Choosing the Best Model for Classifying Spam Messages

BAIS: 6070:0700 Data Science Final Project

Amanda Aird, Cindy Huang, and Josh Hjelmaas

May 6th, 2021

1. Summary

The purpose of our analysis is to find a model that would allow us to filter out spam text messages. We used the perspective of a phone company that will make profit from filtering out spam messages but will lose money filtering our legitimate messages. We used a dataset of text messages labeled as spam or legitimate to build our models. We tested seven different model types on the data set and evaluated those models on AUC. While all our models had impressive results, Naïve Bayes and Random Forest models stood out as the top two predictive models. We found that the Random Forest model was the best model to predict and filter out spam messages based on our profit and lift curves.

1. Description of Dataset and Data Cleaning

Dataset Description

This dataset represents 5,572 instances of English text messages collected by the University of California Irvine (UCI) for SMS spam research. Each message is tagged as either ham (legitimate) or spam. According to UCI, these messages were sourced from a variety of areas ranging from the United Kingdom to Singapore. There are only two features for this dataset—the text of the SMS message and its spam (0) or ham (1) tag.

Distribution

We measured the distribution of our data in four ways: (I) the distribution of spam and ham messages; (II) the number of words in each message; (III) the number of characters in each message; and (IV) the average word length per message. These distributions are displayed below:

Chart, bar chart

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

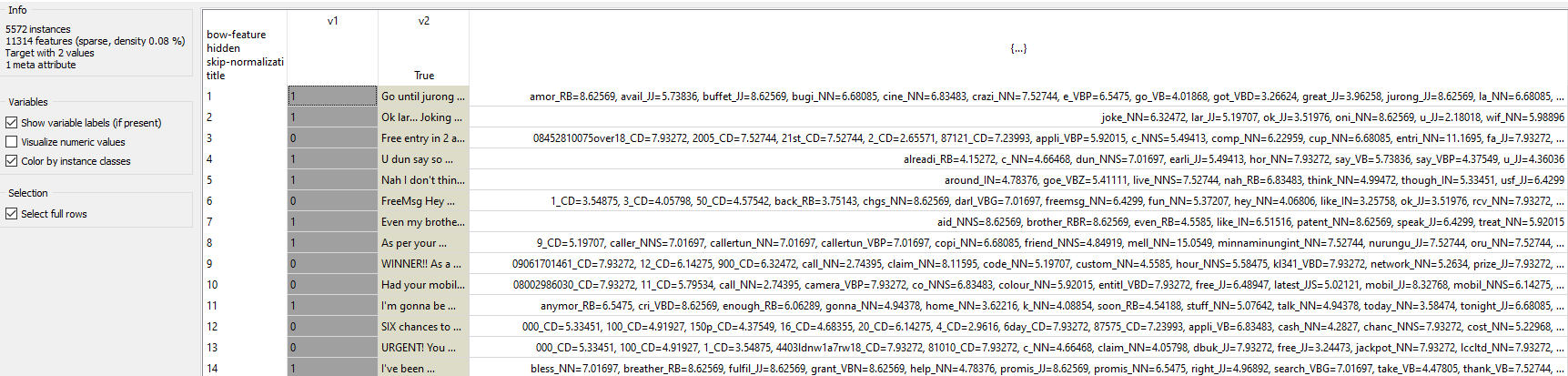
A picture containing shape

Description automatically generated

Overall, these plots tell us that the data skews heavily to the left. Messages are generally quite short, with the average message containing only 15 words, 61 characters, and almost 4 characters per word. Interestingly, there appears to be a greater proportion of spam messages above the averages for world length, word count, and characters per word.

Text Preprocessing

After doing some research, we were able to find an add-in called “Text-Mining” That would allow us to do text analytics. We used the “corpus” widget to transform words into a way to do statistical analysis and hypothesis testing. From there, we connected it to “preprocess text” to split the text into smaller units (known as tokens) to conduct transformation and TFIDF and vectorization of text. Transformation is turning everything into lowercase, getting rid of accents, etc. This makes sure that all of the words are formatted the same, so the actual analysis is not biased. After tokenizing the words, we find the TF-IDF value of each word. TF-IDF is a statistical measure that evaluates how relevant a word is in a document/corpus. It does this by counting how many times a word appears in a document. Text vectorization is the process of converting text into numerical representation. This is a critical step in order to apply machine learning to the text. We take all that into the “bag of words” which captured frequency of term in a document and use that to conduct or models. When connected to the “Data Table” widget it gave us an output like this:



All the text preprocessing done was stored into a new column. That new column contained all the words in the message and assigned a numerical value to it. It also tagged the word’s part-of-speech at the end.

Most Important Terms

Given the TF-IDF scores for each word, we were able to find which words were most important when performing spam classification. Below is a list of the 10 most important terms and a word cloud with those terms highlighted in red. Note that all terms have been stemmed and tagged with their part-of-speech (POS).

Table

Description automatically generated

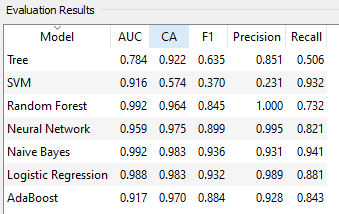
Text, whiteboard

Description automatically generated

1. Model Construction and Evaluation

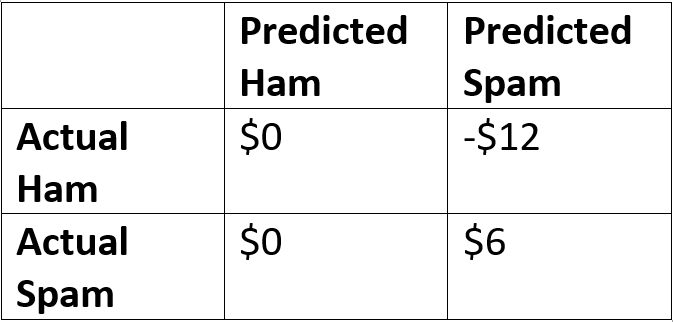
Model Construction

We constructed seven different models to evaluate: Logistic Regression (LR), Support Vector Machine (SVM), Naïve Bayes, AdaBoost, Neural Network, Decision Tree, and Random Forest. For logistic regression, we used a regularization type of Lasso(L-1) with c = 1 for strength. After running the regression, the AUC score was 0.982, which is a high score. Next, we constructed an SVM model. For initial SMV, I used an RBF kernel set to auto with a C = 1.00 and regression loss epsilon of 0.10. This gave me an AUC score of 0.916. After playing around with the parameters, we found that this was also the best combination. Our next model Naïve Bayes gave us an AUC score of 0.992. For AdaBoost, For the AdaBoost model, the parameters set were 50 estimators and a learning rate of 1.00. The classification method used was SAMME.R and the regression loss function was set to linear. The AUC score is 0.917. Our next model constructed was the neural network model. For initial logistic activation neural network, the parameters were set to hidden layers = 100 and regularization = 0.0001. We also set the maximal number of iterations to 600. These parameters gave an AUC score of 0.959. These were also the best parameters I found. Changing the number of hidden layers did not change the AUC score, and increasing regularization only made the AUC score go down. Next, we constructed a decision tree. We set the maximal tree depth to 5 and chose to stop classification when the majority reaches 95%. When lowering the classification threshold percentage, it would also lower the AUC score. With the threshold at 95%, we got an AUC score of 0.784. The last model we constructed the random forest model. With the random forest model, we set the number of trees to 500. With this model, we got an AUC score of 0.992. Looking at all the results, we found that our best models were Random Forest and Naïve Bayes.



Profit Structure

We built our profit structure with the assumption that the average phone plan will give us $120 in profit, but for every 20 customers that receive a spam message we lose one customer. So, by correctly filtering out 20 spam messages we keep one customer plan at $120, and we make a profit of $6 for each individual message filtered out. However, if we predict a message as spam and it was a legitimate message, we lose customers. We assumed that we would lose one customer for every 10 legitimate messages. Because of this we will lose $12 for every falsely predicted spam message.



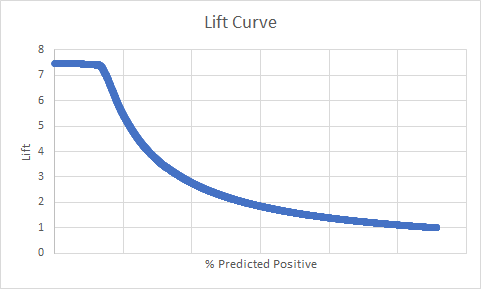
1. Choosing the Best Model

As we have discussed in class, the single best metric to use for comparing models is the AUC score. However, with our project the two best models (Naïve Bayes and Random Forest) had identical AUC scores of 0.992. To choose between these two, we decided to compare their performance plots and use the profit curve as a tiebreaker. We decided that the profit curve would be the next best comparison because our model’s ability to maximize profit would be the most important component of our project for a service provider.

Performance Plot Comparison

Naïve Bayes Random Forest

A picture containing line chart

Description automatically generated

With the lift curves, both models produce strong results. The only distinguishing trait between these two curves is that the ideal lift zone—the area at the beginning of the x-axis where lift is maximized—remains at a near-perfect constant for Random Forest while there is a slight trend downwards for Naïve Bayes in the same area.

Combined ROC Curve

Chart, line chart

Description automatically generated

The ROC curves produced similar results; both perform very well at virtually all thresholds, but the Random Forest model (depicted as the orange line) attains a near-perfect curve that just edges out Naïve Bayes.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Lastly, the profit curves show us that Random Forest again has an advantage, however slight. The maximum profit achieved for the Random Forest model is $4,110 at a 0.1996 threshold, which beats Naïve Bayes’ maximum of $3,804 at a threshold of 0.9042.

1. Recommendation

Our final recommendation based on our analysis is that we should use the random forest model and filter out messages with a threshold value of .1996. In our sample set of data, this would have resulted in a profit of $4,110. However, with such strong results on our ROC curves, it would be useful test our model on more data sets. Our original data set was from Kaggle, so it is possible that any preprocessing done before we saw the data resulted in the modeling appearing to be more accurate than it would have been on another data set. So, while the results from this analysis appear excellent, further analysis will be needed to verify the impressive results from our model.