Determining a Classifier Threshold of NBC

Project 3: Ben Brobbey, Sierra Freytiz, Drew Mangione, Timothy Wilmes, Maitland Witmer

ECES 450/650 - Dr. Rosen

March 21, 2024

Contents

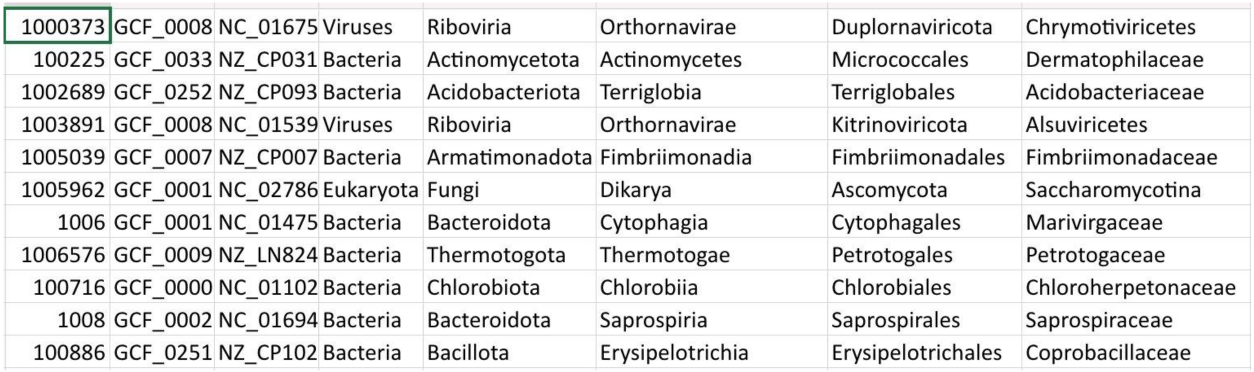
Abstract

Exploratory learning can be a useful tool in bioinformatics, especially when it comes to the classification of a lot of data within the field. Since biological data can be grouped in hierarchies to compare various organisms, this style of classification is very helpful. The problem with many classification algorithms is their iterative process, having to start from the beginning each time a new set of organisms are introduced into the database. A fix for this is creating an algorithm that can learn a set of labeled data and use the findings on future sets of data. This is where the Naïve Bayes Classifier (NBC) algorithm comes in. NBC is a supervised algorithm that is trained on given data and can be used to group future organisms into their respective taxonomic levels. The model works probabilistically, based on the likelihood an organism is close in similarity to a given taxonomic group. In this exercise, we took a database of viruses, bacteria, and archaebacteria and randomly split it three times (50/50,40/60,20/80), where the first part was used from training the NBC, while the second part was used to create two testing cases. The results from classification were then used to score how well each of the NBC cases performed. The case where the training algorithm was able to see 50% of the data had the best scores for finding organisms taxa. In each case, the classifier was accurately able to tell which sequences were more related to the most abundant organism in the training algorithm versus the sequences that were farther away in relation.

1. Materials and Methods

A comprehensive research methodology is required for the implementation of an exploratory learning algorithm for Naive Bayes Classifier (NBC) to ensure effective testing and evaluation of the algorithm's performance. A literature review to gain a deeper understanding of exploratory learning and the functioning of Naive Bayes Classifier was completed prior to the start of each task defined in the proposal. The ability to react to unanticipated classes in query or test data sets, as is the case for machine learning algorithms, is an essential feature of exploratory learning. The first key material used for our project was Picotte. Picotte is Drexel's main high-performance computer cluster. The use of Picotte improved the speed at which we would be able to run our code during the project.

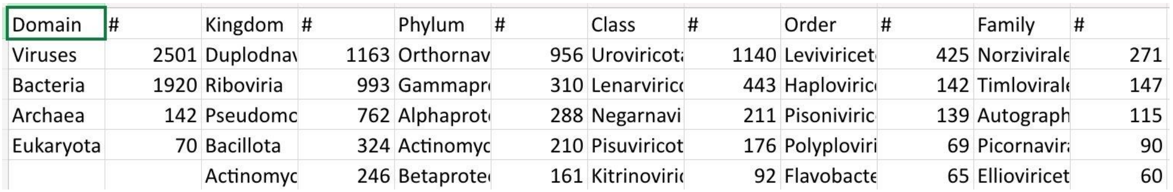
Another material used was the dataset given at the start of the project. This dataset contained various viruses, bacteria, and archaebacteria, which we needed to sort into taxonomic categories from domain to order. To accomplish this task, we first created a file that linked the arbitrary folder numbers to the taxonomic IDs. Next, we looked up each sequence in the National Center for Biotechnology Information (NCBI) database to map to its corresponding organism. The resulting data was then saved into a comma separated values file (CSV), which linked the organism to the folder name that the sequence is associated with, the file name of the sequence, and the taxonomic classification of the organism. Table 1 below shows a snippet of how the CSV file was organized.



1000373	GCF_0008	NC_01675	Viruses	Riboviria	Orthornavirae	Duplornaviricota	Chrymotiviricetes
100225	GCF_0033	NZ_CP031	Bacteria	Actinomycetota	Actinomycetes	Micrococcales	Dermatophilaceae
1002689	GCF_0252	NZ_CP093	Bacteria	Acidobacteriota	Terriglobia	Terriglobales	Acidobacteriaceae
1003891	GCF_0008	NC_01539	Viruses	Riboviria	Orthornavirae	Kitrinoviricota	Alsuviricetes
1005039	GCF_0007	NZ_CP007	Bacteria	Armatimonadota	Fimbriimonadia	Fimbriimonadales	Fimbriimonadaceae
1005962	GCF_0001	NC_02786	Eukaryota	Fungi	Dikarya	Ascomycota	Saccharomycotina
1006	GCF_0001	NC_01475	Bacteria	Bacteroidota	Cytophagia	Cytophagales	Marivirgaceae
1006576	GCF_0009	NZ_LN824	Bacteria	Thermotogota	Thermotogae	Petrotogales	Petrotogaceae
100716	GCF_0000	NC_01102	Bacteria	Chlorobiota	Chlorobiia	Chlorobiales	Chloroherpetonaceae
1008	GCF_0002	NC_01694	Bacteria	Bacteroidota	Saprospiria	Saprospirales	Saprospiraceae
100886	GCF_0251	NZ_CP102	Bacteria	Bacillota	Erysipelotrichia	Erysipelotrichales	Coprobacillaceae

Table 1: Snapshot of how each organism was sorted within our CSV file

With the resulting CSV file, we found the 5 most abundant taxa of each taxonomic group and used an excel sheet that was provided for comparison and verification. The top 5 were chosen because then at least 70 or more organisms existed for each level (except the fifth family group). A snapshot of this excel sheet is shown below in Table 2. This provided enough organisms for both training and testing purposes, resulting in 25 cases, as they were only selected from kingdom to order. Once the data was split, each case had one of the three different percentages, resulting in 75 training cases.



Domain	#	Kingdom	#	Phylum	#	Class	#	Order	#	Family	#
Viruses	2501	Duplodnav	1163	Orthornav	956	Uroviricot	1140	Leviviricet	425	Norzivirale	271
Bacteria	1920	Riboviria	993	Gammapr	310	Lenarviric	443	Haploviric	142	Timlovirale	147
Archaea	142	Pseudomc	762	Alphaprot	288	Negarnavi	211	Pisoniviric	139	Autograph	115
Eukaryota	70	Bacillota	324	Actinomyc	210	Pisuviricot	176	Polyploviri	69	Picornavir	90
		Actinomyc	246	Betaprote	161	Kitrinoviri	92	Flavobacte	65	Ellioviricet	60

Table 2: The 5 most abundant groups within each taxonomy level.

Secondly, training and testing groups were created using python, Figure 1, of which a group for each one of the 5 most abundant names within each taxa was developed. The training contained at least half of all the most abundant cases. Then testing cases were selected based on how closely related it was to its selected grouping. If the query sequence selected was within the same group as the overall grouping it was considered near. If the query sequence was in the same group one taxonomic level above, then it was considered far. For example, kingdom 2 (k2) is

Riboviria, so that becomes the main organism focused on in k2. A sequence that is classified as Riboviria is considered near. A sequence that is not Riboviria but is still a Virus is considered far. This allowed for a mixture of data that was not put in for training.

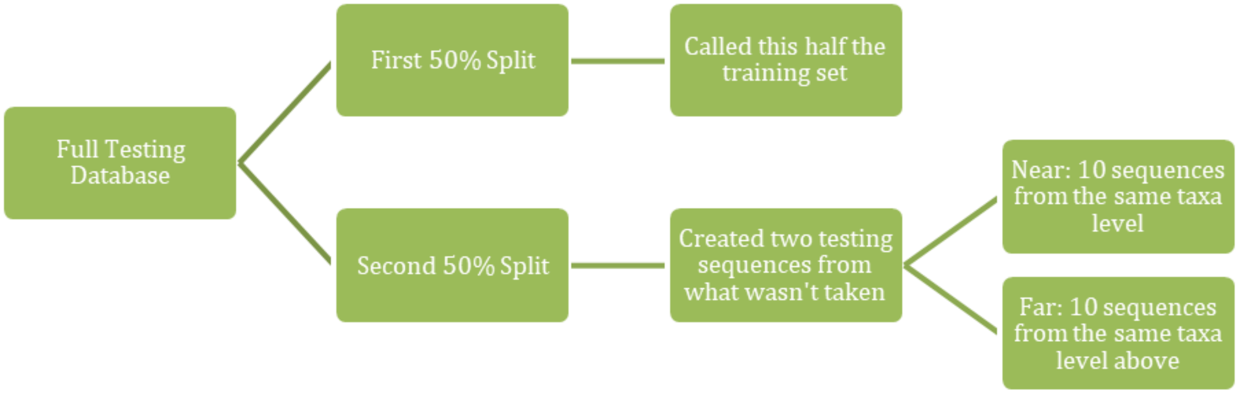

Full Testing Database
First 50% Split
Called this half the training set
Second 50% Split
Created two testing sequences from what wasn't taken
Near: 10 sequences from the same taxa level
Far: 10 sequences from the same taxa level above

Figure 1: A flow chart on the breakdown of our provided data set

The full database is split first 50/50. Then the full dataset is split two other ways 40/60 and 20/80, where the first is what goes to training and the rest to testing. After splitting the data, a shell script, Jellyfish, ran over the data. Jellyfish reads a FASTA file that contains DNA sequences and generates the k-mers counts. The k that was chosen for this project was 3. The jellyfish output the 3-mer counts for each sequence. Using the outputs from jellyfish, the model was trained on the 3-mer counts for the sequences. After training the model, two testing sets were run with the trained model and to evaluate the model. One of the testing sets was a near set; the other set was a far set. Near and far sets were defined in the paragraph above. Each near and far set contained 10 different sequences chosen randomly.

After 150 trials (75 near, 75 far) had been completed, the resulting data comprised of 150 separate CSV files. Each CSV file contains the logarithmic probability of a match for each read that was in the test sequence file, so each file can contain hundreds of values. To simplify the data processing, each CSV file was run through a Python script that took the average probability value and saved it to a new CSV file that contained the average of every CSV file, 150 averages in total. The values from this new CSV file would be the basis that all results would be calculated off. A threshold was then established based on the results, and ROC curves were produced.

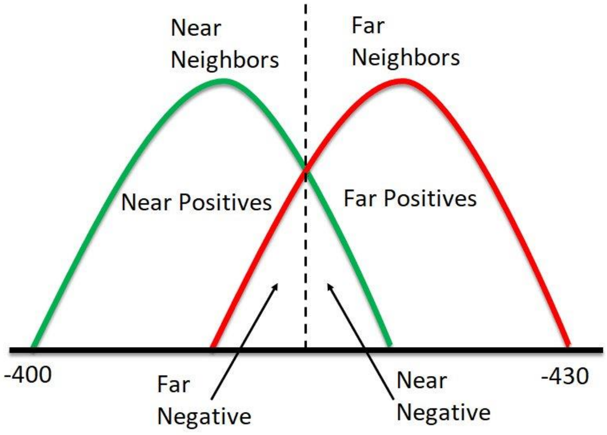


Figure 2: Graphic displaying how the classification of the results was categorized

The image above, Figure 2, shows how our results were put together into a ROC curve. The dashed line indicated the selected threshold, the red curve is our far neighbors, the green curve is the near neighbors. The result is considered positive if it fell to the proper side of the threshold line based on what we had grouped it as. For example, if we put a sequence in our far neighbor classification set and was scoring as a far neighbor, it was positive. If the result fell to the other side of the threshold line, it was considered negative. Another example would be if we put a sequence in our far neighbor classification set and was scoring as a near neighbor, it was negative.

2. Results

After importing the total averages of each trial, various statistics could be calculated for the data. First, the average for categories of data were calculated, shown below in Table 3.

Label	Average

Near	-413.3939731
Far	-415.6971804
20%	-415.0702146
40%	-414.3867739
50%	-414.1797418
Family (All)	-420.2797967
Order (All)	-414.5342288
Class (All)	-417.9020213
Phylum (All)	-409.0898847
Kingdom (All)	-410.9219522
Family (50%)	-419.981754
Order (50%)	-414.144546
Class (50%)	-417.4398853
Phylum (50%)	-408.739904
Kingdom (50%)	-410.5926197

Table 31: Averages from our classified outputs

There are some interesting trends to look at with the values in Table 3. Firstly, the near trials on average have a value 2.30320727 larger than the far trials. While the difference between the two is slight, it shows that NBC can detect the difference between near and far neighbors. Another trend is that the average probability value increases as the trial training dataset increases (20% to 40% to 50%), showing that a larger training dataset results in better probabilities, on average. Lastly, the average probability increases as the taxonomic level of the training dataset gets less specific (Family to Order to Class to Phylum to Kingdom). This could be because the datasets becoming larger due to more organisms belonging to larger taxonomic levels. Another reason could be because more general taxonomic levels have a wider range of organisms, which would increase the likelihood that a test read would be close to the training data.

To calculate the ROC curve for the data, a threshold had to be established first. The initial threshold used was the median between the near and far average values (-414.5455768). To calculate a more accurate threshold, however, a method was used where the Cumulative False Positive Rate and Cumulative True Positive Rate were combined. The threshold was then iterated through to find which threshold resulted in the highest overall number. The final threshold used was -407.1. MATLAB code was also used to calculate the threshold and resulted in a very similar threshold. Using the threshold, the following calculations were done below in Table 4:

Bins	Correctly Predicted		Cumulative		False Positive Rate	True Positive Rate	AUC
	Positive	Negative	Positive	Negative			
			0	0	1	1	0.068966
> -403	6	0	6	0	0.931034483	1	0.034483
-403 - -405	3	1	9	1	0.896551724	0.984126984	0.079183
-405 - -407	7	4	16	5	0.816091954	0.920634921	0.095238
-407- -409	9	6	25	11	0.712643678	0.825396825	0.066411
-409 - -411	7	5	32	16	0.632183908	0.746031746	0.05145
-411 - -413	6	5	38	21	0.563218391	0.666666667	0.061303
-413 - -415	8	6	46	27	0.471264368	0.571428571	0.065681
-415 - -417	10	6	56	33	0.356321839	0.476190476	0.021894
-417 - -419	4	6	60	39	0.310344828	0.380952381	0.039409
-419 - -421	9	15	69	54	0.206896552	0.142857143	0.019704
-421 - -423	12	0	81	54	0.068965517	0.142857143	0.009852
< -423	6	9	87	63	0	0	0

Table 42: Calculations used for ROC and AUC

Bins of 2 were picked as it resulted in 12 bins between -403 and -423. The number of correctly predicted averages was then totaled. Correctly predicted refers to the number of near/far trials that had averages above/below the threshold, respectively. This allows for a false positive and true positive rate to be calculated, which in turn allows for the plotting of a ROC curve. The AUC was also calculated, and the value is displayed at the bottom of each respective graph. An AUC value of above 0.5 means that the threshold resulted in a positive correlation for predicting the near/far status of a given testing dataset. Below is the ROC curve for all 150 trials, Figure 3.

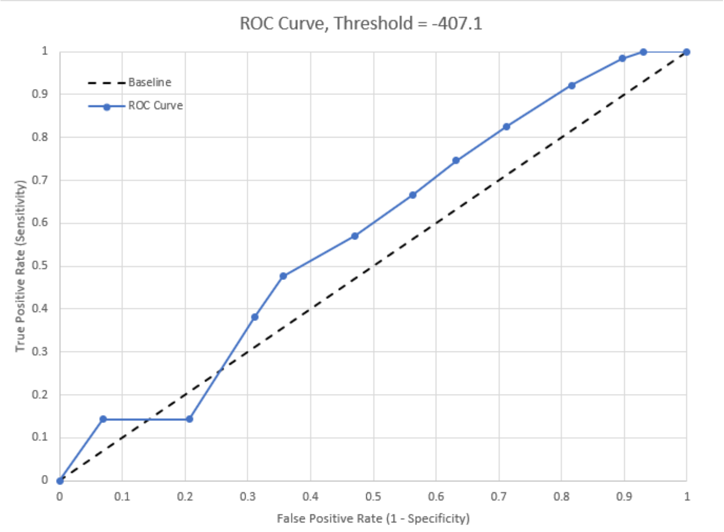


Figure 31: ROC Curve All, Threshold = -407.1, AUC = 0.6135742

Looking at the ROC curve, it is obvious that there is a slightly positive correlation for predicting the near/far status of a given testing dataset. The negative section at the bottom left of the graph is due to a lack of data in the -419 to -423 range, with enough data that should go away. Below are the ROC curves for the 50%, 40%, and 20% trials, respectively Figures 4, 5, and 6.

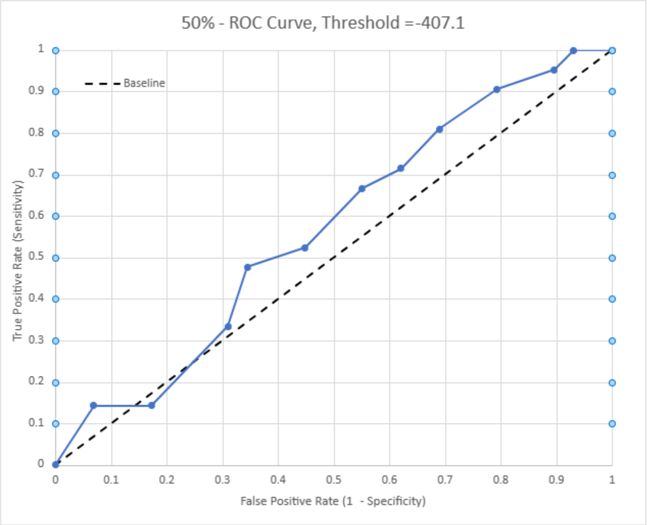


Figure 42: 50% ROC Curve, Threshold = -407.1, AUC = 0.6108374

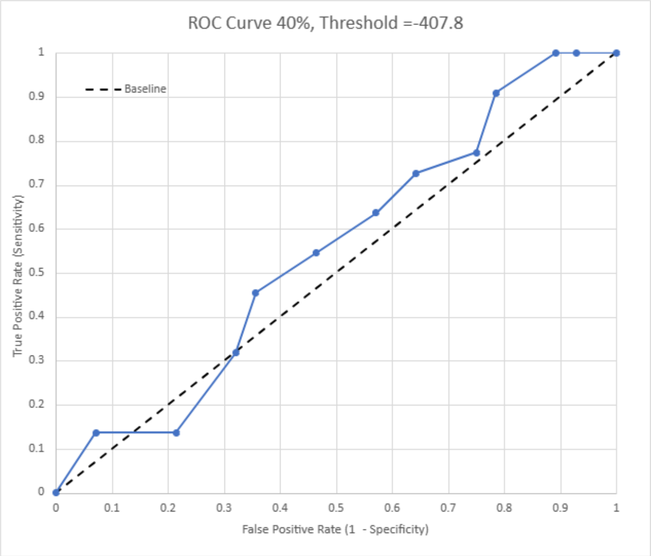


Figure 53: 40% ROC Curve, Threshold = -407.8, AUC = 0.5876623

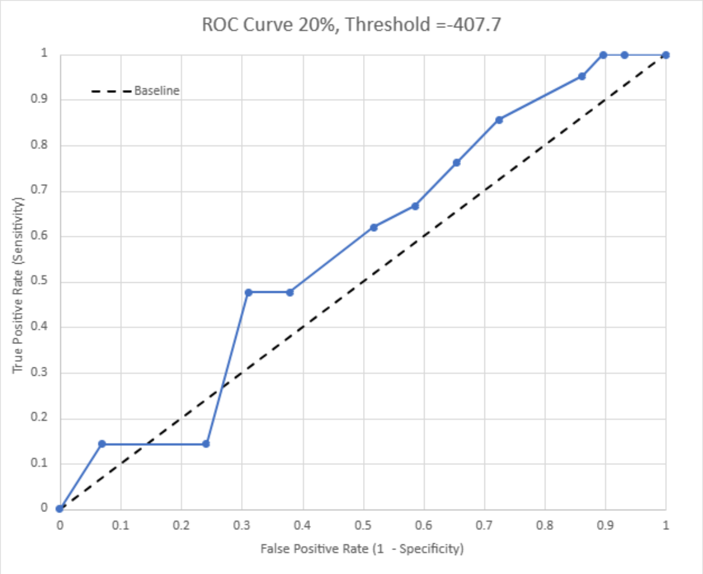


Figure 64: 20% ROC Curve, Threshold = -407.7, AUC = 0.612479

Looking at the ROC curves, it appears that there isn’t much of a pattern between the 50%, 40%, and 20% trials. The overall data curve is much cleaner than any of the individual curves, and this makes sense as it has the largest dataset.

3. Discussion

Based on the results, it is evident that when more training data is available to the machine learning algorithm, the classification of unknown organisms tends to be better. This means that in the future, when a new species appears, we will be able to classify it at lower taxonomy levels. It is also important to note that Naïve Bayes Classifier performed well in identifying closely related organisms even though it had not seen it beforehand. If more trials of testing data had been evaluated with the model, the shape of ROC curve would have been more defined. More trials could not be completed due to the time it took to run each file needed.

One of the limitations of this project is our research was done specifically on viruses and bacteria. While the dataset was vast, there are numerous other organisms outside of these domains. Another limitation is knowing what exactly the classifier wants to identify a sequence as. The output gives us a score on how close it thinks it is, without telling us it classified the organism into in a specific level.

Work for the future that could be done is changing the type of data that is seen by the classifier, such as all eukaryotic organisms to understand if classification comes as close as the current data. Expanding the training dataset to include a broader range of organisms from different taxonomic groups would provide the classifier with more diversity. Another improvement could be adding the lowest level of taxa the algorithm picks out for each query sequence. The size of the k-mers generated by Jellyfish at the beginning of the process could also be altered, to see if k-mer size has any effect on the final ROC curves.

Overall, this exercise with NBC was successful in understanding if it could identify if an unknown organism was related to previously seen organisms. While further experimentation is needed to fully understand the value, the model currently works adequately. Similar to other models, there are limitations, but they should not discourage from the effectiveness of NBC.

Therefore, with the end of this project, we have found NBC a useful model that should be further explored.