COMP 30027 Machine Learning Assignment 1

Aditya Sarkar
1354041
assarkar@student.unimelb.edu.au

Question 1

the initial training dataset with 200 observations was split roughly such that 20% were scam messages while the other 80% were not. See prior probabilities in **Table 1 below.**

P(scam)	P(no_scam)
0.2005	0.7995

Table 1: Prior probabilities in sms_supervised_train.csv

Now to calculate the most probable words in each class we divide the frequency of each word by the total number of words in the class. The results are in **Table 2** below. It can be seen that some of the most probable 'words' or tokens in both classes are punctuation. For example "?" and "!". This is expected as the preprocessing steps strip these characters from the rest of the string such that they are treated as their own token, and as they are used very commonly in speech and text, they will have high probabilities in both classes. Apart from this, there are significant differences in the types of words that have high probabilities in scam texts versus non scam texts. Specifically, non-scam words are clearly more casual and short hand than the scam words.

Rank	Top Scam Words	Scam Probability	Top Non-Scam Words	Non-Scam Probability
1	!	0.028993	?	0.03246
2	call	0.024463	u	0.024363
3	free	0.012555	!	0.021806
4	2	0.010484	go	0.014135
5	?	0.010096	get	0.012572
6	claim	0.00919	s	0.008381
7	customer	0.008413	gt	0.007458
8	u	0.008413	lt	0.007387
9	txt	0.008284	call	0.007245
10	ur	0.007766	ok	0.006819

Table 2: Words with highest probabilities of appearing in a given text in each class.

To calculate the words most indicative of whether a specific text is a scam or not, the probability of a word in each class is compared to the other class. The results are in **Figure 3.** The words most predictive of a scam text are words that relate to inciting an action (e.g. prize, claim, code, award etc.), words relating to rewards or selection. Once again the words most predictive of non scam texts are more colloquial and short hand. These are reasonable results. Thus, because some words are more indicative of one class over another, a naive bayes model will be able to distinguish between the classes with confidence in most cases by multiplying the token probabilities for each in each text together and comparing them.

Rank	Most Predictive Scam Words	Scam Ratio	Most Predictive Non-Scam Words	Non-Scam Ratio
1	prize	92.936707	gt	0.017355
2	tone	60.135516	lt	0.017522
3	select	43.734921	lor	0.032541
4	claim	43.127492	ok	0.037964
5	paytm	40.090344	hope	0.037964
6	code	32.801191	d	0.043388
7	award	30.067758	da	0.046725
8	won	29.156614	let	0.052065
9	18	27.334326	wat	0.053597
10	2000	27.334326	oh	0.056947

Figure 3: Words most predictive of whether a text is a scam or not.

The trained model correctly classified 965 instances in the test set, 30 were incorrect and 5 were omitted as all of the tokens in the instance were out of the trained vocabulary. The total accuracy of the model is therefore approximately 97%. For the scam class the accuracy was 91% while for non scam texts it was 98.4%. Precision and recall metrics for the scam and nonscam classes have been outlined in **Table 6&7.** Notably, both of the precision and recall metrics were lower for the scam class. This suggests that the model is both less accurately able to identify scam messages (compared to nonscam messages) and misses more scam messages than nonscam messages.

Metric	Value
Correct	965
Incorrect	30
Omitted	5
OOV Hits	154
Accuracy	0.9698

Table 4: classification outcomes and out of vocabulary hits

	Predicted Scam	Predicted Non-Scam
Actual Scam	TP = 182	FN = 18
Actual Non-Scam	FP = 12	TN = 783

Table 5: confusion matrix

Metric	Value
Precision	0.9381
Recall	0.91
F1 Score	0.9239

Table 6: SCAM metrics

Metric	Value
Precision	0.9775
Recall	0.9849
F1 Score	0.9812

Table 7: non scam metrics

Text instances with very high confidence in model classification are listed in **table 8**. The patterns outlined in the previous question are further highlighted in the high confidence

classifications. Words which incite an action or try to elicit a response by implying a reward or prize are most likely to be strongly classified as a scam. On the other hand, texts with informal or short hand message are likely to be classified as non-malicious. Finally, texts with no strong connotation to either class (i.e. used frequently in both classes or very infrequently) are likely to have very low confidences in their classification. Some examples of such tokens are “yes” and punctuation such as “?” and “!”.

Index	Processed Text	Confidence Ratio	Classification
553	please call award apply end rs vodafone todays...	4.059043×10^{18}	Scam
348	please call award apply end rs vodafone todays...	3.044282×10^{18}	Scam
453	4 ! call 150 holiday urgent 18 t landline cash...	2.705159×10^{17}	Scam
225	weekend ! call show rs customer prize claim co...	6.906612×10^{16}	Scam
644	? please call award end 350 todays voda number...	6.615189×10^{16}	Scam
223	? ? ? ? u u u u say person yes ! f hello hello...	8.829842×10^{18}	Non-malicious
656	? ? ? 7 8 up up u u u u pick like late already...	1.028223×10^{17}	Non-malicious
340	time rs transaction number lt lt lt lt lt gt gt...	3.728096×10^{16}	Non-malicious
16	? ? send get like stop time time time feel pho...	1.222816×10^{16}	Non-malicious
42	u lor thk say say e nite v lar jus happen noth...	7.576520×10^{13}	Non-malicious

Table 8: non scam metrics

Index	Processed Text	Confidence Ratio
118	? ? 1 2 3 3 4 u text win meet greet currently ...	1.033115
980	yes chat	1.057513
962	university	1.094

Table 9: non scam metrics

Question 3

In this section the model was extended using a semi-supervised method in which the original model is used to classify unlabelled data. This new, now classified, data is then added to the original training data and the model is retrained. In order to test how the model could be further improved, a variety of modified iterations of the final combined training set were made. First a baseline was established in which the new dataset was used to train a new model. Second, we omit any low confidence pseudo-classifications from the new training set. Finally, we take the top 50% of the pseudo-classified set with the highest confidence and reclassifying the unlabelled set, before retraining on the entire unlabelled set. The results of each of these iterations have been listed in **Table 10.** The model with the highest accuracy was the one which excluded the lowest confidence pseudo-classifications at ~97.3% total accuracy which was higher than the original model in question 2 as well as the baseline model as described above. What was not expected was that the iterative approach did not necessarily result in an improvement in the model accuracy with one model even scoring worse than the original model.

It could be hypothesised that because in each iteration of the iterative approach we can be more sure that the pseudo classifications are true, it is likely to produce a better model, but this was not observed. One hypothesis for this was that the initial test of the iterative approach simply took the highest confidence classifications without regard of the distribution of scam texts to noscam texts in the new training set. Upon, further analysis of the distribution of confidence ratios within each pseudo class, it was observed that scam texts were much more likely to have a higher confidence score than non-scam texts. See **figure 1&2.** As such, if the top 50% most confident pseudo classifications were blindly taken, the majority, or at least an unrepresentative amount of the initial iteration would be scam messages. However, when this was corrected such that the final training set would have approximately ~20% scam texts and ~80% noscam texts, reflective of the original training set, only a minor improvement was seen. **See Table 10**

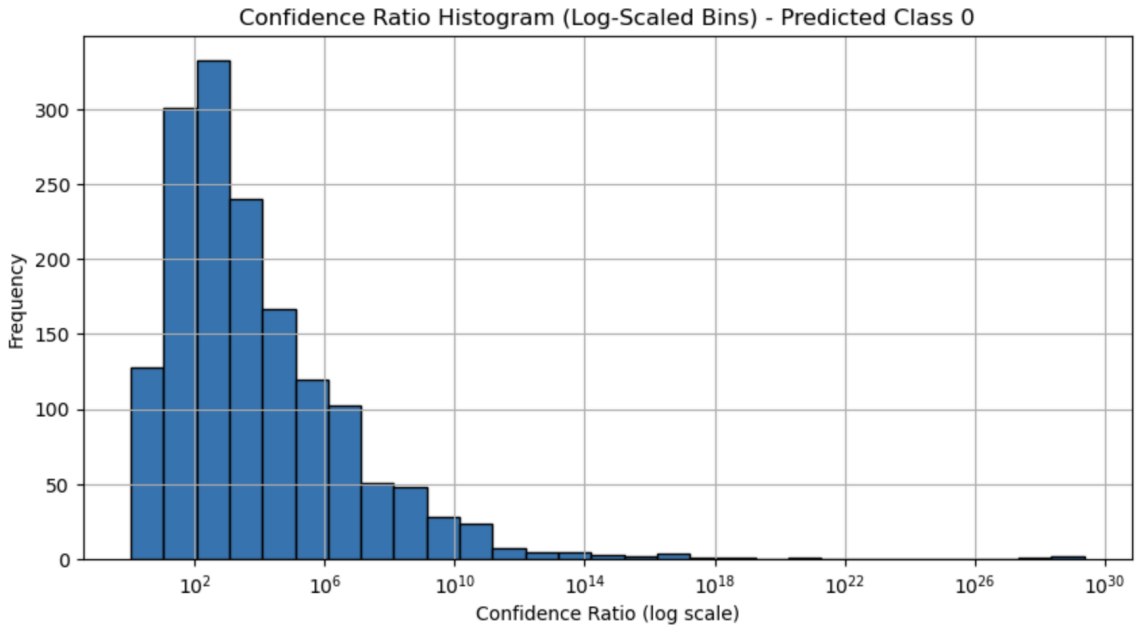

Figure 1: Confidence ratio distribution of non scam class of original model

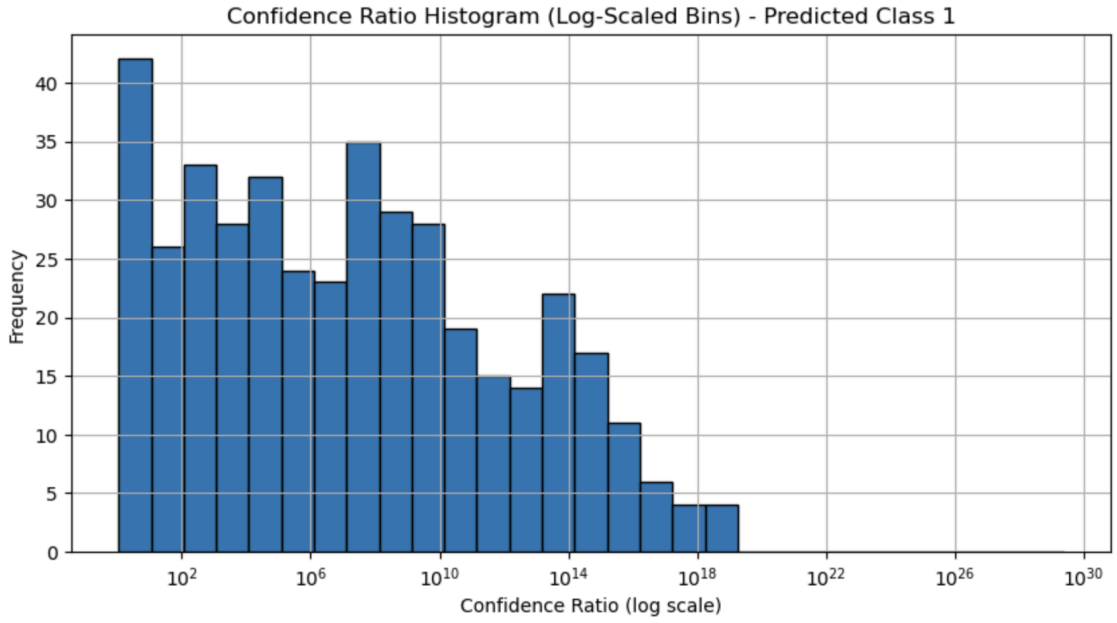

Figure 2: Confidence ratio distribution of scam class of original excluded model

The lack of improvement for the iterative approach may be because early model biases are propagated into later stages or that noisy early labels dominate even as the model is retrained.

Having said this, the performance of the low confidence excluded model suggests that there is a strong case to trade off the quantity of a dataset for its quality.

Approach	Accuracy	TP	FN	FP	TN	Precision (Scam)	Recall (Scam)	F1 (Scam)	Precision (Non-Scam)	Recall (Non-Scam)	F1 (Non-Scam)
Baseline (All pseudo-labels used)	0.9719	183	17	11	784	0.9433	0.915	0.9289	0.9788	0.9862	0.9825
Excluding low Confidence (filtered pseudo-labels)	0.9729	183	17	10	785	0.9482	0.915	0.9313	0.9788	0.9874	0.9831
Iterative (50% label → retrain → label rest)	0.9688	182	18	13	782	0.9333	0.91	0.9215	0.9775	0.9836	0.9806
Iterative (balanced scam/noscam in first 50%)	0.9698	181	19	11	784	0.9427	0.905	0.9235	0.9763	0.9862	0.9812

Table 10: Performance evaluation metrics of semi supervised models

Question 4

As mentioned above the model which had low confidence pseudo classifications removed from its training set was the best performing model out of all the tested models. The accuracy was about 97.3% which was approximately 0.3% higher than the original model in question one. This model also tested equal or better on precision and recall than other models as well. Specifically, the gains were mainly made in false positive classifications which fell from 12 to 10 which indicates a better class separation in the model's representation after training on the new pseudo-classified unlabelled data. The baseline model as described in question 2 also performed better than the original model in question 1 but to a lesser extent, classifying 28 texts incorrectly instead of the low confidence excluded model's 27. This suggests that there is a large benefit to training on more data. However, the low confidence excluded model then performed even better with fewer training instances which emphasises the performance gain that focusing on reducing noise in pseudo classifications can bring, above and beyond iterative training methods.

It is also clear that along with the increase in accuracy of the model, the confidence of the model has also increased. Looking at the distribution of confidence ratios of both the original model (**Figure 5)** and the low confidence excluded model (**Figure 6)** we can see that in the semi-supervised model the entire distribution is shifted right which can be seen in the difference in median confidence scores as seen in **Figure 3 and FIgure 4** for both scam and non malicious classes.

The smoothed likelihood ratios revealed changes in word-level importance. While high-impact scam words such as *"win," "claim,"* and *"paytm"* remained dominant, the semi-supervised model slightly reweighted non-scam indicators like *"hello," "u,"* and *"great,"* improving class separation.

Notably, some words that previously had marginal predictive power became **more strongly associated** with a class, due to their consistent use in confidently labelled unlabelled data. This indicates that the model was able to reinforce existing feature weights and slightly refine its vocabulary distribution without overfitting to noise.

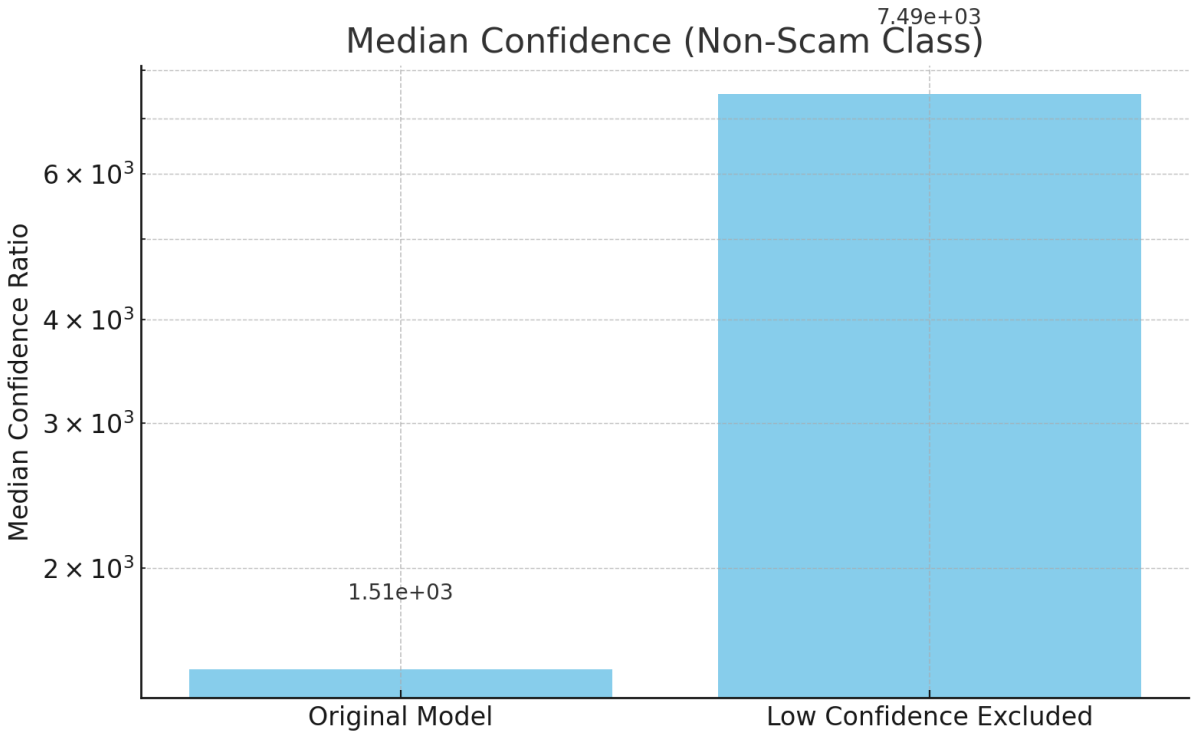

Figure 3: Median Confidence ratio of non scam class of original model vs low confidence excluded model.

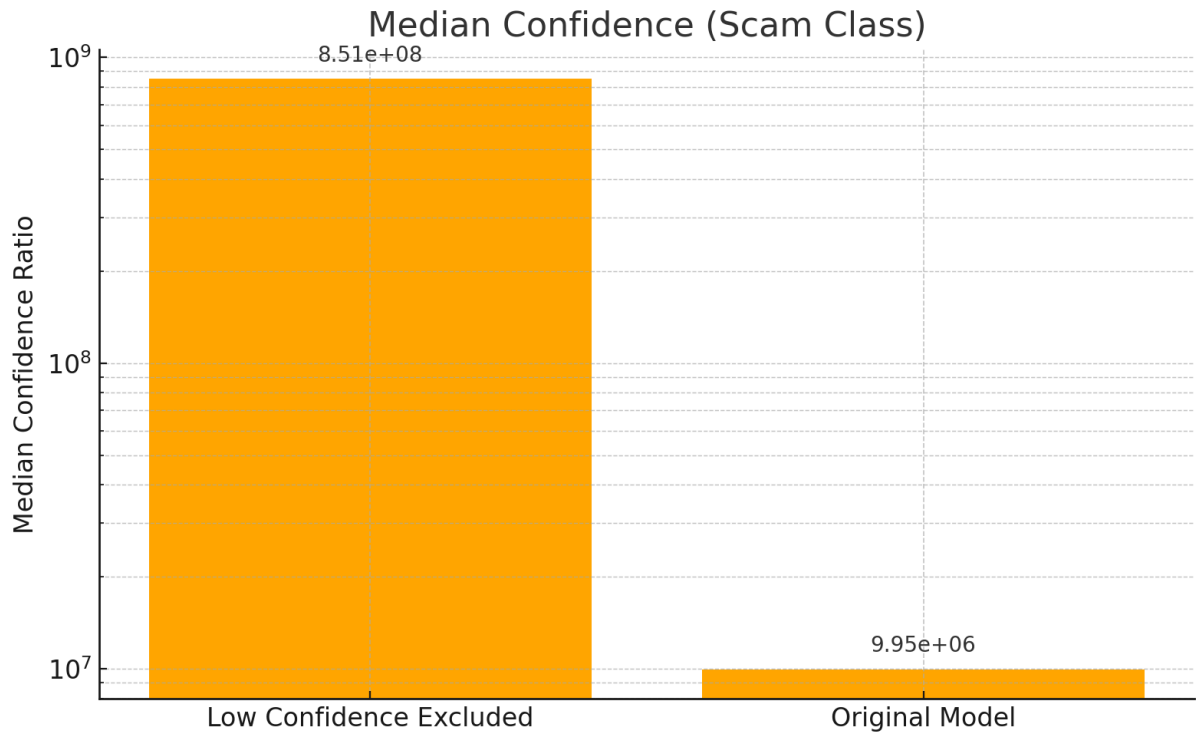

FIgure 4: Median Confidence ratio of scam class of original model vs low confidence excluded model.

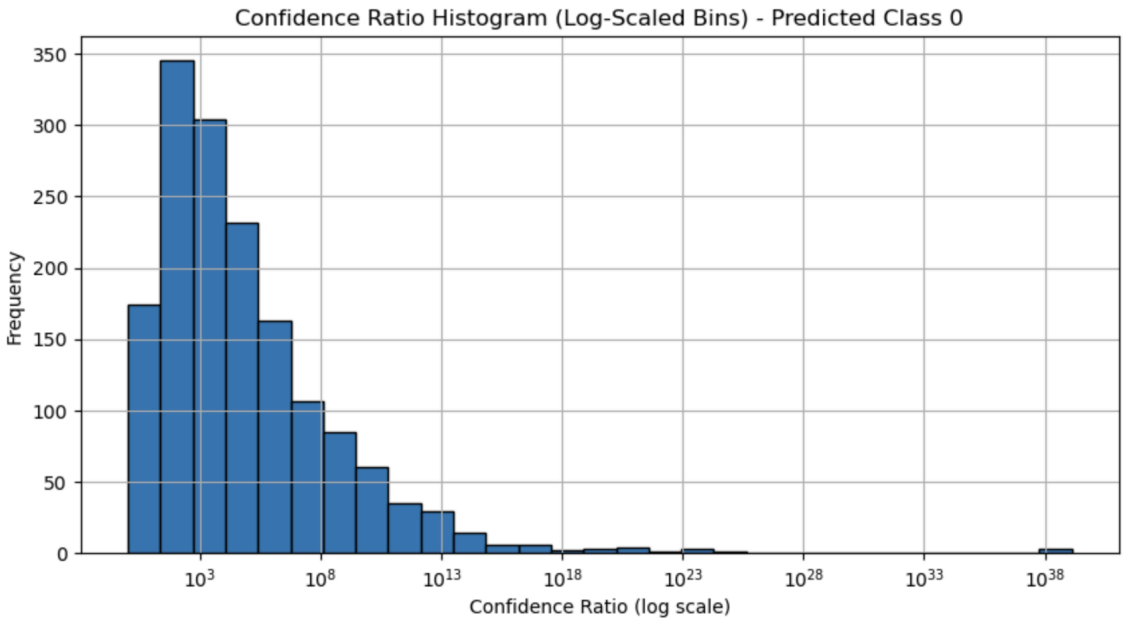

Figure 5: Confidence ratio distribution of non scam class of low confidence excluded model

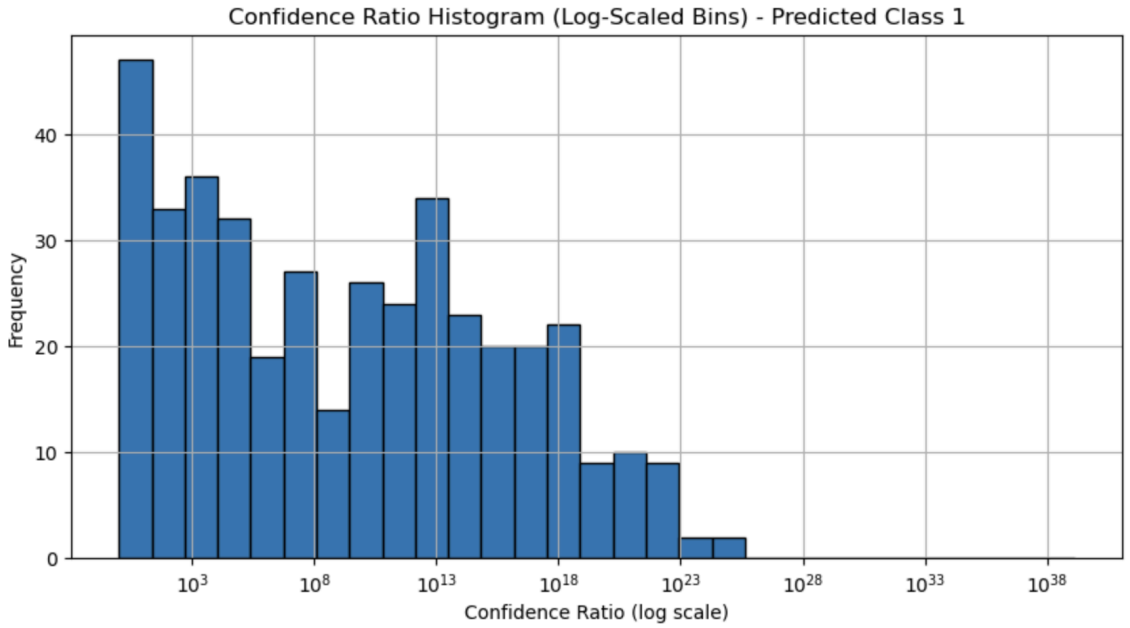

Figure 6: Confidence ratio distribution of scam class of low confidence excluded model

```
📊 Descriptive statistics for predicted_class = 0:
count    1.573000e+03
mean     2.520276e+26
std      7.162497e+27
min      1.090941e+00
25%      9.079942e+01
50%      1.510475e+03
75%      1.407084e+05
max      2.362652e+29
Name: confidence_ratio, dtype: float64

📊 Descriptive statistics for predicted_class = 1:
count    4.120000e+02
mean     3.697662e+16
std      3.598178e+17
min      1.109992e+00
25%      1.654869e+03
50%      9.953145e+06
75%      4.009930e+10
max      5.382116e+18
Name: confidence_ratio, dtype: float64
```

Figure 7: Descriptive statistics of confidence ratios of original model

```
Descriptive statistics for predicted_class = 0:
count    1.576000e+03
mean     9.540974e+35
std      3.316740e+37
min      1.014231e+00
25%      1.907052e+02
50%      7.488873e+03
75%      2.788650e+06
max      1.309165e+39
Name: confidence_ratio, dtype: float64

Descriptive statistics for predicted_class = 1:
count    4.090000e+02
mean     2.153303e+22
std      3.120242e+23
min      1.023605e+00
25%      4.342606e+03
50%      8.511239e+08
75%      2.680790e+14
max      5.891141e+24
Name: confidence_ratio, dtype: float64
```

Figure 8: Descriptive statistics of confidence ratios of low confidence excluded model