ACTIVE LEARNING WITH A MULTINOMIAL NAïVE BAYES MODEL FOR SCAM MESSAGE DETECTION

Analysis and Performance Comparison

Yi Ding

Introduction:

This report presents an analysis of a multinomial Naïve Bayes model applied to scam message detection, with a focus on model representation and the impact of an active learning strategy. Detecting scam messages effectively is important for protecting users from fraud and financial loss. Therefore, reliable automated detection methods highly are significant. By examining the model's predictive features, class likelihoods, and confidence measures, we identify how the model forms its decision boundary between scam and non-malicious messages. We then introduce a semi-supervised approach that utilises active learning to select informative unlabeled instances for retraining. Through comparison of the original supervised model and the active learning-enhanced model, we evaluate performance improvements and discuss potential biases introduced by the active learning strategy.

1. Supervised model training

1.1 Prior probability

Class	Prior probability
Scam	0.2
Non-malicious	0.8

Table 1.1: Prior probability table

In Table 1.1, the model learned that scam messages are significantly less likely to appear. Consequently, the model tends to classify a message as non-malicious when the words have similar likelihood values for both classes. This is reasonable, as real-world messages also contain few scams.

1.2 Most probable words for each class

Rank	Word	Likelihood
1	.	0.07930378329975375
2	,	0.026024177300201477
3	?	0.025576449518692635
4	u	0.0189164987687486
5	…	0.018748600850682785
6	!	0.017181553615401836
7	..	0.014942914707857623
8	;	0.013152003581822252
9	&	0.013096037609133646
10	go	0.01113722856503246

Table 1.2: Top 10 most probable words for class non-malicious

Rank	Word	Likelihood
1	.	0.05652173913043478
2	!	0.02434782608695652
3	,	0.023478260869565216
4	call	0.020543478260869566
5	£	0.01391304347826087
6	free	0.010543478260869566
7	/	0.009130434782608696
8	2	0.008804347826086956
9	&	0.008695652173913044
10	?	0.008478260869565218

Table 1.3: Top 10 most probable words for class scam

In Tables 1.2 and 1.3, the model appears to learn that different punctuations, symbols, and keywords have varying likelihoods across classes. Notably, the exclamation mark '!' ranks higher in the scam class than in the non-malicious class, suggesting the model recognizes that scam messages tend to use attention-grabbing punctuation. Additionally, it learns that symbols and words such as '£', 'call', and 'free' are commonly associated with scam messages. This is reasonable, as real-world scams often aim to provoke a response by implying monetary gain or free offers.

1.3 Most strongly predictive words for each class

Rank	Word	Probability ratio *R*
1	;	60.49921647638236
2	…	57.4957092754272
3	gt	54.06312961719275
4	lt	53.548242668457576
5	:)	47.88448623237072
6	ü	31.922990821580477
7	lor	28.833669129169465
8	ok	24.714573539288114
9	hope	24.714573539288114
10	d	21.110364898141928

Table 1.4: Top 10 most strongly predictive words for class non-malicious

Rank	Word	Probability ratio *R*
1	prize	99.05086956521738
2	tone	64.09173913043477
3	£	49.71965217391304
4	select	46.61217391304348
5	claim	45.96478260869565
6	paytm	36.901304347826084
7	code	34.95913043478261
8	award	32.04586956521739
9	won	31.074782608695653
10	18	29.13260869565217

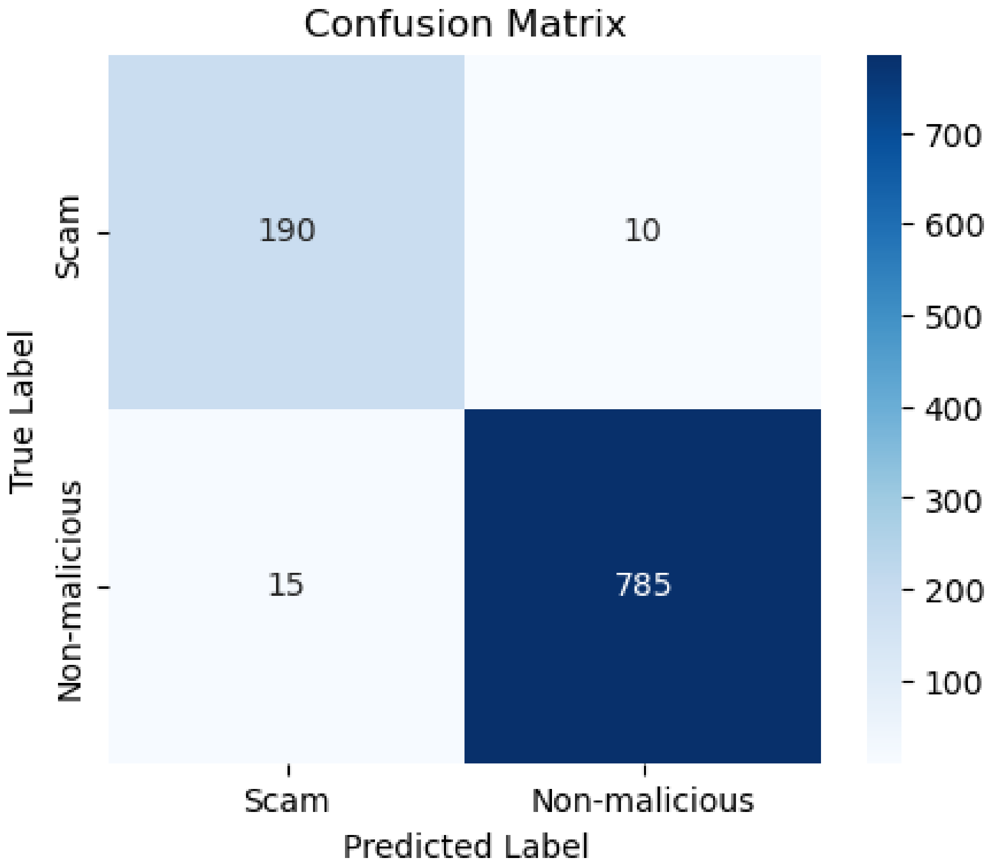
Table 1.5: Top 10 most strongly predictive words for class scam

The values of the probability ratio R, are defined as the predicted class's posterior probability divided by the other class's posterior probability. A larger R indicates a more confident prediction.

Although the top predictive words for the non-malicious class mostly lack clear semantics, the model appears to identify scam messages through meaningful content words. In Table 1.5, nouns such as 'prize', 'code', and 'award' are generally associated with benefits, while imperative verbs like 'select', 'claim', and 'won' encourage the receiver to take action. This suggests that the model's choice of strong predictive words aligns with the intent of real-world scams. Therefore, there is strong evidence that the multinomial Naïve Bayes model can distinguish between the two classes, as its ranking of predictive words for scams appears reasonable.

2. Supervised model evaluation

2.1 Evaluation metrics and OOV



Accuracy	0.975
Precision	0.927
Recall	0.95
F1 score	0.938

Table 2.1: class 'scam' is treated as the positive

Precision	0.987
Recall	0.981
F1 score	0.984

Table 2.2: metrics for class 'non-malicious'

(The following write-up will treat 'scam' as the positive)
Based on the confusion matrix, the model does not require much evidence to classify an instance as positive, as it shows higher recall than precision. This is acceptable, as recall is the more important metric in this context—falsely classifying scam messages as non-malicious can be hazardous in real-world applications. In Tables 2.1 and 2.2, the non-malicious class appears easier to predict, with higher precision and recall compared to the scam class.

The model encountered out-of-vocabulary (OOV) words in 14.2% of the test instances, though no test items were skipped. This suggests that the current vocabulary list used by the model is insufficient.

2.2 Analysis of confidence values for the model's prediction

Rank	Text	Prediction	Confidence *R*
1	time : rs. transaction number & & & & & & & & & & ; ; ; ; ; ; ; ; ; ; lt lt lt lt lt # # # gt gt gt gt gt credit account reference decimal	Non-malicious	9.134994451628908e+37
2	? ? ? ? .. .. 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	Non-malicious	2.690381858561381e+29
3	. every & & & & & & ; ; ; ; ; ; lt lt lt # # # gt gt gt big hr	Non-malicious	3.182924541185421e+25
4	, get like second half & & & & ; ; ; ; lt lt # # gt gt run though almost whole gram gram usually	Non-malicious	6.035603737741724e+20
5	u , , lor … … … … food food eat den oso haha well depend mon n la wana okie okie cheap chinese gd ex	Non-malicious	3.822169718420451e+19

Table 2.3: Top 5 high confidence instances classified as non-malicious

Rank	Text	Prediction	Confidence *R*
1	. 4 + call £ - * holiday & urgent 18 t landline 150ppm cash cs await collection po box sae complimentary 10,000 ibiza	Scam	1.3538895972838023e+20
2	. 3 4 + ! call : £ offer * holiday & urgent 18 t landline 150ppm cash cs await collection po box sae tenerife 10,000	Scam	1.2870905388143088e+20
3	. . . , please order text call / : customer tone number [[service mobile]] colour colour thanks ringtone reference charge 4.50 arrive = red x49 09065989182	Scam	1.1491239652938098e+20
4	. call £ £ guarantee won customer prize prize claim service 1000 yr 2000 representative cash 10am-7pm	Scam	9.079669916728702e+19
5	. . 2 free u + ! 1st / / wk wk txt tone gr8 hit 150p 16 poly 8007 8007 nokia nokia nokia tones polys	Scam	7.580465977853623e+19

Table 2.4: Top 5 high confidence instances classified as scam

The model learns to classify messages based on strongly predictive words, as outlined in Tables 1.4 and 1.5. Due to the multinomial distribution assumption, the confidence values are likely inflated by the repeated occurrence of strongly predictive words such as ";", "&", and "gt" (Table 2.3). This is problematic, as it may allow real-world scams to evade detection by repetitively including such punctuations to artificially boost confidence for the non-malicious class. In contrast, the high-confidence instances classified as scams appear reasonable, as these messages typically contain cues related to free money or offers—through strongly predictive words like "£" and "call"—which are commonly used in scams.

Rank	Text	Prediction	Confidence R
1	. call dear	Non-malicious	1.017098135219966
2	. reply glad	Non-malicious	1.0441124738285463
3	. . tell return re order	Scam	1.0755396179747463
4	? ur * just alrite sam	Scam	1.07975667795837
5	. . reply send person right ! code confirm sort bank acc	Scam	1.1338198625776805

Table 2.5: Top 5 near boundary instances (R near 1)

The model seems to struggle with classifying short messages, as all low-confidence predictions in Table 2.5 are short. This is reasonable, since shorter messages contain less information for the model to learn from and make a decisive prediction.

3. Extending the model with semi-supervised training

3.1 Instance selection criteria

Since the goal is to select instances that provide the most information, out-of-vocabulary (OOV) score $V(x_i)$ and uncertainty score $U(x_i)$ are chosen as the explanatory variables for an instance's importance.

The importance of a test instance x_i is given by $I(x_i, \alpha, \gamma)$:

$$I(x_i, \alpha, \gamma) = (1-\alpha)\big(V(x_i)\big)^{\gamma} + \alpha\big(U(x_i)\big)^{\gamma} \mid \alpha \in [0,1], \gamma \in [1, \infty)$$

OOV score:

$$V(x_i) = \frac{\log_e(count(OOV\ words\ in\ x_i) + 1) - min_v}{max_v - min_v}$$

$$min_v = min\{\log_e\big(count\big(OOV\ words\ in\ x_j\big) + 1\ \big) \mid j = 1,2, \dots, n\}$$

$$max_v = max\{\log_e\big(count\big(OOV\ words\ in\ x_j\big) + 1\big) \mid j = 1,2, \dots, n\}$$

where n = number of test instances

Min-max normalization is sensitive to outliers, so log transformation is used to prevent instances with unusual number of OOV words from distorting the rescaling.

Uncertainty score:

Similarly, apply log transformations to confidence values.

$$C(x_i) = \frac{\log_e\big(R(x_i)\big) - min_c}{max_c - min_c}$$

$$min_c = min\{\log_e\left(R\big(x_j\big)\right) \mid j = 1,2, \dots, n\}$$

$$max_c = max\{\log_e\left(R\big(x_j\big)\right) \mid j = 1,2, \dots, n\}$$

Where $R(x_i) > 1$ is the confidence value.

The uncertainty score is therefore given by:

$$U(x_i) = 1 - C(x_i)$$

$V(x_i)$ and $U(x_i)$ are normalized to a scale of $[0,1]$. This preprocessing prevents the feature that has greater range to dominant the selection process.

Skipped instances have $U(x_i) = 1$ as they are assumed to have maximum uncertainty.

	Correct Predictions	Incorrect Predictions
Average uncertainty score	0.853	0.945

Table 3.1: The average uncertainty score for supervised model prediction results in section 2

Based on Table 3.1, the incorrectly predicted instances generally have higher uncertainty score. Therefore, $I(x_i, \alpha, \gamma)$ is implemented to be positively correlated with both $V(x_i)$ and $U(x_i)$.

The hyperparameters are used to control the preference of each score. Tuning α would adjust the selection to have the best balance between OOV and uncertainty rating. Tuning γ would adjust the level of preference for extreme values of $V(x_i)$ or $U(x_i)$. Hence, the formula for $I(x_i, \alpha, \gamma)$ allows flexibility for selection.

3.2 Methodology for hyperparameter tuning

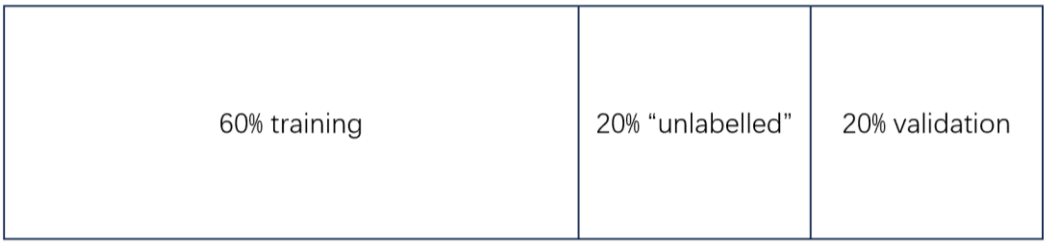


Figure 3.2: Partition for hyperparameter tuning

The entire training dataset is randomly partitioned into three parts (Figure 3.2), with the validation set used solely for model evaluation. This prevents test data leakage during training and allows assessment of the model’s generalization ability.

Stratified sampling is applied to ensure that all partitions maintain the same class proportions. This ensures each set is representative of the original data and prevents the model from overfitting to a class that might otherwise be over-represented.

A guide model NB_guide, is trained on the training set and used to generate predictions on the “unlabelled” set to compute $I(x_i, \alpha, \gamma)$. For each combination of (α, γ), the top 20% most important “unlabelled” instances (equivalent to selecting 200 out of 1000) are selected. A new model is then trained on the combination of these selected instances and the original training set, and evaluated on the validation set.

A single holdout strategy is employed because the time complexity increases rapidly: a total of $a \times b$ models must be trained and tested, where a and b are the number of candidates for α and γ, respectively. Thus, single holdout is used to trade off evaluation accuracy for reduced computational cost.

3.3 Results on the validation set

γ α	0.1	0.3	0.5	0.7	0.9
1	0.9476	0.9453	0.9453	0.9453	0.9476
1.5	0.9476	0.9453	0.9453	0.9453	0.9577
2	0.9476	0.9453	0.9453	0.9553	0.9577
2.5	0.9476	0.9453	0.9553	0.9577	0.9577
3	0.9476	0.9553	0.9553	0.9577	0.9577

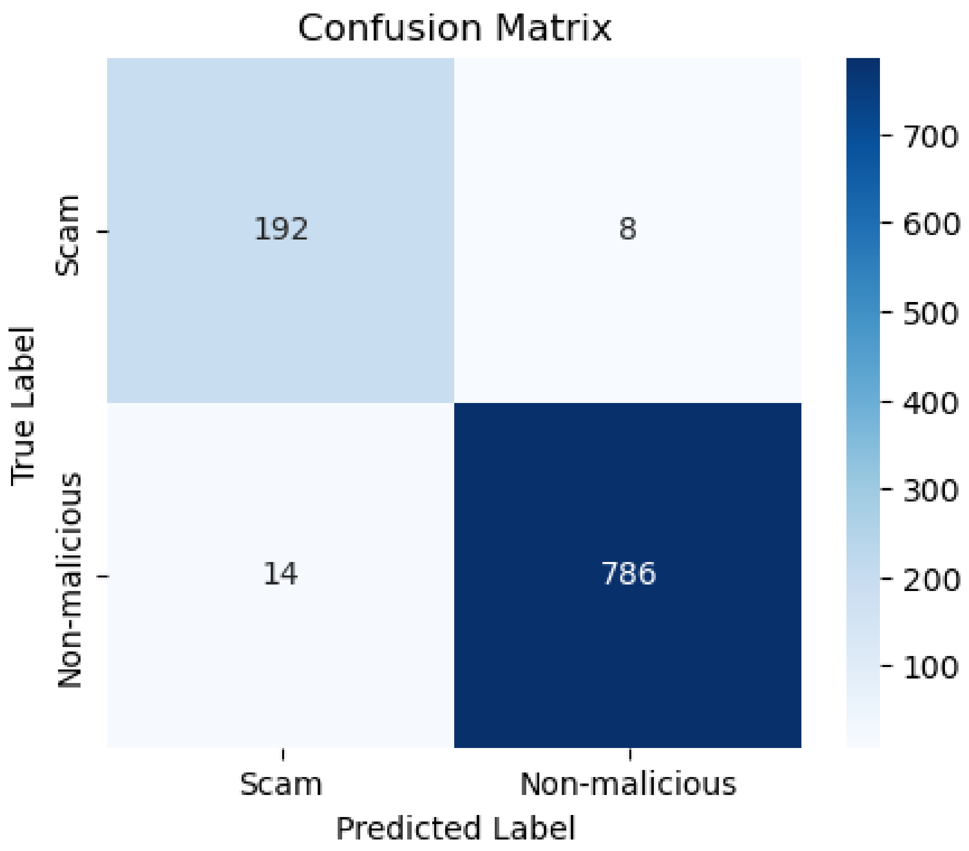
Table 3.3: F2-score for each model’s performance on the validation set

The evaluation metric F2-score is used to assess the model’s performance for each (α, γ). By definition, F2-score prioritizes recall while maintaining a reasonable balance with precision, making it well-suited to the goal of scam detection.

In Table 3.3, the retrained model performs better for large values of α and γ. As a result, the chosen hyperparameters are $\alpha = 0.9$ and $\gamma = 3$, meaning the selection process prioritizes instances with high uncertainty scores (i.e. low confidence). The semi-supervised model is then trained using the full training dataset along with 200 additional instances selected by the supervised model (Section 1), ranked according to $I(x_i, 0.9,3)$

4. Semi-supervised model evaluation

4.1 Model performance comparison



Accuracy	0.978
Precision	0.932
Recall	0.96
F1 score	0.946

Table 4.1: class 'scam' is treated as the positive (semi-supervised)

Precision	0.99
Recall	0.983
F1 score	0.986

Table 4.2: metrics for class 'non-malicious' (semi-supervised)

(The following write-up will treat 'scam' as the positive)
Based on the available test data, the semi-supervised model shows improved performance compared to the supervised model. Notably, recall for the 'scam' class increased from 0.95 to 0.96 (Tables 2.1 and 4.1), suggesting that the semi-supervised model may be more sensitive in detecting scam messages. Precision also increased from 0.927 to 0.932, indicating no apparent trade-off between precision and recall. Overall, the semi-supervised model appears to outperform the supervised model across all metrics.

The semi-supervised model encountered OOV words in 12.3% of the test instances, with no test items skipped. This reduction from 14.2% suggests that the model's vocabulary has expanded, providing it with more information to handle diverse inputs.

However, the level of improvement does not appear statistically significant. Based on the confusion matrices, the semi-supervised model correctly classified two additional ‘scam’ instances and one additional ‘non-malicious’ instance. These differences may result from the randomness inherent in the sampling process for the given test dataset. Therefore, more test data would be needed to provide stronger evidence for the observed gains in recall and precision.

4.2 Model representation comparison

Class	**Prior probability**
Scam	0.198
Non-malicious	0.802

Table 4.3: Prior probability table (semi-supervised)

The selection process for the semi-supervised model may introduce bias into the prior probability distribution of classes. As shown in Table 4.3, the prior probability for ‘non-malicious’ exceeds 0.8. This shift suggests that the selection process tends to favor instances labeled ‘non-malicious.’ If the model were retrained on more data using the same approach, this bias could gradually distort the prior probabilities.

Rank	**Word**	**Probability ratio *R***
1	gt	54.21789507563567
2	lt	53.701534170153415
3	:)	49.054286020813215
4	…	34.59618066731038
5	ü	32.01437613989915
6	;	30.336203197081858
7	lor	28.91621070700569
8	ok	24.785323463147733
9	hope	24.785323463147733
10	d	21.17079712477202

Table 4.4: Top 10 most strongly predictive words for class non-malicious (semi-supervised)

Rank	**Word**	**Probability ratio *R***
1	prize	98.76812798670268
2	tone	63.908788697278204
3	select	46.47911905256597
4	claim	45.83357573239144
5	£	41.63754415125701
6	paytm	36.79596924994806
7	code	34.859339289424476
8	18	30.98607936837731
9	won	30.98607936837731
10	ringtone	30.98607936837731

Table 4.5: Top 10 most strongly predictive words for class scam (semi-supervised)

Similar to the supervised model, the strongly predictive words for ‘non-malicious’ mostly lack clear semantics. In Table 4.4, the rankings and probability ratios of these words have changed, but the top 10 predictive words remain the same as in the supervised model (Table 1.4).

There are notable changes in the semi-supervised model’s strongly predictive words for ‘scam’. The word ‘ringtone’ appears in the top 10 for the semi-supervised model (Tables 4.5 and 1.5). This suggests that the semi-supervised model has reinforced its ability to identify scam messages related to ringtone services.

Text	**Supervised model prediction**	**Semi-supervised model prediction**	**Truth**
. . . . . , great ok call / / - - per noida @ full plot available near book start onwards down	Non-malicious Confidence: 1.35	Scam Confidence: 2.06	Scam
? 1 2 u u hope time / £ & 2nite luv c alone want to chat xx cum calls	Non-malicious Confidence: 2.52	Scam Confidence: 2.85	Scam
? ur * just alrite sam	Scam Confidence: 1.08	Non-malicious Confidence: 1.20	Non-malicious

Table 4.6: All test instances that were classified differently by the two models

The semi-supervised model appears to handle test instances near the boundary between the two classes better. In Table 4.6, it classifies these boundary cases correctly and with higher confidence than the supervised model. This suggests that the semi-supervised model's parameters (words' likelihoods) are likely more finely tuned.

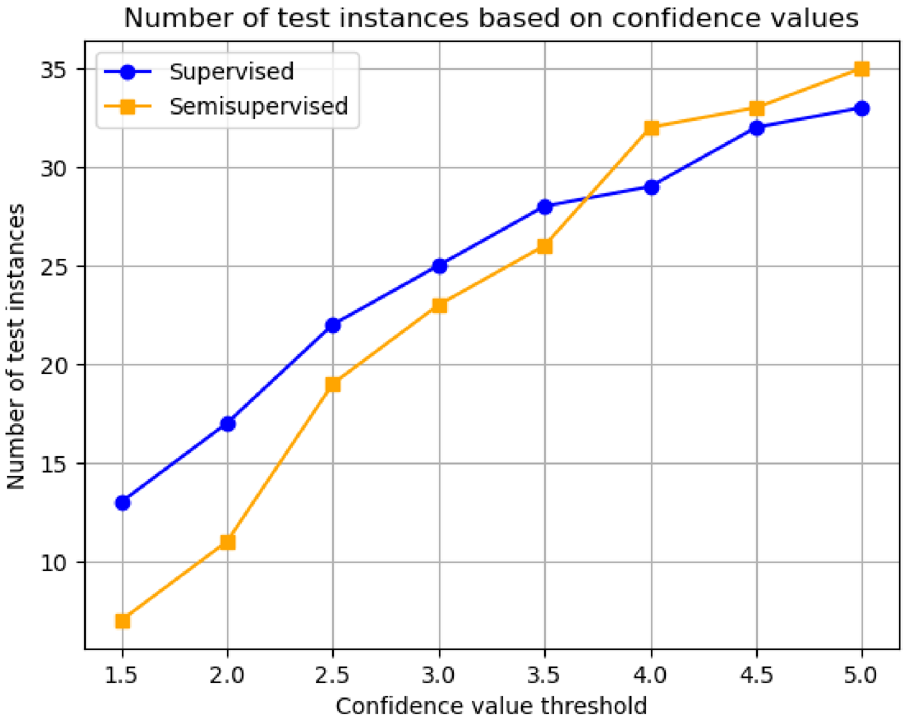


Figure 4.7: number of test instances fall under each confidence value threshold

The semi-supervised model appears to produce fewer highly uncertain predictions. Using a low confidence threshold of $R = 2.0$, there are 17 test instances below this confidence level in the supervised model, compared to 11 in the semi-supervised model (Figure 4.7). This shift suggests that the semi-supervised model may have better separation of the feature space (words' likelihoods) between the two classes.

5. Limitation: Independence Assumption of Naïve Bayes

A key limitation of the multinomial Naïve Bayes model is its assumption that words occur independently given the class label. The current model treats each word in a message as an independent sample from a probability distribution of possible words. This is not really a correct model for natural language – words in a message are not independent and word order matters. This issue could be addressed through exploring models that incorporate word dependencies, such as n-gram based approaches.

Conclusion:

Upon inspection of the model's representation, the multinomial Naïve Bayes model identifies scam messages by relying on strongly predictive features such as monetary terms and attention-grabbing punctuation. In the semi-supervised setting, recall and precision both show slight improvements, suggesting a more sensitive decision boundary for scam detection. However, these improvements are not statistically significant, and the active learning process may introduce bias in class priors. Future work could explore improved preprocessing techniques such as stop word removal. Additionally, models that account for word dependencies could be investigated. Evaluation on larger, more diverse datasets would also help validate the approach and assess its generalisability.