Assignment 8

Ryan Lee

MSDS454 Sec55

11/22/20

For the following assignment we will be building a SPAM filter production model, by analyzing three models; a basic WOE binning classification, a Naïve Bayes model, and a logistic regression model. Utilizing training and test data provided within the assignment. Originally sources from the UCI Machine Learning Repository.

Starting with exploratory data analysis we observe, the initial dataset has 61 variables and a length of 4601 split into a training and test set with a length of 2318 and 58 variables and 2283 and 58 variables respectively. Next we will perform computational exploratory data analysis with the xgboost package. Extreme gradient boosting (xgboost) training utilizes gradient boosting to find the most relevant and predictive variables for identifying spam, using a 10 round 4 depth tree search to find the following variables as the most important for predicting spam. As seen in Figure 1 below:

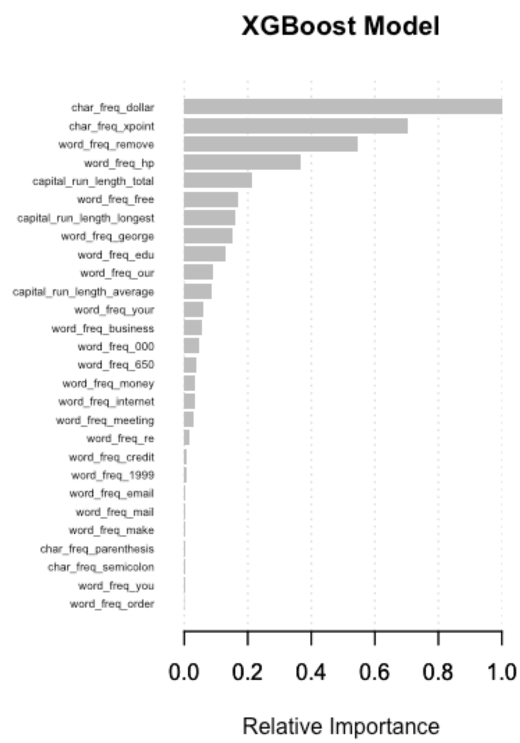


Figure 1

As we see in the figure the most important variables for detecting spamm are the following, $,!, remove, hp, capital run length total ending the list at 20% relative importance. As we can see from the variables, they typically involve finances and utilize symbols to illicit emotion like "!". Now that we have seen the most important features of the data set, we will next explore Weight-Of-Evidence Discretization.

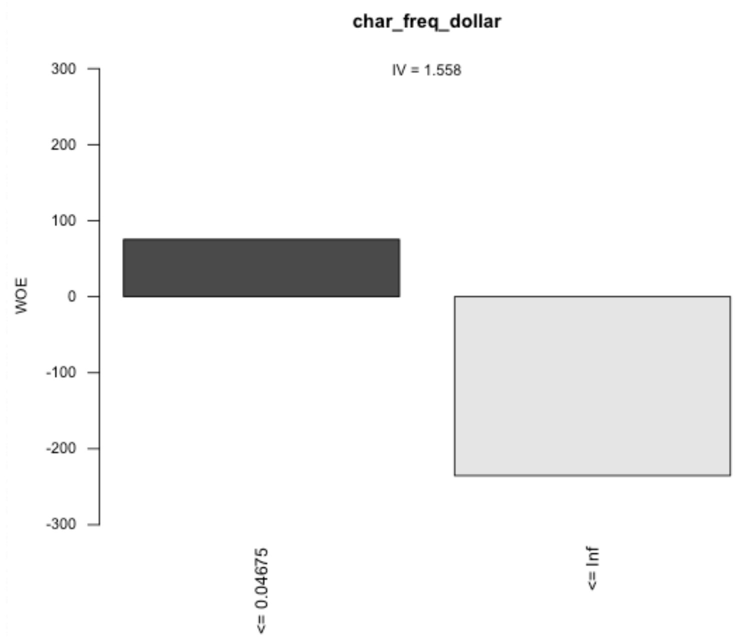


Figure 2

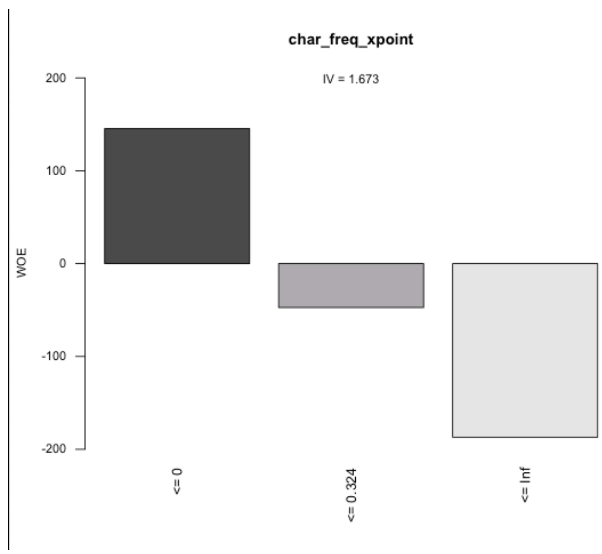


Figure 3

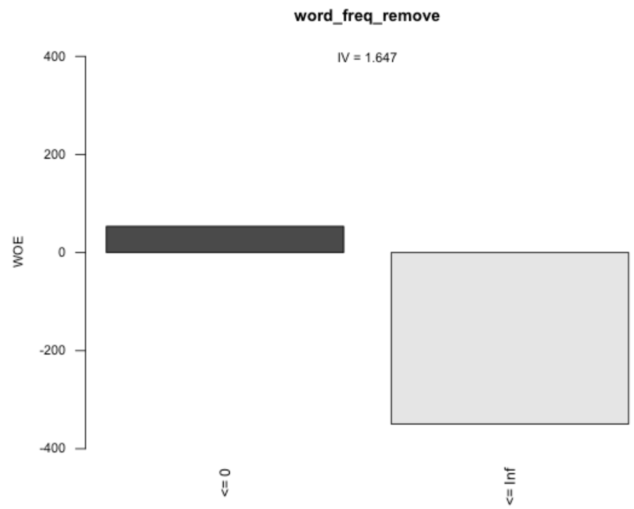


Figure 4

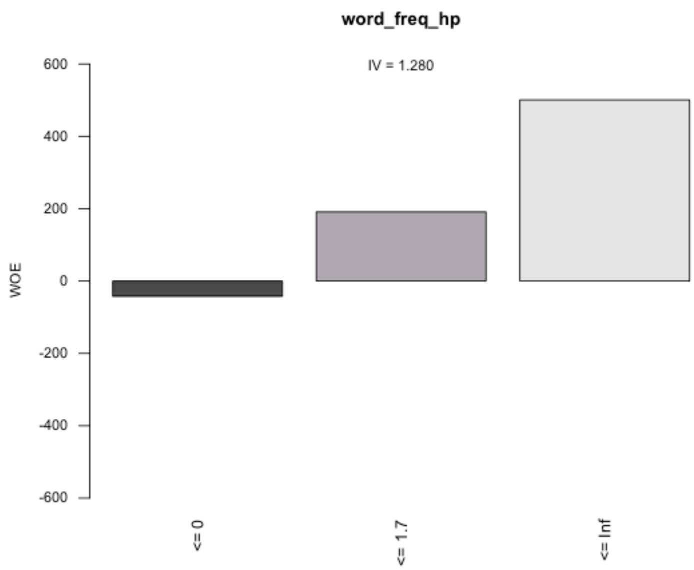


Figure 5

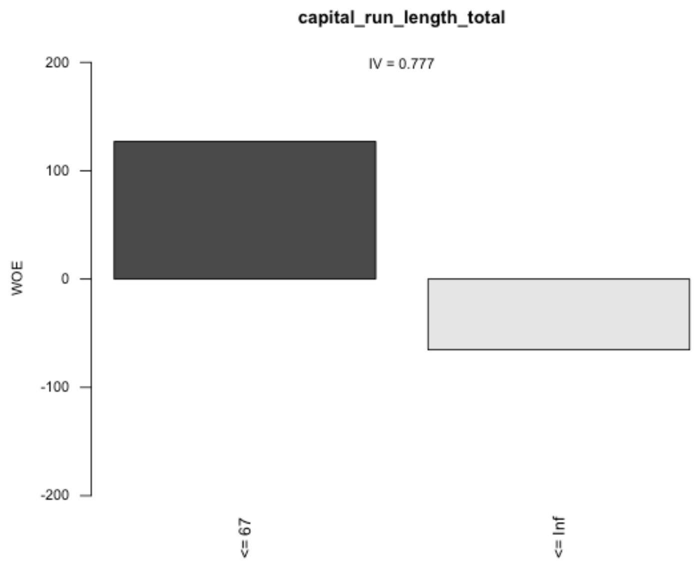


Figure 6

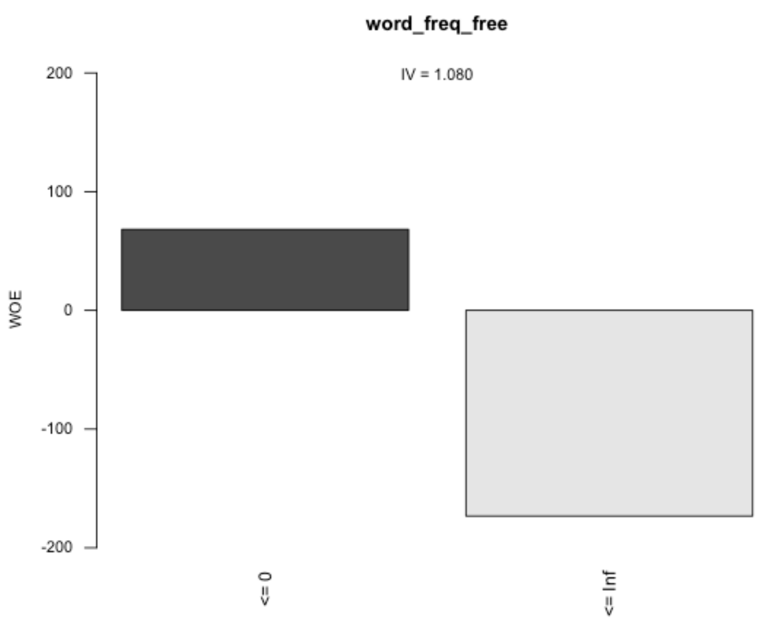


Figure 7

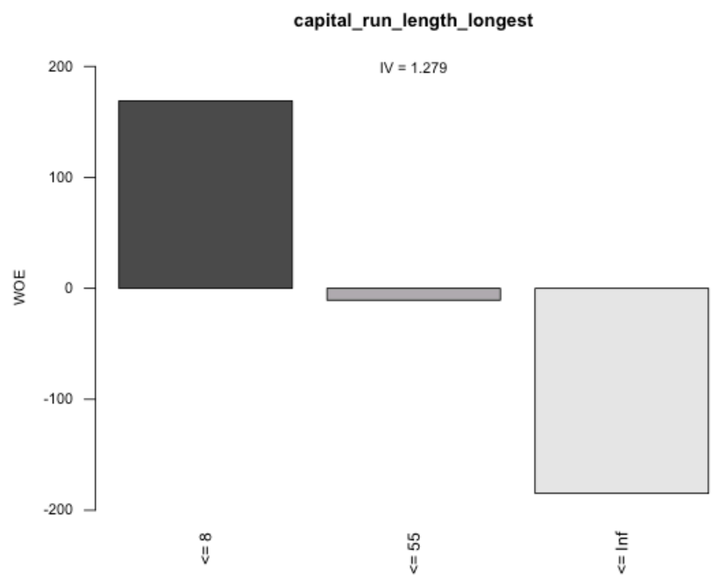


Figure 8

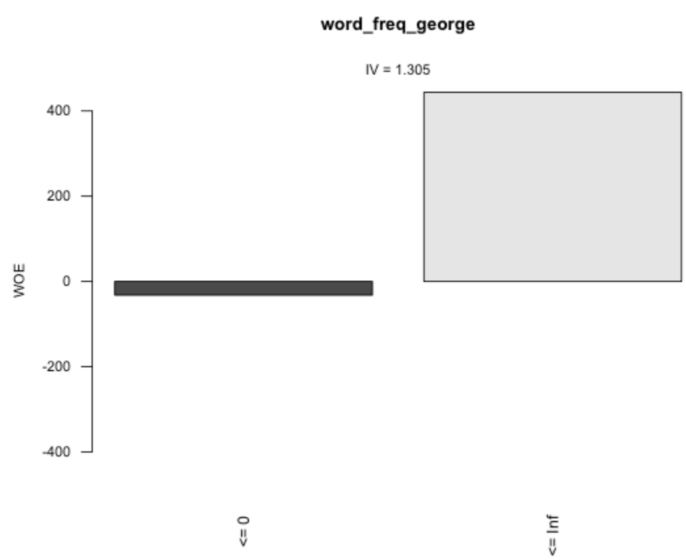


Figure 9

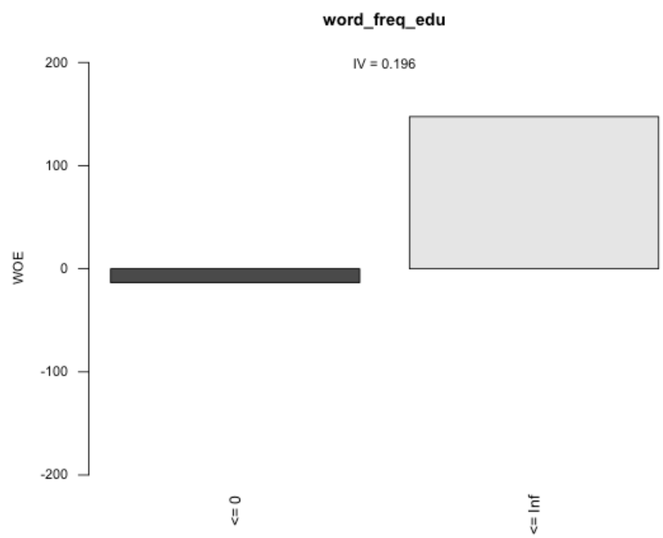


Figure 10

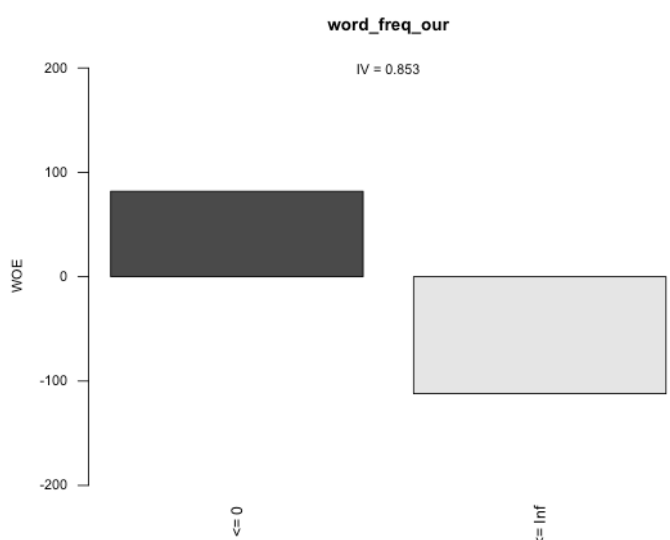


Figure 11

3.

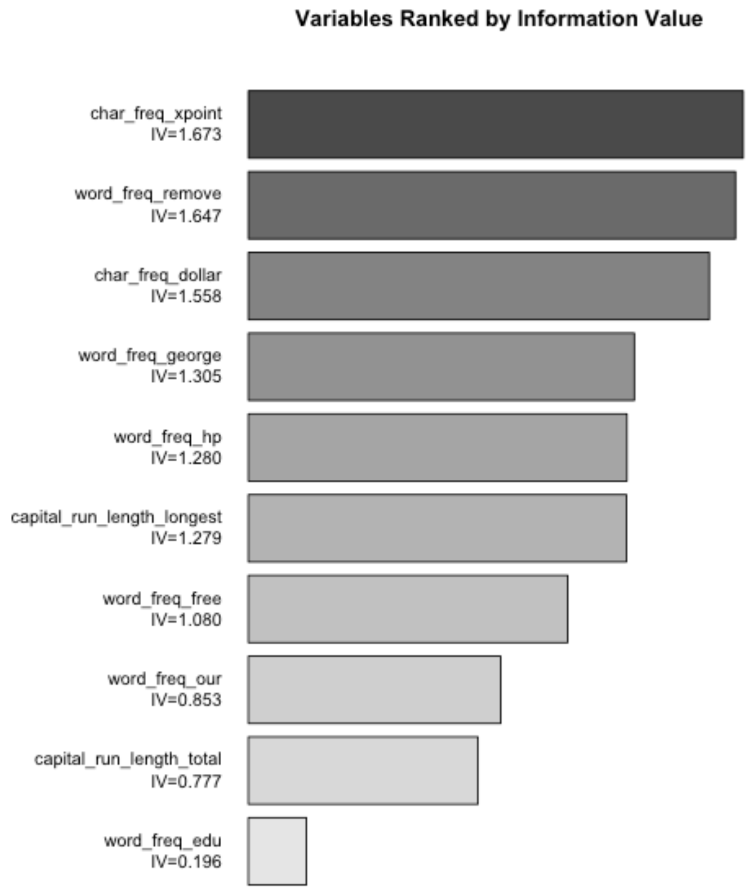


Figure 12

Rank	Variable	IV
1	Char_freq_xpoint	1.673
2	Word_freq_remove	1.647
3	Char_freq_dollar	1.558
4	Word_freq_george	1.305
5	Word_freq_hp	1.28
6	Capital_run_length_longest	1.279
7	Word_freq_free	1.08
8	Word_freq_our	0.853
9	capital_run_length_total	0.777
10	Word_freq_edu	0.196

Table 1

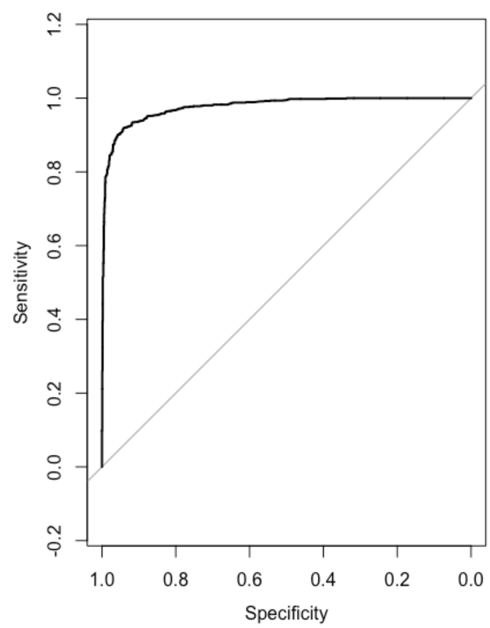
As we see in the table and graphs above the WOE ranking are different the XGBOOST findings Where exclamation point is now the highest information value vs #2 in XGBoost. Additionally, George moved up to 4 and HP moved down.

3.

Logitistic regression: we use logisitic regression to see the predictive ability of the model first with the gradient boosted variables. Which get the following results.

Model #1: Logistic Regression

	Dependent variable:
	spam
word_freq_remove	5.80***
	(0.78)
capital_run_length_total	0.001***
	(0.0002)
capital_run_length_longest	0.02***
	(0.003)
word_freq_edu	-3.51***
	(0.66)
char_freq_xpoint	0.42***
	(0.11)
word_freq_hp	-3.29***
	(0.44)
word_freq_free	1.26***
	(0.18)
word_freq_george	-7.41***
	(1.65)
word_freq_our	0.47***
	(0.10)
Constant	-1.25***
	(0.10)
Observations	2,318
Log Likelihood	-704.40
Akaike Inf. Crit.	1,428.80
Note:	*p<0.1; **p<0.05; ***p<0.01



With an AUC score of .9544 and a threshold of 0.3214, specificity of 0.868, and a sensitivity of 0.925. Finally the model provides the following confusions matrix

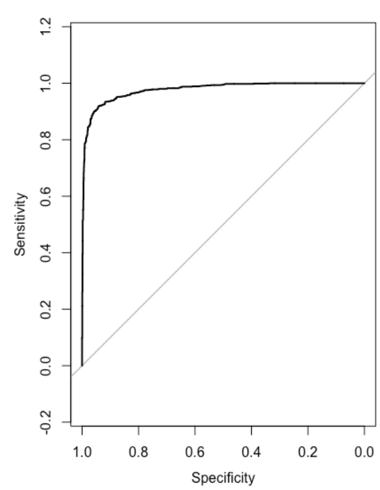
		Not Spam	Spam
Logistic Regression Gradient Boost	Not Spam	0.888	0.111
	Spam	0.0744	0.926

We see a false negative rate 7.4% and false positive rate of 13%.

When compared to model2 utilizing the WOE binning we see the following output.

Model #2: Logistic Regression

	Dependent variable:
	spam
char_freq_xpoint.binned(0,0.324]	0.13
	(0.22)
char_freq_xpoint.binned(0.324, Inf]	1.67***
	(0.22)
woe.char_freq_xpoint.binned	
word_freq_remove.binned(0, Inf]	2.66***
	(0.36)
woe.word_freq_remove.binned	
char_freq_dollar.binned(0.04675, Inf]	2.09***
	(0.24)
woe.char_freq_dollar.binned	
word_freq_george.binned(0, Inf]	-4.40***
	(0.88)
woe.word_freq_george.binned	
word_freq_hp.binned(0,1.7]	-3.14***
	(0.33)
word_freq_hp.binned(1.7, Inf]	-5.34***
	(1.03)
woe.word_freq_hp.binned	
capital_run_length_longest.binned(8,55]	1.47***
	(0.23)
capital_run_length_longest.binned(55, Inf]	1.81***
	(0.34)
woe.capital_run_length_longest.binned	
word_freq_free.binned(0, Inf]	1.23***
	(0.21)
woe.word_freq_free.binned	
word_freq_our.binned(0, Inf]	1.17***
	(0.20)
woe.word_freq_our.binned	
capital_run_length_total.binned(67, Inf]	0.62***
	(0.23)
woe.capital_run_length_total.binned	
word_freq_edu.binned(0, Inf]	-3.54***
	(0.42)
woe.word_freq_edu.binned	
Constant	-2.86***
	(0.18)
Observations	2,318
Log Likelihood	-456.57
Akaike Inf. Crit.	941.14
Note:	*p<0.1; **p<0.05; ***p<0.01

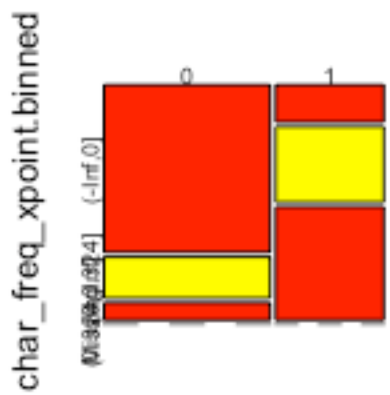


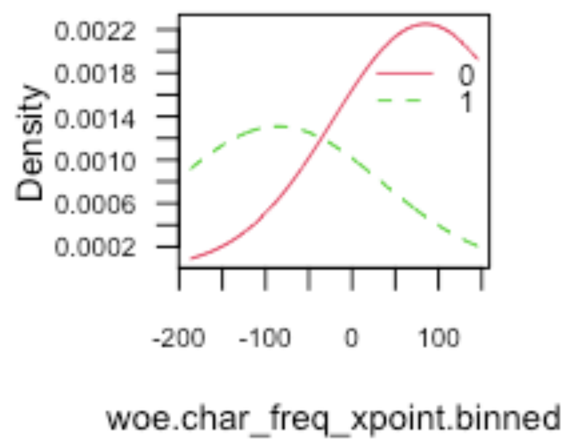
Here we see AUC score of .9769 and a threshold of 0.430, specificity of 0.9402, and a sensitivity of 0.920. Finally the model provides the following confusions matrix

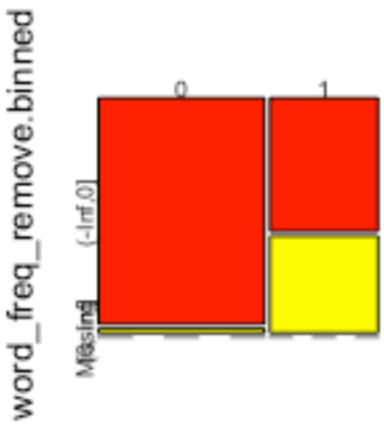
We see a false negative rate 8.3% and false positive rate of 7.0%.

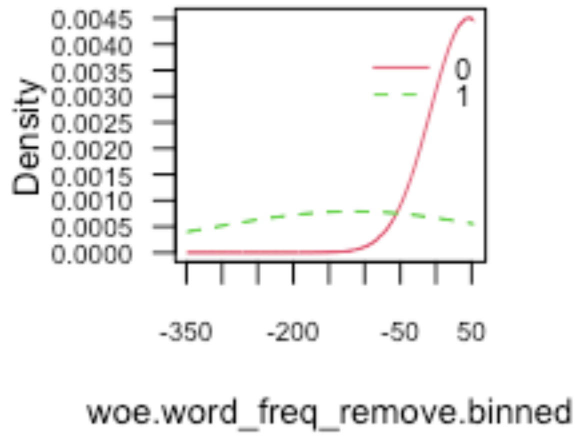
		Not Spam	Spam
Logistic Regression WOE Binning	Not Spam	0.93	0.07
	Spam	0.083	0.917

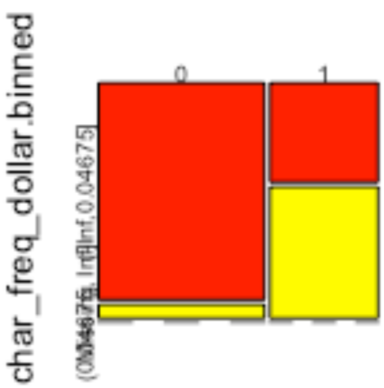
When producing a Naïve Bayes model we get the following plots:

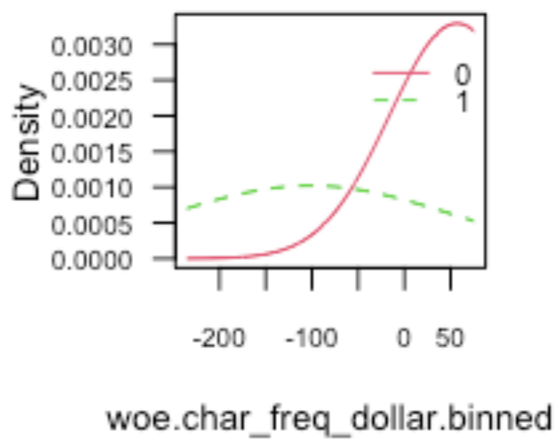


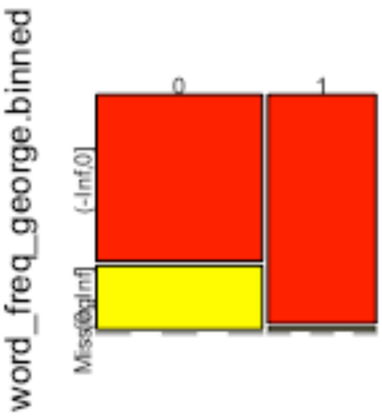


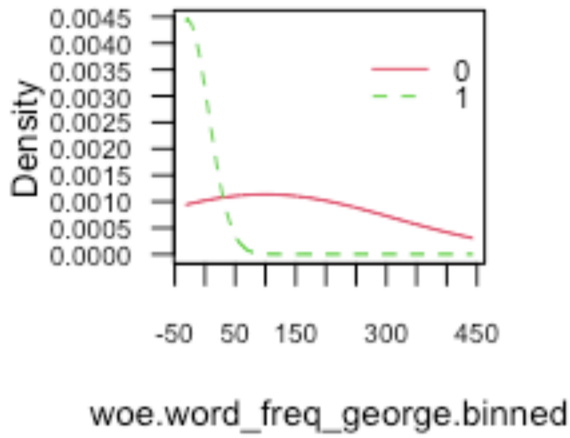

word_freq_remove.binned
0
1
(-Inf,0]
Missing

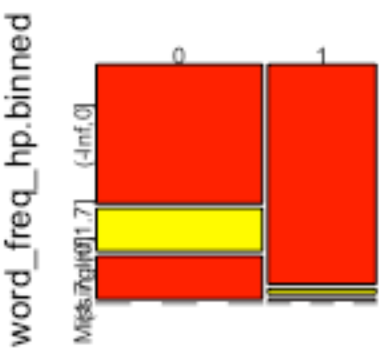

Density
0.0045
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
-350
-200
-50
50
0
1
woe.word_freq_remove.binned

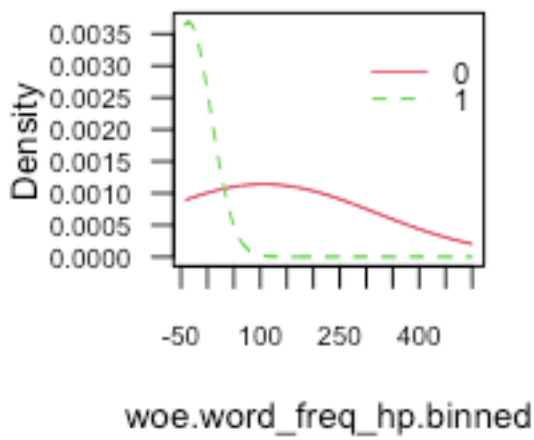

char_freq_dollar.binned
0
1

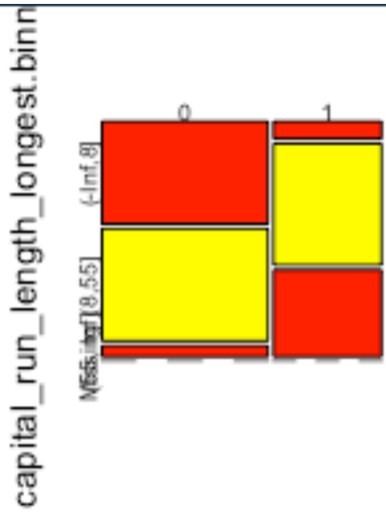
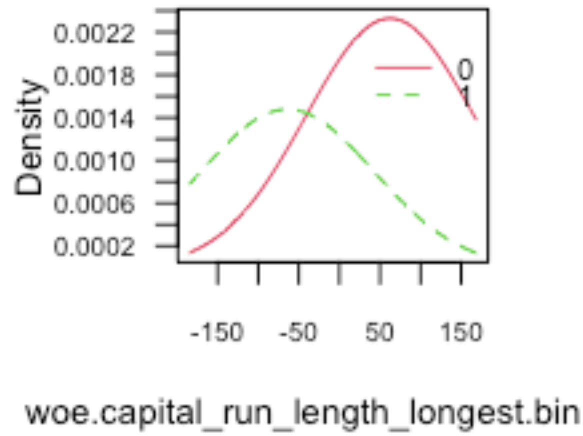
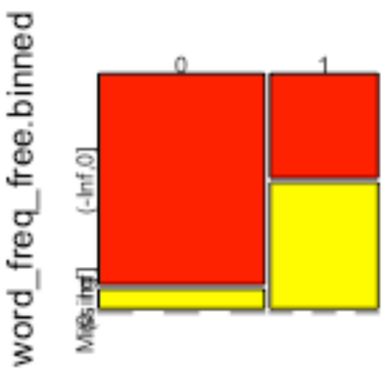
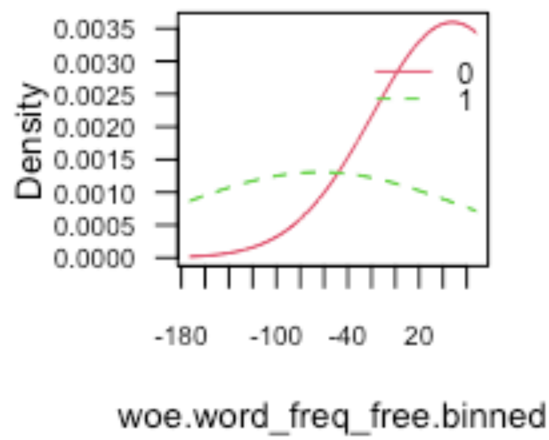

Density
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
-200
-100
0
50
0
1
woe.char_freq_dollar.binned

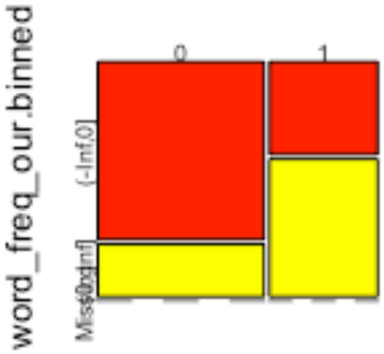

word_freq_george.binned
0
1
(-Inf,0]

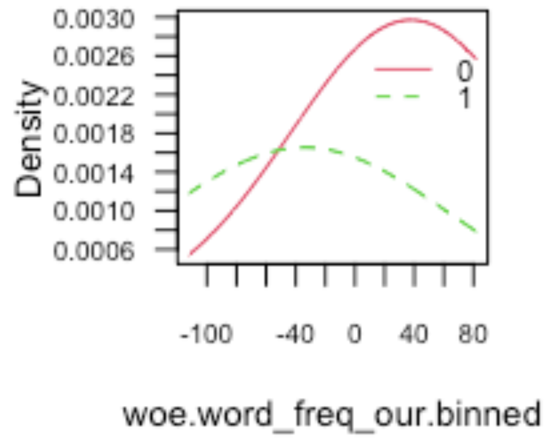

Density
0.0045
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
1
-50
50
150
300
450
woe.word_freq_george.binned

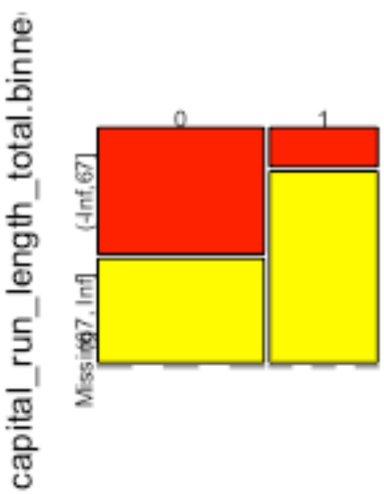

word_freq_hp.binned
0
1
(-Inf,0]

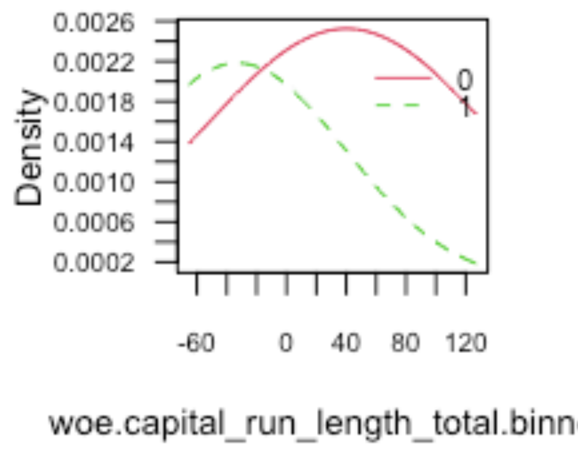

Density
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
1
-50
100
250
400
woe.word_freq_hp.binned

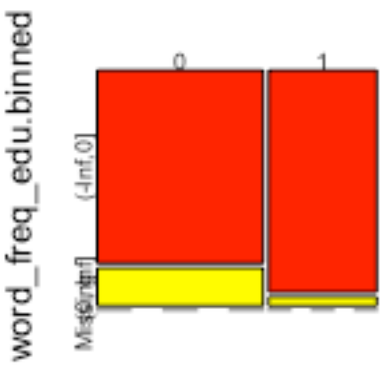

capital_run_length_longest.binn
0
1
(-Inf,8]

Density
0.0022
0.0018
0.0014
0.0010
0.0006
0.0002
0
1
-150
-50
50
150
woe.capital_run_length_longest.bin

word_freq_free.binned
0
1
(-Inf,0]

Density
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
1
-180
-100
-40
20
woe.word_freq_free.binned

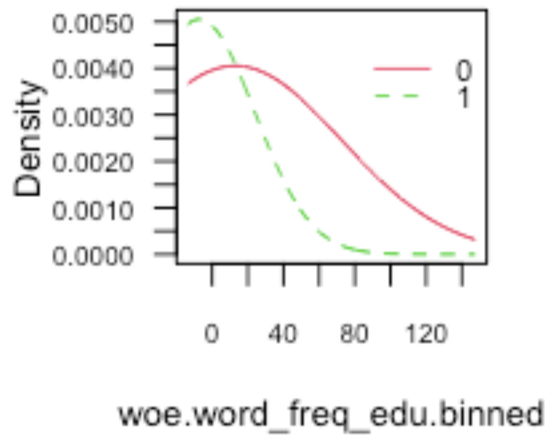

word_freq_our.binned
0
1
(-Inf,0]


Density
0.0030
0.0026
0.0022
0.0018
0.0014
0.0010
0.0006
-100
-40
0
40
80
0
1
woe.word_freq_our.binned


capital_run_length_total.binne
0
1
(-Inf,67]


Density
0.0026
0.0022
0.0018
0.0014
0.0010
0.0006
0.0002
-60
0
40
80
120
0
1
woe.capital_run_length_total.binne





As we see in the confusion matrix, the model has a false positive rate of 6.6% and a false positive rate of 11%

		Not Spam	Spam
Naïve Bayes	Not Spam	0.933	0.066
	Spam	0.11	0.89

Given the three models, it seems like WOE binning logistic regression is best. It has lowest false positive and negative results. Given the training data. Now we will predict with the test data and confirm the results.

Test			
		Not Spam	Spam
Logistic Regression Gradient Boost	Not Spam	0.888	0.111
	Spam	0.0744	0.926
		Not Spam	Spam
Logistic Regression WOE Binning	Not Spam	0.937	0.063
	Spam	0.094	0.905
		Not Spam	Spam
Naïve Bayes	Not Spam	0.944	0.0557
	Spam	0.106	0.893

As we see in the test results, the Naïve Bayes and Logistic Regression models were the best performing model on test data also.

In summary, the similarities in model performance is likely due to woe binning as both methods used the WOE binning, I think that the application of both models are interesting and given the similarities, we would need more data to make a decision for use in a production environment. However, is this was all the data, I would use the Naïve Bayes model because of the distributios used and as a result might be more robust.