Mobile Phone Sms-Spam Filter

Gulsum Oz

Table of Contents

**The Problem2**

**Motivation 2**

**Method2**

Naïve Bayes2

**Analysis 3**

Word Cloud 3

Data Partition 4

Bayesian Model 1 and 24

Model 1 accuracy 5

Model 2 accuracy 5

KNN Model5

Decision tree Model6

**Model comparison** 7

**Conclusion** 8

The Problem

The problem is building a spam filter for SMS messages by using machine learning algorithms.

The Motivation

Our motivation in this project was our interest on text mining area wish to combine text mining with machine learning algorithms. Email spam filtering was a good example and we wanted to see the same classification on the phone sms text too. We used many machine learning algorithm to see which algorithm perform better.

The Methods

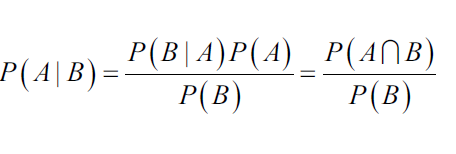
For this project first we used text mining method to clean the data and transformed words and sentences into a form that a computer can understand. We transformed data into a representation known as bag-of-words, which ignores the order that words appear in and simply provides a variable indicating whether the word appears at all. Then, we used Naive Bayes which basically takes advantage of pattern in the word frequency to determine whether the SMS message seem to be better fit to the profile of spam and ham. We tried to do same classification with different machine learning algorithm. We used "klaR", "MASS", "caret", "tm", "pander", "dplyr", packages in R to accomplish the project.

Naïve Bayes

Bayesian probability theory is rooted in the idea that the estimated likelihood of an event should be based on the evidence at hand. Events are possible outcomes. Probability of an event can be estimated from the observed data by dividing the number of trials in which an event occurred by the total number of trials.

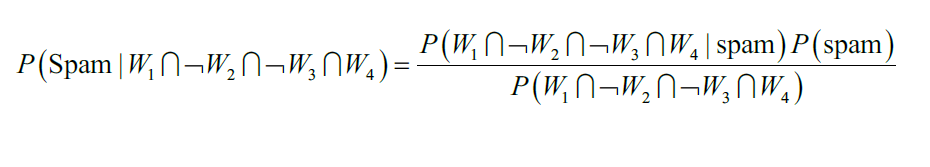
Bayesian probability can be calculated by combining prior probability and conditional probability. Prior probability is the probability assigned to a parameter or to an event in advance of any empirical evidence.

Relationship between dependent evets can be estimated by conditional probability which probability of event A is given that event B occurred.



To find the posterior probability Bayesian algorithm use prior probability and the class conditional probability. While doing this Naïve Bayes makes an assumption which is all of the features in the dataset is equally important and independent.

Posterior probability of spam detection project is calculated by using the formula given below. For given words to be spam is depend on prior probability of the words being spam and class conditional probabilities.



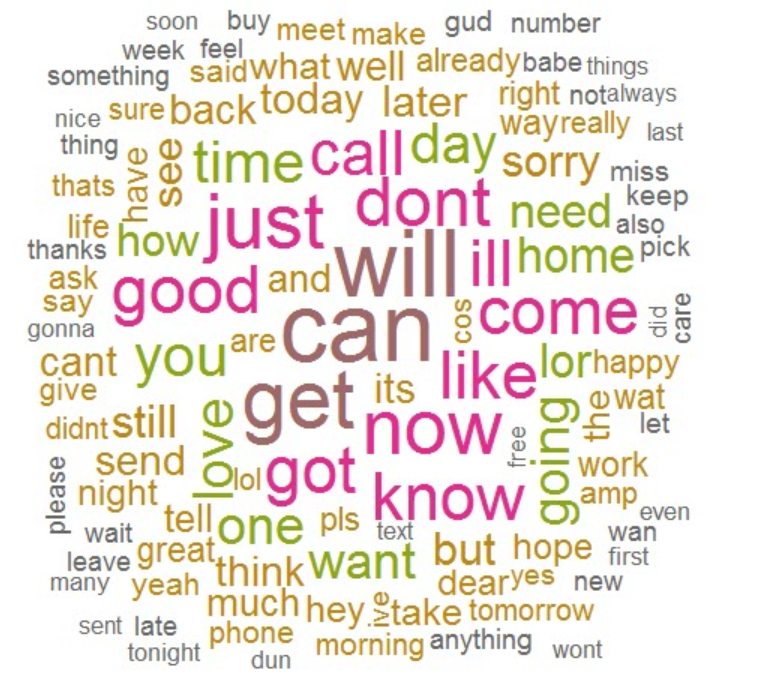
The Analysis

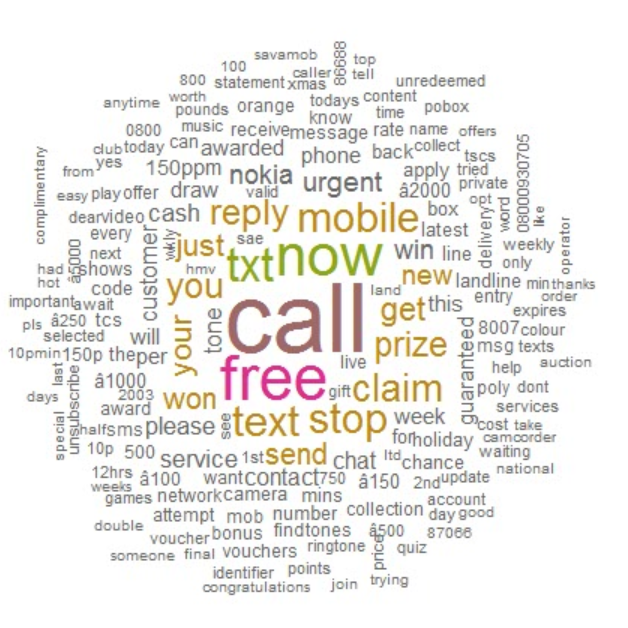
First step toward constructing the classifier involves processing the raw data for analysis. We uploaded data to R and our data includes 5574 observations and 2 variables. Data is basically a text document labelled as either spam or ham. Sms messages are string of text composed of words, spaces, numbers, and punctuations. We formed a corpus from the text document which refers to a collection of text document. In this project a text document refers to a single sms message.

Then we used (tm) package to handle the cleaning. We removed punctuations, stop words such as “and” , “but”, “or” and extra white spaces between words. Now our data is ready to become a sparse matrix which rows of the matrix indicates the sms messages and column indicates term that is words. Each cell in the matrix stores a number indicating a count of times the word indicated by the row.

Word Cloud

Our data now ready for the analysis but, before doing anything we tried to visualize the data by forming word clouds. We formed two separate word cloud for each class in the training data in order to see word pattern of each class. Below is the Word Cloud of ach class.





**Figure1. Word cloud for spam(right) and ham (left)**

From word clouds at Figure 1 we can see that non-spam (ham) sms are just composition of any word whereas in spam text we can see some spam indication of words such as urgent, free, call, claim, prize, cash, txt etc… These stark differences suggest that our naïve Bayes model will have some key words to differentiate between the classes.

Data Partition

As a next step we divided data into training and the test set by %75 training and %25 test partition ratio. We set an index to do the data partition we tried to have same percent spam and ham in the train and the test data.

| &nbsp; | Original | Training set | Test set |

|:----------:|:----------:|:--------------:|:----------:|

| \*\*ham\*\* | 86.6 | 86.6 | 86.6 |

| \*\*spam\*\* | 13.4 | 13.4 | 13.4 |

Bayesian model

Next we implemented Naïve Bayes algorithm. The algorithm will use the presence or absence of words to estimate the probability that given sms is spam. We used 10 fold cross validation and constructed two different model, the difference between the models will be that the first one does not use the Laplace correction and the training procedure figure out whether to user or not a kernel density estimate, while the second one fixes Laplace parameter to one (fL=1) and explicitly forbids the use of a kernel density estimate (useKernel=FALSE).

Model1 Prediction and accuracy

By using the Naïve Bayes model we predicted class for the test data and compared with the actual class by using confusion matrix.

Confusion Matrix and Statistics

Reference

Prediction ham spam

ham 1199 24

spam 7 162

Model2 Prediction and accuracy

In second model by using Naïve Bayes with fixed Laplace parameter to one (fL=1) and explicitly forbids the use of a kernel density estimate (use Kernel=FALSE) we got slightly different result.

Confusion Matrix and Statistics

Reference

Prediction ham spam

ham 1203 30

spam 3 156

K-NN Model

By using the same data set and same partition we applied K-NN method to classify spams by setting the number of k=1 and k=2, KNN also did good job for classifying spam.

|  |
| --- |
| K=1 confusion matrix k=2 confusion matrix  knn.pred ham spam  ham 1205 85  spam 1 101  Accuracy =0.938  knn.pred ham spam  ham 1202 54  spam 4 132  accuracy= 0.958 |
|  |
| |  | | --- | |  | |

Decision Tree model

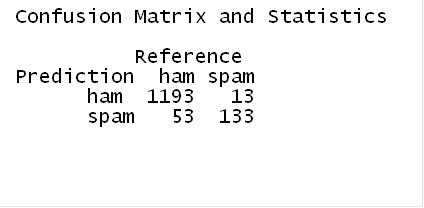
By using (rpart) and (c50) package we performed decision tree model on same data. Decision model gave better result than KNN below is the confusion matrix of both packages.

testlable ham spam

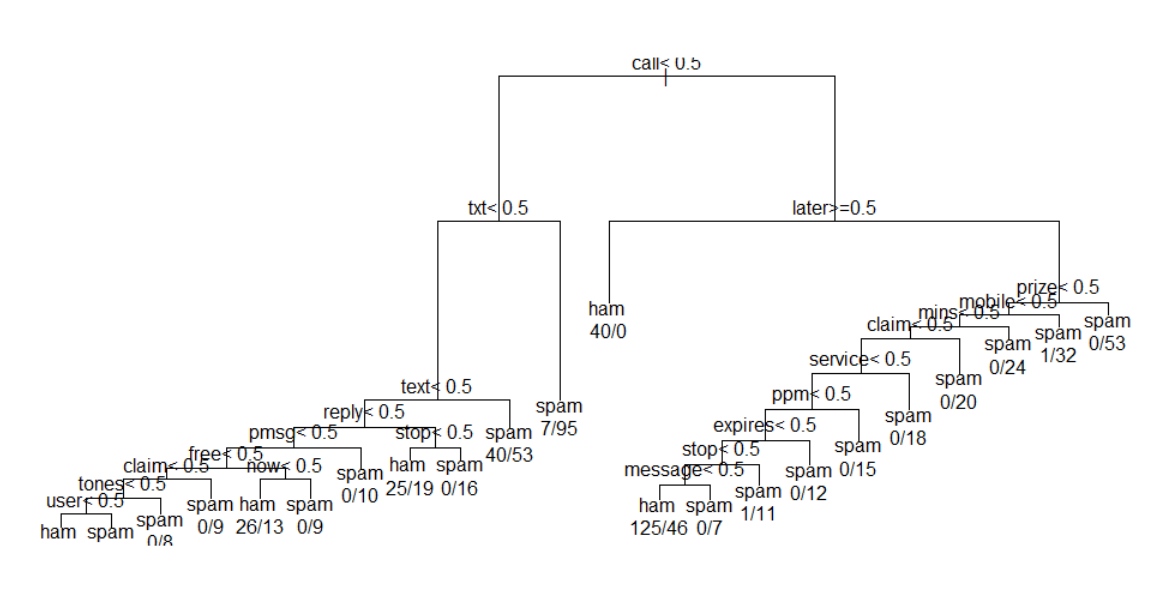
ham 1178 28

spam 63 123

Accuracy= 0.934 Kappa : 0.63



(rpart) Package result (c50) Package result



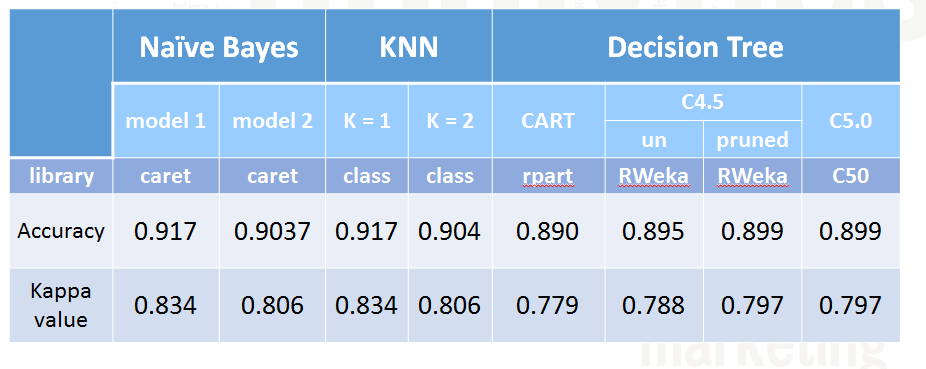
Model Comparison

Based on the model results we can say that our classifier did a good job on classifying spam on Sms text messages. Some of the models have done really good job whereas some gave slightly lover accuracy.



Considering all the model results above we can say that Naïve Bayes gives the highest accuracy as expected. In text mining area naïve Bayes and decision tree model gives higher accuracy than the other models. We had a chance to proof this. In Naïve Bayes we used Laplace smoothing and kernels Density separately to see the difference but in this particular project we didn’t see much difference.

We also had a chance to work on another data set which has almost same size Spam and Ham ratio. We followed the same procedure and got little lower accuracy. Below table is the result from second data set.



Conclusion

Over all, this project taught us a lot. First of all we saw how important text mining area can be. Second it was nice to see different machine learning algorithm application and their performances. We had a chance to work on a big dimensional data like text body of the text messages and learned how to handle it with application in R. Results of the classifications were very satisfying for us and we will continue digging more to explore new ways to achieve.