Naïve-Bayes Classifier Analysis:

Naïve-Bayes Classifier is a method used in Machine Learning with supervised learning. The way Naïve-Bayes Classifier works is that we use labeled data to train the classifier. In this assignment, we have a dataset of comments with the label negative or positive. Theoretically, we would train the classifier with vocabulary and labels. The algorithm would classify the appearance of certain vocabularies based on the label to generate a conditional probability. And use this probability to determine the label of any dataset given in the future. Since Naïve-Bayes Classifier is based on probabilities, the results can not be 100% accurate.

From our python code, we have generated a text file listing the label and corresponding line number. By comparing the original file and the generated file, we find out that the number of negative labels from the original file is 1282, and for positive label is 1101; number of negative labels from the generated file is1239, and for positive label is 1153.

We can see that Naïve-Bayes Classification is not 100% accurate. In case of a label difference between the original file and the generated file, the cause might be that some appearance of certain words in the instance make the classifier think that it belongs to the opposite label. For example, line #10 state: *“i agree with other reviewers that it feels good and does n't smell too much, however , i 've experimented with it several times to confirm my findings , and it turns out to give me really bad blackheads . i 'm 25 with an oily t-zone and very dry facial skin . on mornings after using this cream, i have nasty blackheads on my forehead and chin . there are better products out there”.* By reading the whole instance we know that this is a negative review, but Naïve-Bayes classifier only look at each word in this instance, here we have the appearance of the word *“good”.* The word good should appear in a lot of positive reviews so that it makes the classifier think that this is a positive review rather than a negative review. Another example, one the line #880, it states: *“i am having the same problem . i ordered this product from amazon on august 26. it is now october 26. i have not recieved it yet . i have never had this problem with amazon before . i may have to cancel my order too and go somewhere else . it is a shame , because amazon does have the best price i have found so far . i gave it 5 stars only because i had to give it a rating. i wanted to be fair to the company , but i have honestly never used one”*. We can see that the reviewer was unhappy about the delivery, but he gives a good review of the product. The reason that the classifier senses a negative label for the review is that there are vocabularies such as “problem”, “shame”, make the classifier decide that this is a negative review.

Mathematically, the general formula for Naïve-Bayes Classifier is:



To avoid underflow, we use summation of log instead of product of conditional probability. For this assignment, we have, at first, calculated the probability of the positive label as well as the probability of the negative label with the training set. The conditional probability of separate vocabularies will be defined for the specific label. So when the classifier starts doing its job, it would use the above formula to calculate the summation of all the conditional probability of each word(vocabulary) to determine the label.

Base D-T:

The Decision Tree struggles with some misclassification, most of which could be justified. One example of such misclassification is whenever the user uses a tone that does not match with the review he gave. For example in line 9840 ( “ software pos 713.txt it would be nice if the contents were listed , with all of microsoft 's versions , it gets confusing ” ), even though the review was positive, the tone that the user used seemed more of a complaint, hence why the model classified it as negative instead of positive.

Another contributor to the model’s misclassification of documents is the labels provided for the documents may be incorrect. For example, Line 11579 says “ i was very happy with my purchase . the company chose to mail it to me via priority mail in order for it to arrive before christmas . i was very pleased ! ! ! ! ” which is clearly a positive review, however it was labeled as a negative review in the data set.

Furthermore, in many cases the model tends to incorrectly classify neutral sounding reviews as positive. For instance in line 11,416 the review is as follows “ books neg 455.txt i purchased this took and then found that it was just a reprint from several decades ag “. The model classified this review as positive. This is an expected behavior, since if we take a look at the confusion matrix, we can clearly see that the model misclassified negative documents as positive, way more than it misclassified positive documents as negative. This is a normal behavior since the training data provided for the model consisted of 4,847 positive documents as opposed to 4,684 negative documents. Since the model learns from these training documents, an imbalance in documents would result in a biased model.

Best D-T

The performance of the Best Decision Tree is somewhat similar to the Base decision Tree, except it has a slight improvement on the negative recall. Just like the base decision tree algorithm, the best decision tree algorithm struggles with some misclassifications as well.

One issue that the tree struggles with is that it is sensitive to certain keywords. For instance, strongly positive words like “good” in a negative review are highly likely to cause a misclassification. One example is line 1175 : *“ camera neg 984.txt this is a good charger - quick and portable . but the battery life of the battery sucks ! ”*  The algorithm misclassified the review as a positive rather than negative because of the word “good”. To verify this claim, we can substitute the word good by the word “okay” and rerun the algorithm. With the word “good” substituted by the word “okay”, the machine correctly classifies the above mentioned review as a negative one, implying that the word good has a significant effect in the decision making of the machine.

Another example of misclassification can be seen online 11456, which states: *“this product does not make you hair grow at all . im so freakin mad it is such a lie it makes your hair grow..........im soooooooooooo madddddddddddddddddddddddddddddddddddddd*”. This instance is classified as pos by the algorithm, but in reality it has a negative label. The reason why this happens, by assumption, is that there exists only one instance in the training set with the vocabulary *“soooooooooooo”*, which is classified as positive.Hence it is likely that the algorithm considers the word “soooooooooooo” as a big discriminator and classifies all other instances containing that vocabulary as a positive instance.

By comparing the confusion matrices of all 3 algorithms, we found Naive-Bayes to be the most reliable one for this dataset. Out of all 3 algorithms, Naive-bayes had the lowest number of misclassifications, giving it the highest accuracy among the 3 algorithms, while also maintaining a higher recall and precision when compared to the Base Decision Tree and Best decision Tree.