Bank Marketing

Miguel Villamayor, Miguel.villamayor@ryerson.ca
Tanya Sheryl, tsheryl@ryerson.ca
Zafrina Somani, zsomani@ryerson.ca
Kabriya Thavaratnam, kabriya.thavaratnam@ryerson.ca
Huzaifa Gul, huzaifa.gul@ryerson.ca

Abstract

The bank marketing dataset is used to devise an effective telemarketing strategy to sell long-term deposit accounts. This dataset was collected from a Portuguese bank that wanted to develop a telemarketing strategy using up to 17 different attributes provided. The dataset was used to implement a strategy to analyze which type of customers would subscribe to a long-term deposit account. It was determined that this was a supervised dataset and that in this dataset, three classification algorithms were used for analysis. The three algorithms were; the Decision tree, Naïve Bayes and Random Forest. After applying the algorithms on the dataset, accuracy was compared between the three algorithms. Using the performance measures such as accuracy, recall, precision, true positive and false positive rates it was determined that the Random Forest algorithm acquired the highest accuracy thus suggesting it to be the best performing predictive algorithm. In the post analysis, the “no” class was filtered out creating an unsupervised dataset. The Apriori algorithm and K-means clustering was utilized on the unsupervised dataset for pattern mining. Seven top rules were determined to have a confidence level of 100%, while K-means clustering generated a list of the best characteristics that represented people saying yes.

Workload Distribution

Miguel Villamayor	Experiment/methods, Discussion
Tanya Sheryl	Introduction, Discussion
Zafrina Somani	Results, Appendix, Discussion
Kabriya Thavaratnam	Abstract, Discussion
Huzaifa Gul	Introduction, Discussion, Conclusion

Introduction

The dataset was collected from a Portuguese bank that wants to have an effective telemarketing strategy to sell long-term deposit accounts (e.g., bonds, saving accounts, etc.). The marketing campaigns were based on phone calls and multiple contacts were often needed to determine whether a customer would subscribe to a long-term deposit account. Your team of data scientists will help this bank in determining such customers and devising an effective telemarketing strategy by applying data analytics methods on the given dataset.

The goal of classifying this data is to predict if the client will subscribe to a long term deposit or not and to find strategies to improve the next marketing campaign.

There are 17 attributes provided, where the class attribute indicates if the client has actually subscribed to the account or not. There are 6 numeric attributes and the rest are qualitative/categorical. The following are the attributes classified under customer, finance, campaign or other.

Customer: Age, Job, Martial, Education

Finance: Default, Balance, Housing , Loan

Campaign: Contact, Day, Month, Duration, Campaign, Pdays, previous, Poutcome

Other: Class

Since the outcome of the dataset is already known, supervised learning algorithms can be implemented such as Decision Trees, Naïve Bayes and Random Forest Algorithms.

Decision Tree

The decision tree algorithm is used to classify data by comparing the attributes of the data set to predict a target variable. It classifies the data set by creating an inverted tree starting at the root node and partitioning the data along the child nodes [5].

There are basically two types of Decision Trees - Classification tree where the outcome of the class will be a discrete value and a Regression Tree where the outcome of the class will be a continuous value.

Naive Bayes

Naive Bayes is a classification method based on Bayes' Theorem and the assumption that a particular feature in a class is independent of other features.

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

A is the response variable and B is the input attribute.

P(A|B) : conditional probability of response variable belonging to a particular value, given the input attributes. This is also known as the posterior probability.

P(A) : The prior probability of the response variable.

P(B) : The probability of training data or the evidence.

Random Forest

Random forest classifier creates a group of decision trees by randomly selecting data from the training set. It then aggregates the votes from different decision trees to decide the final class of the test object. A Random forest works better than a single decision tree as an aggregate of many such decisions would reduce the effect of noise and provides more accurate results [7].

The different parameters that could be used as input to the tree could be the total number of trees to be generated, minimum split and the split criteria.

10-Fold-Cross-Validation:

Cross-validation is a statistical technique for machine learning model evaluation.[1] A value of k=10 is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

Percentage split:

Percentage split divides the data set into two parts, training and testing, perhaps two-thirds of it for training and one-third of it for testing. Both training and test sets are produced by independent sampling from an infinite population. The training set is used to build the model and the test set is used to evaluate.

Apriori:

Apriori is a simple algorithm designed to be applied on a transaction dataset to discover patterns. It is a significant algorithm for mining frequent itemsets and association between various item sets.

K-Mean Clustering

K-Mean is a clustering algorithm that seeks hidden patterns that potentially exist in the data set. Its purpose is to group data into distinct clusters whereby each cluster is similar as it can, yet different. The "K" represents the number of clusters applied to a data set, with each cluster being represented by a centroid. The K-Means algorithm performs its task by calculating the squared distances between the inputs and the centroids, and decides which cluster the input belongs based on the shortest squared distance [3]. To provide the most optimal results for K-Means clustering, the optimal cluster values need to be known. For this experiment, the elbow method was utilized to determine the optimal cluster value.

Experiment:

Data was imported into both R and Weka for analysis. R's "summary ()" function was utilized to determine the max, min and mean for all the numeric attributes. It was subsequently converted into a data frame to utilize the "sapply(x, sd)" function to determine their standard deviations. Outlier determination was done through R's "boxplot ()" function. Histograms were generated using the "hist()" function to determine all of the numeric attribute's distribution visually.

Within the Weka software; under the "Select Attribute" tab; attribute correlation towards the class attribute were ranked through the use of "CorrelationAttibuteEval" with "Ranker" as the search method. The data was determined to be imbalance and was rebalanced by utilizing "Smote" filter in the preprocess tab.

Predictive analysis was done through the use of Weka's Classifier. Under the Classify tab 3 classifiers were chosen, and for each classifier two kinds of data split were utilized. The 3 classifiers chosen were the J48 (Decision Tree), Naïve Bayes, and Random Forest. The two data split chosen were the 10-fold cross validation, and the percentage split utilizing 80% as its value. Screenshots of the performance measure were obtained of the three-different classifier and their respective data split.

For the post predictive analysis, class rebalancing with Smote was undone, and all the instances with the "no" outcome within the class attribute were manually removed with the use of Weka's edit. First, the optimal number of clusters were obtained for K-Means clustering; by graphing the # of clusters vs "Within Cluster sum of squared errors" values, utilizing excel's scatter plot. The range of the clusters started from 0 and ended with 10 clusters, while the best # of clusters was determined visually through the elbow method. Utilizing the optimal cluster number, along with the percentage split of 80%; a list of characteristics was obtained based on which cluster held the highest amount of data. For the Apriori algorithm, all numeric values were converted to nominal values utilizing the "NumerictoNominal" filter. After the conversion, association learning was applied through the use of Apriori algorithm, having the confidence set to 0.80, and the minimum support to 0.70. The "numRules" were set to 50 and the top "37" rules were formulated.

Results

For the numeric data, the following shows 6-number summary from R:

Age	Balance	Day	Duration	Campaign	PDays	Previous
Min: 19	Min: -3313	Min: 1	Min: 4	Min: 1	Min: -1	Min: 0
1st Qu: 33	1st Qu: 69	1st Qu: 9	1st Qu: 104	1st Qu: 1	1st Qu: -1	1st Qu: 0
Median: 39	Median: 444	Median: 16	Median: 185	Median: 2	Median: -1	Median: 0
Mean: 41.17	Mean: 1423	Mean: 15.92	Mean : 264	Mean : 2.794	Mean : 39.77	Mean : 0.5426
3rd Qu: 49	3rd Qu: 1480	3rd Qu: 21	3rd Qu: 329	3rd Qu: 3	3rd Qu: -1	3rd Qu: 0
Max: 87	Max: 71188	Max: 31	Max: 3025	Max: 50	Max: 871	Max: 25
St.Dev: 10.576	St.Dev: 3009.638	St.Dev: 8.248	St.Dev: 259.857	St.Dev: 3.11	St.Dev: 100.121	St.Dev: 1.694

The J48 decision tree was run in Weka using 10-fold cross validation. The results show an accuracy of 87.9%. The true positive and false positive rate for the "no" class was 0.93 and 0.32, respectively. The precision for the "no" class was 0.92 and recall was 0.93. Furthermore, the true positive rate and false positive rate for the "yes" class was 0.68 and 0.07, respectively. The precision for the "yes" class was 0.72 and the recall was 0.68.

Alternatively, the J48 decision tree was run in Weka using percentage split with an 80% training set and 20% test set. The accuracy was 86.6%. The true positive and false positive rate for the “no” class was 0.94 and 0.41, respectively. The precision for the “no” class was 0.90 and recall was 0.94. The true positive and false positive rate for the “yes” class was 0.60 and 0.06, respectively. The precision for the “yes” class was 0.71 and recall was 0.60.

The Naïve Bayes classifier was run on Weka using 10-fold cross validation. The accuracy of the model was 83%. The true positive and false positive rate of the “no” class was 0.89 and 0.37, respectively. The precision for the “no” class was 0.90 and recall was 0.89. Alternatively, the true positive rate and false positive rate for the “yes” class was 0.63 and 0.11, respectively. The precision for the “yes” class was 0.59 and recall was 0.63.

The Naïve Bayes classifier was also run on Weka using percentage split with an 80% training set and a 20% test split. The accuracy was 84%. The true positive and false positive rate for the “no” class was 0.90 and 0.39, respectively. The precision for the “no” class was 0.90 and recall was 0.90. The true positive and false positive rate for the “yes” class was 0.61 and 0.11, respectively. The precision for the “yes” class was 0.60 and recall was 0.61.

The Random Forest classifier was run on Weka using 10-fold cross validation. The accuracy of the model was 90%. The true positive and false positive rate of the “no” class was 0.96 and 0.33, respectively. The precision for the “no” class was 0.91 and recall was 0.96. Alternatively, the true positive rate and false positive rate for the “yes” class was 0.67 and 0.03, respectively. The precision for the “yes” class was 0.82 and recall was 0.67.

The Random Forest classifier was also run on Weka using percentage split with an 80% training set and a 20% test split. The accuracy was 89%. The true positive and false positive rate for the “no” class was 0.97 and 0.40, respectively. The precision for the “no” class was 0.90 and recall was 0.97. The true positive and false positive rate for the “yes” class was 0.60 and 0.03, respectively. The precision for the “yes” class was 0.82 and recall was 0.60.

The Apriori Algorithm was run in Weka. The minimum support was set to 70% and the minimum confidence level was set to 80%. The best rules that lead to our class attribute were shown to be:

1) Default = no → class = yes [confidence = 1]
2) Loan = no → class = yes [confidence = 1]
3) Default = no, loan = no → class = yes [confidence = 1]
4) Contact = cellular → class = yes [confidence = 1]
5) Default = no, contact = cellular → class = yes [confidence = 1]

6) Loan = no, contact = cellular → class = yes [confidence = 1]

7) Default = no, loan = no, contact = cellular → class = yes [confidence = 1]

Graph 1, shows how the optimal clusters were obtained for K-Mean clustering. The optimal number was determined to be 5 clusters based on the elbow method.

Discussion

Data Preparation

Based on the summary and histogram of all attributes, we have determined that data is not normally distributed. Using a box plot model, there were some outliers identified in the dataset, for example; age attribute outliers were found to be between 78 to 83. Then an analysis on the distribution of numeric attributes was conducted by plotting histograms for attributes of concern to analyze the influence on the various class attributes within the dataset.

Correlation was found between the class attribute and duration using Weka's Correlation Attribute Evaluator. The next ranked attribute using this correlation method was contact, however, the pearson correlation was shown to be 0.22, which is a weak correlation. A correlation above r=0.50 was seen between the outcome of the previous marketing campaign, the number of days the client was last contacted from previous campaign, and the number of contacts performed before the campaign for the client.

We loaded the dataset into Weka for more detailed analysis. There are 4521 instances in total, including 4000 clients who haven't subscribed and only 521 clients have subscribed. This poses a problem for us because the dataset is imbalanced. When we take a close look at the accuracy measurement of our initial classifier, because there are significantly more "no '' than "yes" in the class attribution, the model automatically predicts "no". Our true intention is to try to focus on predicting "yes". Therefore, by only looking at the accuracy of the model is not enough to evaluate how reliable the model is. We have concluded that in order to make the classifier predicting the outcome of "yes", we must apply some statistical technique.

There are a number of ways to address the class imbalance. SMOTE was used ("Synthetic Minority Over-sampling Technique") , which is an algorithm that rebalances the data by oversampling the minority class [2]. SMOTE uses the nearest neighbor information to synthetically generate new (but representative) data for the minority class.

It was decided that class balancer was inappropriate as it rebalances the weight equally between the “yes” and “no” instances. It was determined that this will not be a good representation of the actual situation; and that smote was more favorable as it still reflected the actual situation while also avoiding overfitting.

Identify the research questions: By building a predictive modeling (Decision Tree or Naïve Bayes), we can help the bank to find an effective telemarketing strategy to promote its long-term deposit accounts to its customer.

By studying the post-predictive analysis (Clustering analysis), we can identify the characteristics of existing customers to better understand the potential subscriber of this long-term deposit account.

All the attributes from the dataset were included in the final analysis as their effect is limited or insignificant.

Predictive

The highest information gain for the decision tree comes from the duration of the call, which is the top node of the decision tree. If the call was less than or equal to 220 seconds, the decision tree then looks at the outcome of the previous marketing campaign. When the marketing campaign was a success and the duration of the call was greater than 146 seconds, the client subscribed to the long-term deposit. However, when the duration of the call was shorter than 146 seconds, the tree then looks at whether the customer has a housing loan. If the client has a housing loan, then the customer is likely to say no to the term deposit. If they don’t have a housing loan and they are under the age of 34, they will likely say yes to the term deposit. Alternatively, if they are over the age of 34 and don’t have a housing loan, then they are likely to say no to the term deposit.

The duration of the call being the top node speaks to the importance of the ability of the telemarketer to get the client’s attention. In marketing, it is in the initial pitch and the first few minutes of the call that the customer will generally make a decision. This is reflected in our decision tree as duration of the call gives us the most information. Interestingly, the month that the customer was contacted last year and the contact method had an impact when the call duration was greater than 645 seconds. When the call was less than or equal to 645 seconds, the tree then looks at the number of days that passed since the client was last contacted from a previous campaign.

The customer’s bank balance, education, loans, housing, marital status and job type were found lower on the decision tree than expected. Instead, the data from the previous attempts, time of year, days passed since previous contact and whether the client was previously contacted were higher indicators of marketing success. Therefore, the most effective strategy to telemarket the long-term saving account would be for the telemarketed to be able to pitch within 220 seconds of the call and review the outcome of the previous campaign. If the previous campaign failed, the time since the customer was last contacted should be taken into consideration.

Comparison of Classification Models

The values of precision were compared for all three classification models. From the results it was determined that precision values were similar for everything under the no value for the class attribute. Cross validation was observed to perform better with 92% precision. In the yes values under the class attribute Random Forest was the optimal algorithm that performed well with a precision rate at 82%.

When evaluating recall for the class attribute's no values, Random Forest showed a 97% rate using percentage split and this was the ideal algorithm amongst the others. The yes values under class attribute had an optimal recall rate for the J48 algorithm. Since there wasn't much differences Random Forest was chosen using cross validation for both the yes and no values of the class attribute. The optimal accuracy rates were found when using the Random forest algorithm with cross validation at an accuracy of 90%. Precision values were also evaluated where J48 showed a 92% precision for the no values using cross validation and 90% for the yes values using percentage split.

Post-predictive

Both Apriori and K-mean algorithms were used to identify the characteristics of the clients that subscribed to the term deposit, which was represented by all the "yes" in the class attribute. Based on the scatter plot results, it was determined that five clusters were the optimal number. Utilizing the percentage split with the value of 80%, it yielded a result that showed that cluster #3 was the best representation having 35% of the total test set being grouped into this cluster. Moving on to Apriori; the values chosen to be the confidence and support were both 80% and 70% respectively. 80% was chosen as the minimum confidence to filter out all the sequences that had moderate to low probability of attaining the "yes" instance. The support was determined to be 70% to also filter out infrequent itemsets that lead to the "yes". The values filtered out were interpreted as having minimal significance to the class attribute. In addition, utilizing a higher support value yielded a set of rules that only took into consideration two attributes (default and loan); having the 70% minimum support also allowed other attributes to be represented in our set of rules.

K-Means Clustering determined that the best cluster possessed these characteristics shown below:

age	job	marital	education	default	balance	housing	loan	contact
39.6	blue-collar	married	secondary	no	1120.19	yes	no	cellular

day	month	Duration (s)	campaign	pdays	previous	poutcome	y
15.8167	may	682.5	2.35	47.5	0.71	unknown	yes

Cluster #3 contained 35% of the test data; while the other 4 clusters only possessed 13% to 19% of the total test data. Interestingly cluster #3 distinguishes itself from the rest by having “blue collar” as its job type, and “yes” to housing loans. This suggests that a person is likely to say yes to the long term deposit if this individual is a blue collar worker holding a house loan. Another interesting finding, comes from the cluster with the second highest amount of test data. Cluster #0 main distinguishing feature is that individuals within this cluster are most likely single, which may have an impact in their decision to purchase the long term deposit.

The Apriori algorithm determined the best rules to be that when the customer does not have a history of defaulting they are more likely to subscribe to a long-term deposit account. The second rule was if the customer did not have a loan they were also more likely to subscribe and if both conditions were also met they were also more likely to subscribe. All three of these rules had a confidence level of 1 which is higher than the minimum support of 0.7. This makes sense because customers who haven’t defaulted are more eligible to obtain a long-term deposit account whereas customers who have defaulted are probably not eligible and customers who don’t have a long are more likely to subscribe to this type of account.
Customer’s who’s contact method is primarily cellular was also determined to be one of the top rules for subscribing to an account with confidence level at 1, which essentially means that they probably have good credit to qualify for a long-term deposit account. Another top rule is if they haven’t defaulted and had no loan and main contact was cellular the confidence level was 1. The top seven best rules all retained a confidence level of 1 which is above the minimum support and these are all good rules to use to predict whether a customer is likely to subscribe to a long term account or not.

Sources of Error

The only information provided for the origin of this dataset is that it is from a bank in Portugal. It doesn’t inform us whether it is from a particular city/village. The data may not be representative of the population depending on what this bank wants to look at.

For example, they may want information regarding loan approval for people in the whole country or only in cities but we do not know where the data comes from. This means it is unknown whether the data is representative of the population and if the sample size is sufficient (n number).

There isn’t one perfect algorithm, there will be advantages and disadvantages to each one which may end up having a source of error. If the decision tree was solely used it would be susceptible to errors such as over fitting with large datasets. A disadvantage of Naive Bayes is that there is an assumption that each attribute is independent of the other; however, this is not the case as, for example, job will be dependent on age and education [4]. The disadvantage of the Random Forest decision tree is the high computation power required. To reduce errors such as bias and variance random forest was identified as an ideal algorithm.

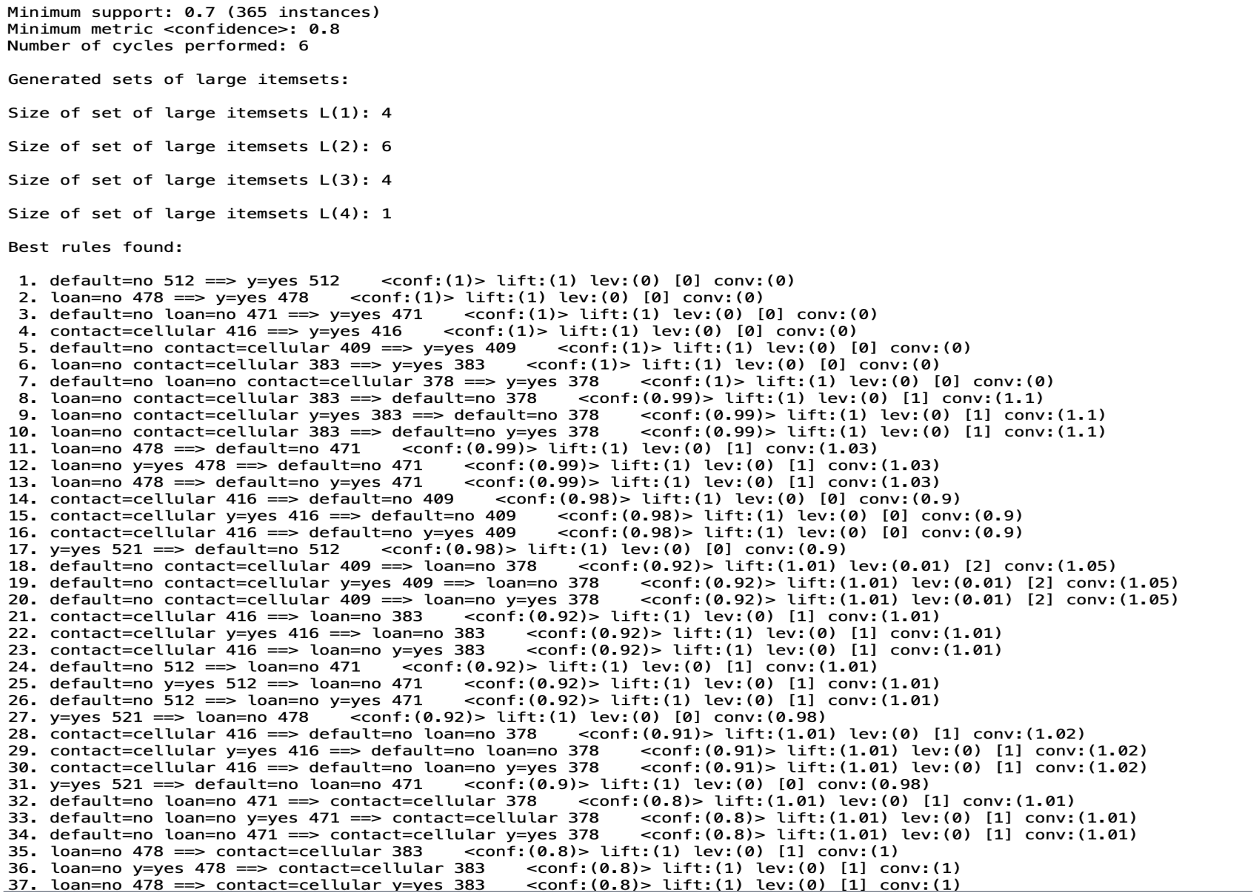
Conclusion

Clients that are more likely to subscribe for a long term deposit account were identified through a series of predictive modeling and post-prediction analysis. Of all the classifications used the Random Forest model using 10-fold cross validation was the most accurate. Random Forest yields the ideal TP and FP rate. Through the K means clustering model, patterns were identified between candidates that are more likely to subscribe. It was concluded that candidates who are blue collared, married, with secondary education, no defaults, bank balance of greater than 1000, and with no loan are most likely to subscribe. Furthermore, the Apriori algorithm found best rules relating to no default, no loan, and customers’ being contacted on their cellphones. In conclusion, by focusing on candidates with the listed attributes and who have been contacted in prior campaigns, the marketing strategy will result in more long-term deposit accounts being signed up for.

Reference

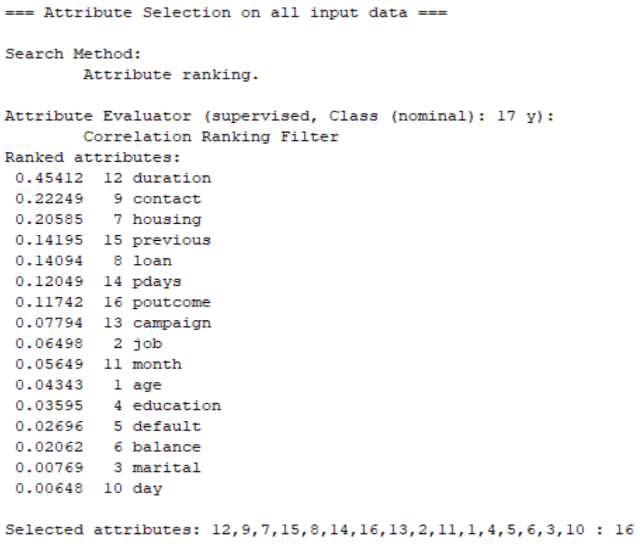
1. Biswas, S. (2019, May 29). Importance of K-Fold Cross Validation in Machine Learning. Retrieved from https://medium.com/towards-artificial-intelligence/importance-of-k-fold-cross-validation-in-machine-learning-a0d76f49493e
2. Collier, A. B. (2018, April 21). Classification: Get the Balance Right. Retrieved from https://datawookie.netlify.com/blog/2018/04/classification-get-the-balance-right/
3. Xing, K., Hu, C., Yu, J., Cheng, X., & Zhang, F. (2017). Mutual privacy preserving k -means clustering in social participatory sensing. IEEE Transactions on Industrial Informatics, 13(4), 2066-2076. doi:10.1109/TII.2017.2695487
4. Zhang, H. (2004). The optimality of naive Bayes. *AA*, *1*(2), 3.
5. Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044
6. D’Angelo, G., D’Angelo, G., Rampone, S., Rampone, S., Palmieri, F., & Palmieri, F. (2017). Developing a trust model for pervasive computing based on apriori association rules learning and bayesian classification. *Soft Computing, 21*(21), 6297-6315. doi:10.1007/s00500-016-2183-1
7. Ishwaran H. (2015). The Effect of Splitting on Random Forests. *Machine learning*, *99*(1), 75–118. https://doi.org/10.1007/s10994-014-5451-2

Appendix:



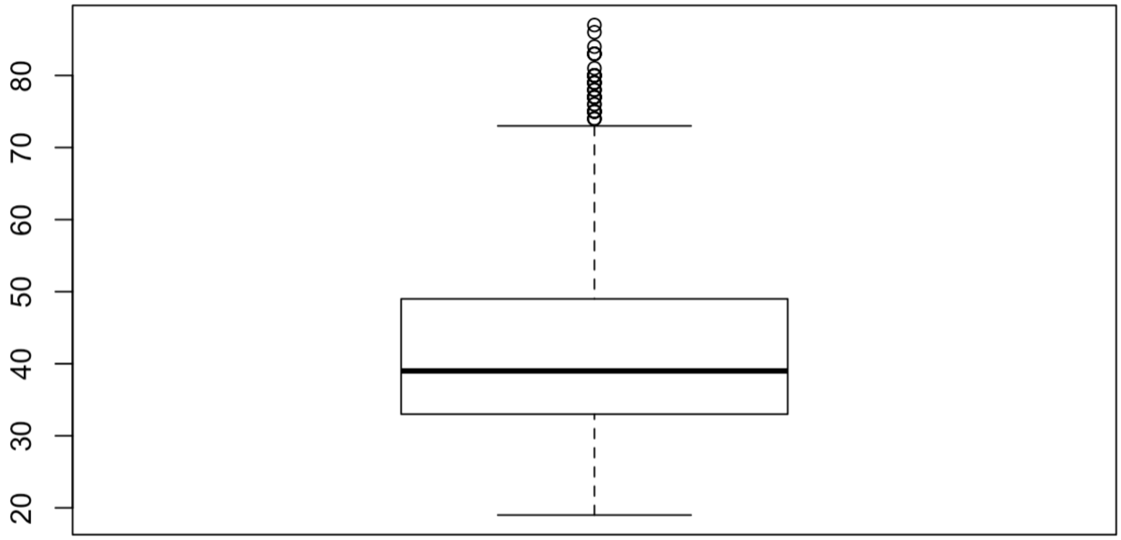
```
Minimum support: 0.7 (365 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 6

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 6

Size of set of large itemsets L(3): 4

Size of set of large itemsets L(4): 1

Best rules found:

 1. default=no 512 ==> y=yes 512    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. loan=no 478 ==> y=yes 478    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. default=no loan=no 471 ==> y=yes 471    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. contact=cellular 416 ==> y=yes 416    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. default=no contact=cellular 409 ==> y=yes 409    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. loan=no contact=cellular 383 ==> y=yes 383    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. default=no loan=no contact=cellular 378 ==> y=yes 378    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. loan=no contact=cellular 383 ==> default=no 378    <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.1)
 9. loan=no contact=cellular y=yes 383 ==> default=no 378    <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.1)
10. loan=no contact=cellular 383 ==> default=no y=yes 378    <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.1)
11. loan=no 478 ==> default=no 471    <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.03)
12. loan=no y=yes 478 ==> default=no 471    <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.03)
13. loan=no 478 ==> default=no y=yes 471    <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.03)
14. contact=cellular 416 ==> default=no 409    <conf:(0.98)> lift:(1) lev:(0) [0] conv:(0.9)
15. contact=cellular y=yes 416 ==> default=no 409    <conf:(0.98)> lift:(1) lev:(0) [0] conv:(0.9)
16. contact=cellular 416 ==> default=no y=yes 409    <conf:(0.98)> lift:(1) lev:(0) [0] conv:(0.9)
17. y=yes 521 ==> default=no 512    <conf:(0.98)> lift:(1) lev:(0) [0] conv:(0.9)
18. default=no contact=cellular 409 ==> loan=no 378    <conf:(0.92)> lift:(1.01) lev:(0.01) [2] conv:(1.05)
19. default=no contact=cellular y=yes 409 ==> loan=no 378    <conf:(0.92)> lift:(1.01) lev:(0.01) [2] conv:(1.05)
20. default=no contact=cellular 409 ==> loan=no y=yes 378    <conf:(0.92)> lift:(1.01) lev:(0.01) [2] conv:(1.05)
21. contact=cellular 416 ==> loan=no 383    <conf:(0.92)> lift:(1) lev:(0) [1] conv:(1.01)
22. contact=cellular y=yes 416 ==> loan=no 383    <conf:(0.92)> lift:(1) lev:(0) [1] conv:(1.01)
23. contact=cellular 416 ==> loan=no y=yes 383    <conf:(0.92)> lift:(1) lev:(0) [1] conv:(1.01)
24. default=no 512 ==> loan=no 471    <conf:(0.92)> lift:(1) lev:(0) [1] conv:(1.01)
25. default=no y=yes 512 ==> loan=no 471    <conf:(0.92)> lift:(1) lev:(0) [1] conv:(1.01)
26. default=no 512 ==> loan=no y=yes 471    <conf:(0.92)> lift:(1) lev:(0) [1] conv:(1.01)
27. y=yes 521 ==> loan=no 478    <conf:(0.92)> lift:(1) lev:(0) [0] conv:(0.98)
28. contact=cellular 416 ==> default=no loan=no 378    <conf:(0.91)> lift:(1.01) lev:(0) [1] conv:(1.02)
29. contact=cellular y=yes 416 ==> default=no loan=no 378    <conf:(0.91)> lift:(1.01) lev:(0) [1] conv:(1.02)
30. contact=cellular 416 ==> default=no loan=no y=yes 378    <conf:(0.91)> lift:(1.01) lev:(0) [1] conv:(1.02)
31. y=yes 521 ==> default=no loan=no 471    <conf:(0.9)> lift:(1) lev:(0) [0] conv:(0.98)
32. default=no loan=no 471 ==> contact=cellular 378    <conf:(0.8)> lift:(1.01) lev:(0) [1] conv:(1.01)
33. default=no loan=no y=yes 471 ==> contact=cellular 378    <conf:(0.8)> lift:(1.01) lev:(0) [1] conv:(1.01)
34. default=no loan=no 471 ==> contact=cellular y=yes 378    <conf:(0.8)> lift:(1.01) lev:(0) [1] conv:(1.01)
35. loan=no 478 ==> contact=cellular 383    <conf:(0.8)> lift:(1) lev:(0) [1] conv:(1)
36. loan=no y=yes 478 ==> contact=cellular 383    <conf:(0.8)> lift:(1) lev:(0) [1] conv:(1)
37. loan=no 478 ==> contact=cellular y=yes 383    <conf:(0.8)> lift:(1) lev:(0) [1] conv:(1)
```

Apriori Algorithm: Best rules from Weka



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 17 y):
        Correlation Ranking Filter
Ranked attributes:
 0.45412  12 duration
 0.22249   9 contact
 0.20585   7 housing
 0.14195  15 previous
 0.14094   8 loan
 0.12049  14 pdays
 0.11742  16 poutcome
 0.07794  13 campaign
 0.06498   2 job
 0.05649  11 month
 0.04343   1 age
 0.03595   4 education
 0.02696   5 default
 0.02062   6 balance
 0.00769   3 marital
 0.00648  10 day

Selected attributes: 12,9,7,15,8,14,16,13,2,11,1,4,5,6,3,10 : 16
```

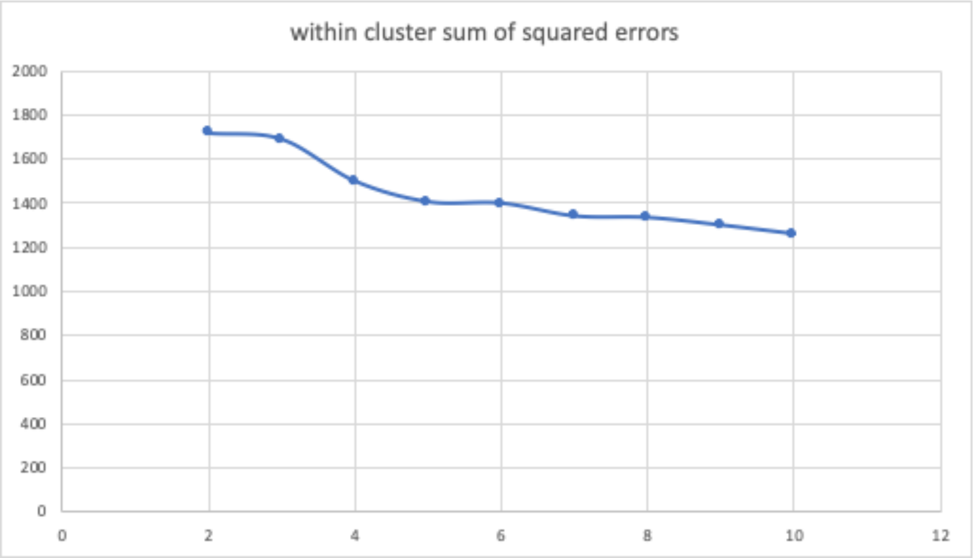
Correlation between attributes run on Weka



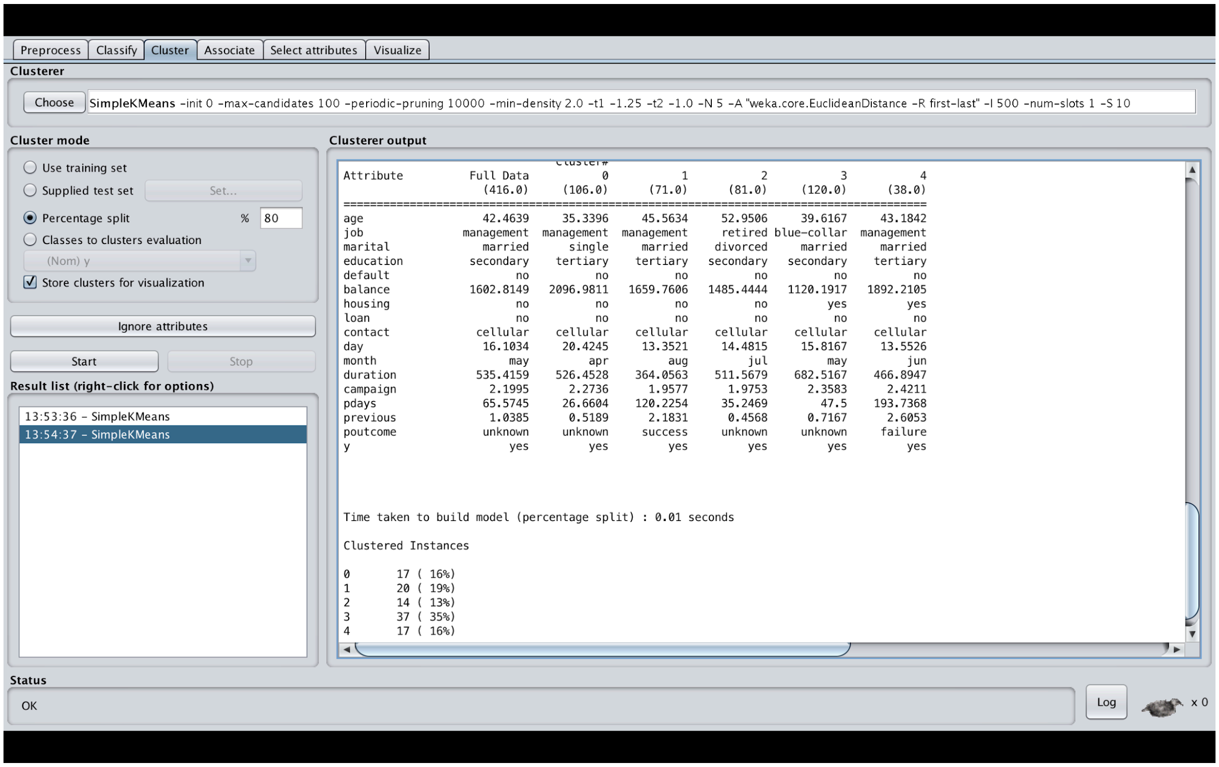
Box plot for age, box plots were used to determine number of outliers for each numeric attribute in R

Classification Method	Data Split	TP Rate	FP Rate	Precision	Recall	Accuracy
J48 Decision Tree	Cross Validation*	No – 0.93	No – 0.32	No – 0.92	No – 0.93	88%
		Yes – 0.67	Yes – 0.07	Yes – 0.72	Yes – 0.68	
	Percentage Split**	No – 0.94	No – 0.41	No – 0.90	No – 0.94	87%
		Yes – 0.60	Yes – 0.06	Yes – 0.71	Yes – 0.60	
Naïve Bayes	Cross Validation*	No – 0.89	No – 0.37	No – 0.90	No – 0.89	83%
		Yes – 0.63	Yes – 0.11	Yes – 0.59	Yes – 0.63	
	Percentage Split**	No – 0.90	No - 0.39	No – 0.90	No – 0.90	84%
		Yes – 0.61	Yes – 0.11	Yes – 0.60	Yes – 0.61	
Random Forest	Cross Validation*	No – 0.96	No – 0.33	No – 0.92	No – 0.96	90%
		Yes – 0.67	Yes – 0.04	Yes – 0.82	Yes – 0.67	
	Percentage Split**	No – 0.97	No – 0.40	No – 0.90	No – 0.97	89%
		Yes – 0.60	Yes – 0.04	Yes – 0.82	Yes - 0.60	

*Cross validation was done using 10-fold **Percentage Split was done using 80% training set and 20% test set



Within Cluster Sum of Squared Errors used to find ideal number of clusters for K-means Clustering



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

○ Use training set
○ Supplied test set
● Percentage split % 80
○ Classes to clusters evaluation
(Nom) y
☑ Store clusters for visualization

13:53:36 - SimpleKMeans
13:54:37 - SimpleKMeans

```
                          Cluster#
Attribute         Full Data          0          1          2          3          4
                    (416.0)    (106.0)     (71.0)     (81.0)    (120.0)     (38.0)
===================================================================================
age                 42.4639    35.3396    45.5634    52.9506    39.6167    43.1842
job              management management management    retired blue-collar management
marital             married     single    married   divorced    married    married
education         secondary   tertiary   tertiary  secondary  secondary   tertiary
default                  no         no         no         no         no         no
balance           1602.8149  2096.9811  1659.7606  1485.4444  1120.1917  1892.2105
housing                  no         no         no         no        yes        yes
loan                     no         no         no         no         no         no
contact            cellular   cellular   cellular   cellular   cellular   cellular
day                 16.1034    20.4245    13.3521    14.4815    15.8167    13.5526
month                   may        apr        aug        jul        may        jun
duration           535.4159   526.4528   364.0563   511.5679   682.5167   466.8947
campaign             2.1995     2.2736     1.9577     1.9753     2.3583     2.4211
pdays               65.5745    26.6604   120.2254    35.2469       47.5   193.7368
previous             1.0385     0.5189     2.1831     0.4568     0.7167     2.6053
poutcome            unknown    unknown    success    unknown    unknown    failure
y                       yes        yes        yes        yes        yes        yes




Time taken to build model (percentage split) : 0.01 seconds

Clustered Instances

0       17 ( 16%)
1       20 ( 19%)
2       14 ( 13%)
3       37 ( 35%)
4       17 ( 16%)
```

OK

K-means clustering output from Weka