Machine Learning A1 Write-up – James Nakos 1462474

Note: Throughout the report I refer to Class 0 as Non-Malicious and Class 1 as Malicious instead of 1 & 2. (Due to compatibility with code outputs)

Part 1:

1.1 Prior Probabilities:

The model has learned that the prior probabilities (assuming we know nothing about an instance) of it being in class 0 or 1 are:

- P(c = 0) = 0.8
- P(c=1) = 0.2

These are the proportions of messages of each class. Indicating that most of the training set is comprised of non-malicious messages, this may bias the model towards non-malicious predictions.

1.2 Most probable words: Figure 1.2.1

Class 0 Words	Class 0 Probabilities	Class 1 Words	Class 1 Probabilities
.	0.079304	.	0.056522
,	0.026024	!	0.024348
?	0.025576	,	0.023478
u	0.018916	call	0.020543
...	0.018749	£	0.013913
!	0.017182	free	0.010543
..	0.014943	/	0.00913
;	0.013152	2	0.008804
&	0.013096	&	0.008696
go	0.011137	?	0.008478

The most probable words for both classes are mainly punctuation and short words like “u” and “call”. The punctuation mostly overlaps and are therefore not suggestive of class which is something I expected. Aside from the punctuation which wont heavily affect our model due to the nature of Naïve Bayes (considers the predictive power of words), the most frequent words in the Malicious class are “call” & “free”. These are words commonly associated with malicious offers that offer free things and are facilitated over call; hence these may prove to be informative to the model.

1.3 Most predictive words

Figure 1.3.1

Most Predictive Words (Non-Malicious)	Probability Ratios (Non-Malicious)	Most Predictive Words (Malicious)	Probability Ratios (Malicious) R1/R0
;	60.49922	prize	99.05086957
...	57.49571	tone	64.09173913
gt	54.06313	£	49.71965217
lt	53.54824	select	46.61217391
:)	47.88449	claim	45.96478261
ü	31.92299	paytm	36.90130435
lor	28.83367	code	34.95913043
hope	24.71457	award	32.04586957
ok	24.71457	won	31.07478261
d	21.11036	18	29.1326087

Our most predictive words appear much more indicative of the message's intent than our frequent words which is to be expected. The malicious word "prize" and "£" clearly indicate intent of monetary exploitation and given high probability ratios as opposed to ":)" or "ok" which frequent non malicious messages and have high inverse ratios. The presence of these high ratio (predictive) 'words', will aid the model to classify messages by giving it information which helps the Bayes model to create probabilities for each class. Therefore, since we have a wide range of probability ratios, it should be possible to distinguish the classes

Part 2:

2.1:

Accuracy: 0.975

Confusion Matrix

	Class 0 Predicted	Class 1 Predicted
Class 0 True	785	15

Class 1 True	10	190

Precision: 0.93

Recall: 0.95

These metrics indicate that the model performed well, classifying 97.5% of instances correctly and this was achieved without relying on the majority class label to classify, (we only had 10 false negatives). Our recall was hence large (0.95), and we also had a small number of False Positives (15). Therefore, the model generalises well.

2.2: Figure 2.2.1

Proportion of unique words in vocab	0.9427178549664839
Proportion of unique words not in vocab	0.05728214503351615
Proportion of test words in vocab	0.9836797957695113
Proportion of test words not in vocab	0.016320204230488744

We observed no test instances without a word in our training vocabulary, so we could classify every item. To see how often our model saw out of vocab words w calculated the proportion of all test data words not in vocab ~0.016. Indicating that most instances should be easily classified.

2.3

A, Figure 2.3.1

Confidence Ratios (Class 0)	Text Instances (Class 0)
9.1349944516292e+37	time : rs. transaction number & & & & & & & & & ; ; ; ; ; ; ; ; ; lt lt lt lt lt # # # gt gt gt gt gt credit account reference decimal
2.6903818585609323e+29	? ? ? ? .. .. u u u u , , ... say person yes ! f : hello hello hello o o wen knw knw girl girl mean @ " " " " t name name g g n d d d d d d lift bt real dat h girlfrnd girlfrnd moral
3.1829245411853877e+25	. every & & & & & & ; ; ; ; ; ; lt lt lt # # # gt gt gt big hr

B, Figure 2.3.2

Confidence Ratios (Class 1)	Text Instances (Class 1)
1.3538895972837586e+20	. 4 + call Â£ - * holiday & urgent 18 t landline 150ppm cash cs await collection po box sae complimentary 10,000 ibiza
1.2870905388142676e+20	. 3 4 + ! call : Â£ offer * holiday & urgent 18 t landline 150ppm cash cs await collection po box sae tenerife 10,000
1.1491239652937078e+20	. . . , please order text call / : customer tone number [[service mobile]] colour colour thanks ringtone reference charge 4.50 arrive = red x49 09065989182

A,B Discussion:

The high confidence class 0 predictions do not appear to be “normal” text messages, they are unsurprisingly filled with punctuation which were also our most probable words. The model is therefore highly confident that instances like these are non-malicious, this could potentially be exploitable (an area to improve on). On the other hand, messages with high class 1 confidence are as expected, incorporating monetary language like “cash” which are commonly associated with malicious intent. This outlines that class 1 is likely easier to identify due to its more distinctive vocabulary while non malicious messages were more difficult.

C, Figure 2.3.3

R Values Closest to 1	Text Instances
1.0170981352199624	. call dear
1.0441124738285466	. reply glad
0.9297658433847452	. . tell return re order

The model is rightfully uncertain about these messages as they are short and hence do not give adequate information to the model to help classify them. We also have the presence of words that can be associated with both categories like “call”, leaving the model uncertain.

Part 3 (Option 2):

To improve my model from Q1, I chose to employ **Active Learning** which involved adding 200 records from the unlabelled dataset to the training data to enhance & diversify the training set. At first, I selected the 200 records at random and added them to the training data, then retrained the model. I firstly evaluated this on a random validation sample (20%) of the data.

Here were our results:

Accuracy: 0.965

Precision: 0.90

Recall: 0.94

Each of these were noticeable lower than how our original model performed. I decided that this was likely due to the fact that the random sampling does not ensure an equal distribution of classes in both the training and validation set. This could lead to the training set overfitting to the majority class and hence a loss of performance.

To combat this, I reevaluated this on a stratified validation set using the sklearn train_test_split function, this kept the proportions of the classes in both samples close to the same as the full dataset.

Results:

Accuracy: 0.975

Confusion Matrix:

	Class 0 Predicted	Class 1 Predicted
Class 0 True	315	5
Class 1 True	5	75

Precision: 0.94

Recall: 0.94

These results are extremely close to that of our first Naïve Bayes model and hence I needed to reselect 200 instances that were more informative to the model to see an improvement.

To do this I utilised the original model to examine all the unlabelled instances and select the 200 it was most uncertain about classifying. This was done by calculating the ratio of the posterior probability of each class given an instance. These were

hence instances that the original model was unconfident about labelling, so they should be more informative and help the model improve on similar inputs.

After retraining these were our validation evaluation results:

Accuracy: 0.98

Confusion Matrix:

	Class 0 Predicted	Class 1 Predicted
Class 0 True	317	3
Class 1 True	5	75

Precision: 0.96

Recall: 0.94.

These results outline that selecting more informative instances had a positive effect and reduced the number of False Positives, subsequently increasing our precision.

To further improve our model, I attempted to utilise decision entropies rather than R values to select my 200 instances as they better capture uncertainty which is what we are after, not just confidence. To do this I found the entropy of the two posterior probabilities predicted by our original model for each record and selected the records with the 200 highest entropy values. Slightly improving our results by reducing the number of False Positives as follows:

Accuracy: 0.9825

Confusion Matrix:

	Class 0 Predicted	Class 1 Predicted
Class 0 True	318	2
Class 1 True	5	75

Precision: 0.97

Recall: 0.94

I also attempted other improvements to no avail, such as adjusting the Laplace smoothing parameter alpha (which we left as 1), I tried smaller values to increase the model's reliance on rare words like 0.5 which reduced our accuracy to 0.977, and larger values to improve generalisation like: alpha=1.4, which kept the same accuracy but increased precision to 0.99. However, I chose to reject using this value as it

lowered recall to 0.93 which is something I identified as important as we want to minimise false negatives to prevent malicious messages from going undetected.

Lastly, I evaluated the model on our test data:

Accuracy: 0.977

Confusion Matrix:

	Class 0 Predicted	Class 1 Predicted
Class 0 True	785	15
Class 1 True	8	192

Precision: 0.93

Recall: 0.96

This supports what we saw in the validation data that the use of Active Learning improves the model's performance by focusing on uncertain instances. The accuracy, precision and recall values here are an improvement to our original model's performance as expected.

Part 4:

4.1:

The test data performance above clearly outlines that utilising Active Learning was an improvement (although minor) to our original supervised model. While the original model had an accuracy of 97.5%, the semi supervised classifier achieved 97.7% as well as an even higher 98.25% on our validation data. Suggesting a small performance gain was achieved by employing active learning. The largest performance gain came from the semi supervised model's ability to classify malicious messages more effectively, we had 2 less False Negatives which in turn increased our recall. The minor increase can still be viewed as positive, since generally, adding data that the model is unsure about could create noise and make it harder for the model to establish decision boundaries for easier to classify instances. The fact that we saw only positive effects indicates the method's usefulness, which I believe could be further enhanced by possibly weighting our 200 values more (eg: using upsampling,

something I ran out of time to experiment on) since 200 is quite small relative to the dataset's size.

Overall, while the performance increases were minor, the semi-supervised model demonstrated its helpfulness in increasing recall, as well as its marginally better performance on a stratified validation set (when using the 200 selected instances vs random), hence the model is robust and generalises well to real world data distributions.

4.2

Phase	Class Ratio	Average Confidence
Before Training	c0/c1	9.1350e+34
Before Training	c1/c0	5.9150e+17
After Training	c0/c1	3.2162e+17
After Training	c1/c0	7.7935e+14

Figure 4.2.1: Average confidence before/after semi-supervised training

As illustrated in figure 4.2.1, the average confidence in classifying instances for both classes dropped quite significantly. Especially for classifying non-malicious instances (exponent decreased from 34 to 17), at base level this doesn't seem desirable as our model is less confident with assigning labels and hence our results are less certain. However, in the context of our model, this is desirable. After adding our previously uncertain instances, our model now knows that some test instances won't be as easy to distinguish, adjusting its probabilities to account for this.

Therefore, our semi supervised model is now less overconfident (our average confidence ratios were and still are extremely large either way), this is a sign of improved generalisation as the model no longer commits to overly simplistic patterns, better handling edge cases. The drop in recall is likely attributed to this, we avoided labelling some ambiguous messages as non-malicious (with high confidence as seen in our table) and correctly classified them as malicious instead.

Figure 4.2.2 (Before Active Learning)

Class	Word	Ratio
0	;	60.49922
0	...	57.49571
0	gt	54.06313
0	lt	53.54824
0	:)	47.88449
0	Ã¼	31.92299
0	lor	28.83367
0	hope	24.71457
0	ok	24.71457
0	d	21.11036
1	prize	99.05087
1	tone	64.09174
1	£	49.71965
1	select	46.61217
1	claim	45.96478
1	paytm	36.9013
1	code	34.95913
1	award	32.04587
1	won	31.07478
1	18	29.13261

Figure 4.2.3 (After Active Learning)

Class	Word	Ratio
0	gt	49.38611165142968
0	lt	48.86072748492511
0	:)	36.77689165531998
0	...	28.68597549114958
0	Ã¼	27.845360824742265
0	;	27.582668741489982
0	lor	23.116903326201125
0	ok	17.863061661155417
0	d	17.863061661155417
0	hope	17.863061661155417
1	prize	87.55497963717141
1	tone	51.39096630877453
1	claim	36.79847
1	select	36.16401332839689
1	paytm	36.16401332839689
1	£	33.62618783166729
1	ringtone	28.55053683820807
1	won	26.647167715660867
1	code	26.647167715660867
1	18	24.743798593113663

A similar effect can be observed when looking at our most predictive words: for class 0, the top words stayed mostly the same but with lower predictiveness ratios. Indicating a drop of the model's reliance on extremely predictive words to classify.

For class 1, the high impact words remained dominant (like 'prize' and 'tone') and their ratios only dropped a small amount: like from ~29 to ~24 for '18'.

This similarly unveils that the model's understanding of Class 0 instances was the most affected by Active Learning, likely due to their diverse nature as opposed to malicious messages' more limited vocabulary.

Overall, this helps the model balance its over confidence in predicting specifically class 0 instances and better handle ambiguous unseen data.