	Incorrectly Classified Instances	Time taken to build model
85.0842% (10513)	14.9158% (1843)	0.03s

For detailed accuracy by class under the weighted average, the result is showed as follow:

TP Rate	FP Rate	Precision	Recall	F-Measure
0.851	0.309	0.888	0.851	0.865

Based on the result table, we can clearly see the rules generated by Naive Bayesian Classification also has a high accuracy on correctly classified instances, but is not good as J48 decision tree dose. In detailed, the whole process of running Naive Bayesian Classification to generate classification rules totally takes 0.03 seconds, which is more faster than J48 algorithm. And in the 30% testing data which has totally 12356 instance, there are 10513 instances which takes 85.0842% of all testing data can correctly classified, while there are 1843 instances which takes 14.9158% of all testing data cannot correctly classified.

Now let's consider the three classification rules derived from J48 algorithm. We want to utilize the Naive Bayesian Classification to verify the three rules.

For rule-1:

- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(145.5-221.5]' **AND** poutcome = success **THEN** y = yes;
- Let X =" pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(145.5-221.5]' **AND** poutcome = success", then according to the Naive Bayesian Classification result, we know:
- P(y = "no")*P(X| y = "no") = 0.89*(549/36550)*(2756/36553)*(7684/36553)*(480/36552) = 2.78 * 10^(-6)
- P(y = "yes")*P(X| y = "yes") = 0.11*(968/4645)*(2210/4645)*(601/4645)*(895/4644) = 0.00027196
- As P(y = "no")*P(X| y = "no") < P(y = "yes")*P(X| y = "yes"), thus according to Naive Bayesian Classification, rule-1 should be P(X| y = "yes"), which agrees with rule-1 derived from J48 algorithm, thus rule-1 is reliable in Naive Bayesian Classification

For rule-2:

- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(221.5-367.5]' **THEN** y = yes;
- Let Y = "pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(221.5-367.5]'", then acceding to the Naive Bayesian Classification, we know:
- P(y = "no")*P(Y| y = "no") = 0.89*(549/36550)*(2756/36553)*(7211/36553) = 0.000198
- P(y = "yes")*P(Y| y = "yes") = 0.11*(968/4645)*(2210/4645)*(1021/4645) = 0.002397

- As P(y = “no”)*P(Y| y = “no”) < P(y = “yes”)*P(Y| y = “yes”), thus according to Naive Bayesian Classification, rule-2 should be P(Y| y = “yes”), which agrees with rule-2 derived from J48 algorithm, thus rule-2 is reliable in Naive Bayesian Classification.

<u>For rule-3:</u>

- **IF** pdays = '(-inf-513]' **AND** nr.employed = ‘(-inf-5087.65]' **AND** duration = '(367.5-inf)' **THEN** y = yes;
- Let Z = “pdays = '(-inf-513]' **AND** nr.employed = ‘(-inf-5087.65]' **AND** duration = ‘(367.5-inf)’”, then acceding to the Naive Bayesian Classification, we know:
- P(y = “no”)*P(Z| y = “no”) = 0.89*(549/36550)*(2756/36553)*(5533/36553) = 0.0001525
- P(y = “yes”)*P(Z| y = “yes”) = 0.11*(968/4645)*(2210/4645)*(2719/4645) = 0.00638
- As P(y = “no”)*P(Z| y = “no”) < P(y = “yes”)*P(Z| y = “yes”), thus according to Naive Bayesian Classification, rule-3 should be P(Z| y = “yes”), which agrees with rule-3 derived from J48 algorithm, thus rule-3 is reliable in Naive Bayesian Classification.

Under this condition, those three “meaningful” rules derived from J48 algorithm are all reliable in Naive Bayesian Classification. However, as the result from J48 has higher accuracy than Naive Bayesian Classification, then we should consider improve the accuracy of Naive Bayesian Classification.

Let’s consider the twice-filtered attributes in J48, that is {pdays, nr.employed, duration, poutcome, contact, previous, month, default, age}. We try to use Naive Bayesian Classification to test whether those twice-filtered attributes make sense in other classifiers. We extract those attributes and apply them into Naive Bayesian Classification, then we can get the following result: (Also 70% data for building classification, 30% data for testing)

Correctly Classified Instances	Incorrectly Classified Instances	Time taken to build model
89.066% (11005)	10.934 % (1351)	0.03s

For detailed accuracy by class under the weighted average, the result is showed as follow:

TP Rate	FP Rate	Precision	Recall	F-Measure
0.891	0.496	0.882	0.891	0.885

Acceding to the new result, we find that the correctly classified instances percentage has raised from 85% to 89%, while the incorrectly classified instances percentage has fall down from 14% to 10%. That means the twice-filtered attributes from J48 algorithm definitely raise the accuracy of classification. Therefore it indicates that the twice-filtered attributes {pdays, nr.employed, duration, poutcome, contact, previous, month, default, age} based on J48 algorithm is the meaningful and influential attributes we should focus on.

- ***Neural Network Algorithm: MultilayerProceptron***

Finally we consider applying the filtered attributes with values into Neural Network Algorithm. This method utilizes gradient descent to model the data through “minimizing the mean-squared distance between the network’s class prediction and the actual class label of data tuples”(437), that is, it continues updating the weights to optimize the neural network of the whole data set. The Neural Network algorithm has multiple advantages as follows:

- Data driven and self-adaptive
- Universal function approximations
- Non-linear model making, flexible for real world applications

- High accuracy and noise tolerance

Under this context, we apply the previous filtered attributes with their data values into the MultilayerProceptron of Neural Network Classification algorithm. We split the all data into two part, where one part containing 70% data is used for generating classification rules, while another part containing 30% data is used for testing. For the first test, we set the number of hidden layer as 1, and then get the result as follows:

Correctly Classified Instances	Incorrectly Classified Instances	Time taken to build model
87.2936% (10786)	12.7064% (1570)	50.73s

For detailed accuracy by class under the weighted average, the result is showed as follow:

TP Rate	FP Rate	Precision	Recall	F-Measure
0.873	0.244	0.904	0.873	0.892

According to the result table, we can clearly see the rules generated by Neural Network Classification also has a higher accuracy on correctly classified instances than Naive Bayesian Classification, but is not good as J48 decision tree dose. In detailed, the whole process of running Naive Bayesian Classification to generate classification rules totally takes 50.73 seconds, which is too slow and occupies to much resource for running. And in the 30% testing data which has totally 12356 instance, there are 10786 instances which takes 87.2936% of all testing data can correctly classified, while there are 1570 instances which takes 12.7064% of all testing data cannot correctly classified. As the Neutral Network Classification takes much time and resources on running but get a not good enough result, we should consider improve its accuracy.

Let's firstly consider changer the layer number of Neural Network Classification. In this case, we change the number of hidden layer as 2, and then we get the result as follow:

Correctly Classified Instances	Incorrectly Classified Instances	Time taken to build model
90.5795 % (11192)	9.4205 % (1164)	91.21s

For detailed accuracy by class under the weighted average, the result is showed as follow:

TP Rate	FP Rate	Precision	Recall	F-Measure
0.906	0.564	0.893	0.906	0.892

Acceding to the new result, we find that the correctly classified instances percentage has raised from 87% to 90%, while the incorrectly classified instances percentage has fall down from 12% to 9%. That means adding one more hidden layer could definitely improve the accuracy of the Neural Network Algorithm, but it would require more time and resources, which is not efficient. Under this context, we should come up with another method to improve the accuracy without increasing the cost.

Let's consider using the twice-filtered attributes in J48, that is {pdays, nr.employed, duration, poutcome, contact, previous, month, default, age} to improve the accuracy of Neural Network Algorithm. In this case by removing other meaningless attributes and running the Neural Network Algorithm, we get the result as follows:

Correctly Classified Instances	Incorrectly Classified Instances	Time taken to build model
88.0949 % (10885)	11.9051 % (1471)	32.89s

For detailed accuracy by class under the weighted average, the result is showed as follow:

TP Rate	FP Rate	Precision	Recall	F-Measure
0.881	0.881	0.776	0.881	0.825

Acceding to the new result, we find that the correctly classified instances percentage has raised from 87% to 88%, while the incorrectly classified instances percentage has fall down from

12% to 11%. That means the twice-filtered attributes from J48 algorithm definitely raise the accuracy of classification. Therefore it indicates that the twice-filtered attributes {pdays, nr.employed, duration, poutcome, contact, previous, month, default, age} based on J48 algorithm is the promising attributes list which is worthy to discuss.

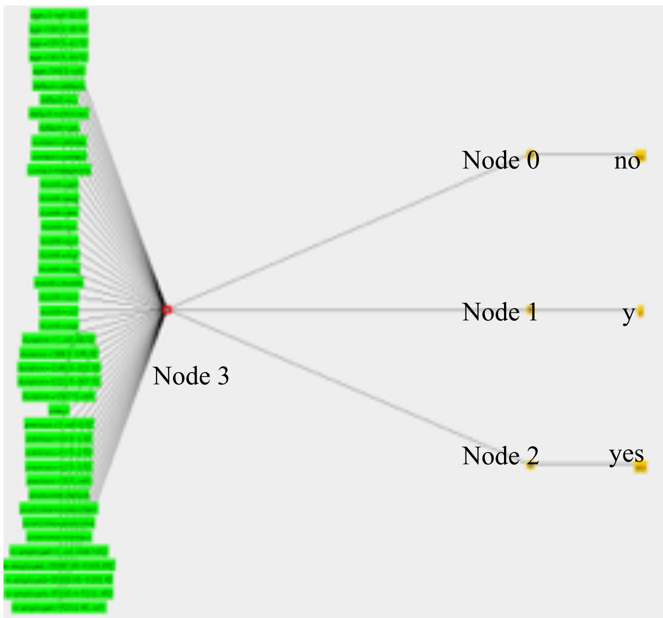

Fig-4

```
Sigmoid Node 0
    Inputs    Weights
    Threshold    0.1862202830333742
    Node 3    3.4397636132770395
Sigmoid Node 1
    Inputs    Weights
    Threshold    -7.535446502195323
    Node 3    -2.2224820840757227
Sigmoid Node 2
    Inputs    Weights
    Threshold    -0.18622028303337423
    Node 3    -3.4397636132770395
```

Fig-5

The above figures Fig-4 shows the graph of this Neural Network Classification, and the Fig-5 shows the detail information of the out nodes.

After improving the accuracy this algorithm, let’s consider the three classification rules derived from J48 algorithm. We want to utilize the Neural Network Classification to verify the three rules.

<u>For rule-1:</u>

- **IF** pdays = '(-inf-513]' **AND** nr.employed = ‘(-inf-5087.65]' **AND** duration = ‘(145.5-221.5]' **AND** poutcome = success **THEN** y = yes;

- Let X =" pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(145.5-221.5]' **AND** poutcome = success", then according to the Neural Network Classification result, we know: (let L(A) represents the Node 3's output of A in Neural Network Classification, O(A) represents the out node's output of A))
- L(X) = 12.755632338336204 + (-27.949616706549122) + (-5.045526174226076) + (-4.333819750254262) = -24.57;
- Notice that Node 0's condition c_0 = 3.44, Node 1's condition c_1 = -2.22, Node 2's condition c_2 = -3.43. Since -24.57 < -3.43 = c_2, thus the path would go to Node 2
- Notice that Node 0's threshold n_0 = 0.186, Node 1's threshold n_1 = -7.535, Node 2's threshold n_2 = -0.186, then the final output is O(X) = L(X) + n_2 = -24.767.
- According to the graph of neural network, Node 2 would go to "yes", then rule-1 should be y = "yes", which agree with rule-1 derived from J48 algorithm, thus rule-1 is reliable in Neural Network Classification.

For rule-2:

- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(221.5-367.5]' **THEN** y = yes;
- Let Y =" pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(221.5-367.5]' ", then according to the Neural Network Classification result, we know: (let L(A) represents the output of A in Neural Network Classification)
- L(X) = 12.755632338336204 + (-27.949616706549122) + (-8.434233356339677) = -23.62 ;

- Notice that Node 0's condition c_0 = 3.44, Node 1's condition c_1 = -2.22, Node 2's condition c_2 = -3.43. Since -23.62 < -3.43 = c_2, thus the path would go to Node 2
- Notice that Node 0's threshold n_0 = 0.186, Node 1's threshold n_1 = -7.535, Node 2's threshold n_2 = -0.186, then the final output is O(X) = L(X) + n_2 = -23.806.
- According to the graph of neural network, Node 2 would go to "yes", then rule-2 should be y = "yes", which agree with rule-2 derived from J48 algorithm, thus rule-2 is reliable in Neural Network Classification.

<u>For rule-3:</u>

- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(367.5-inf]' **THEN** y = yes;
- Let X ="pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(367.5-inf]' ", then according to the Neural Network Classification result, we know: (let L(A) represents the output of A in Neural Network Classification)
- L(X) = 12.755632338336204 + (-27.949616706549122) + (-34.726416006656265) = -49.92 ;
- Notice that Node 0's condition c_0 = 3.44, Node 1's condition c_1 = -2.22, Node 2's condition c_2 = -3.43. Since -49.92 < -3.43 = c_2, thus the path would go to Node 2
- Notice that Node 0's threshold n_0 = 0.186, Node 1's threshold n_1 = -7.535, Node 2's threshold n_2 = -0.186, then the final output is O(X) = L(X) + n_2 = —59.106.
- According to the graph of neural network, Node 2 would go to "yes", then rule-3 should be y = "yes", which agree with rule-3 derived from J48 algorithm, thus rule-3 is reliable in Neural Network Classification.

According above results, those three “meaningful” rules derived from J48 algorithm are also reliable inNeural Network Classification.

Result and Conclusion

After respectively applying the above three classification algorithms onto the data set of bank market campaign, we have found a specific meaningful attributes and three most influential classification rules. As applying same instances for classifying and same instances for testing during the classification process, the three algorithms result in different accuracies on their own classifications. The J48 decision tree has the highest accuracy, Neural Network Classification has the average one, and the Naive Bayesian Classification has the worst accuracy. Therefore, we utilized the J48 to generate the twice-filtered attributes and influential classification rules only rely on decision tree. Then we applied the twice-filtered attributes into Naive Bayesian Classification and Neural Network Classification, and then we found both of the tow classification algorithm result in higher accuracy. Based on that, we now can claim the twice-filtered attributes are reliable and worthy to discuss more. Moreover, we also applied the three influential classification rules based on J48 decision tree into Naive Bayesian Classification and Neural Network, in order to verify the correctness of these rules. The results of two verifying results shows that the two classification algorithm generate the same value of label class based on the same conditions of the three influential rules. As a result, the three influential rules definitely makes sense and are valuable.

Knowledge Discovered:

- Meaning filtered attributes list:{pdays, nr.employed, duration, poutcome, contact, previous, month, default, age}

- Most Influential Classification Rules :

The above filtered attributes indicates bank officers should pay more attention on those related attributes of {pdays, nr.employed, duration, poutcome, contact, previous, month, default, age}. That means the key point to identify whether a customer would subscribe a term deposit is to consider the above attributes.

Furthermore, the three influential rules also indicates that bank officer should get known that the potential rules that:

- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(145.5-221.5]' **AND** poutcome = success **THEN** y = yes; - ***(accuracy = 72.2%)***
- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(221.5-367.5]' **THEN** y = yes; - ***(accuracy = 81.8%)***
- **IF** pdays = '(-inf-513]' **AND** nr.employed = '(-inf-5087.65]' **AND** duration = '(367.5-inf)' **THEN** y = yes; - ***(accuracy = 84.4%)***

In the case where the intervals of contacting the specific customer is shorter, the duration of the phone call towards the specific customer is as long as possible, the number of employers who try to contact the specific customer is larger, and the customer used to purchase the market product, the specific customer is more possible to subscribe a term deposit. That means bank officer should put more resource on those customers who has been contacted few days ago with a long duration phone call communication by many employees and used to buy market product, since those costumer are the most promising ones who would like to subscribe the term deposit.

Extensions and Improvements

The meaningful and promising twice-filtered attributes derived from J48 decision tree helps on improving the accuracy of other classier, we then should consider whether this twice-filtered attributes could be optimized into less attributes with stronger relationship towards the label class. That means there may still exist some weak related attributes in the twice-filtered attributes list, which should be remove further.

Furthermore, the number of instances with value “yes” is much smaller than the number of instances with value “no”. Under this case, we actually just have around one thousand instances with value “yes” could be utilized for classification, which is still a small number of instance. Because of that, the classification results may not be general enough. That means the result could be influenced by number of applied instances, so that it could cause the biased conclusion. In order to get more general result, we then should consider collect more instance with the value “yes”.

Reference

Han, Jiawei, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques. 3rd ed. San Francisco: Morgan Kaufmann, 2011. Print.

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014