Text Processing

Sharath Devanand - ACP23SD

Statistical Approach

The "readfiles" function processes data from Rotten Tomatoes and Nokia files, then splits it into 80-20 training and test sets with the help of the random Python packages. A dictionary is generated cataloging word sentiment as either ‘positive’ or ‘negative’ in both datasets.

The "trainBayes" function computes the conditional probability of word sentiment by computing the averages of word frequency in positive and negative sentences.

In the “testBayes” function, the probability of a sentence being positive given the words is computed and used for predictions. (Appendix - 1.1)

Naive Bayes	**Training Data**	**Test Data**	**Nokia - All Data**
Accuracy	0.890	0.772	0.586
Precision-Pos	0.894	0.778	0.783
Precision-Neg	0.885	0.765	0.386
Recall - Pos	0.883	0.772	0.564
Recall - Neg	0.896	0.771	0.637
F1 Score - Pos	0.889	0.775	0.656
F1 Score - Neg	0.891	0.768	0.481

[Step 3.2]From the above table, the highest performance based on all the metrics is observed to be of the training data (88%-90%) of the Movies dataset. This is followed by the test data (76-78%) of the Movies dataset and the lowest performance by the Nokia dataset (38% - 78%).

One notable observation is the high positive precision in the Nokia dataset compared to the negative precision indicating an imbalance between the two sentiment classes.

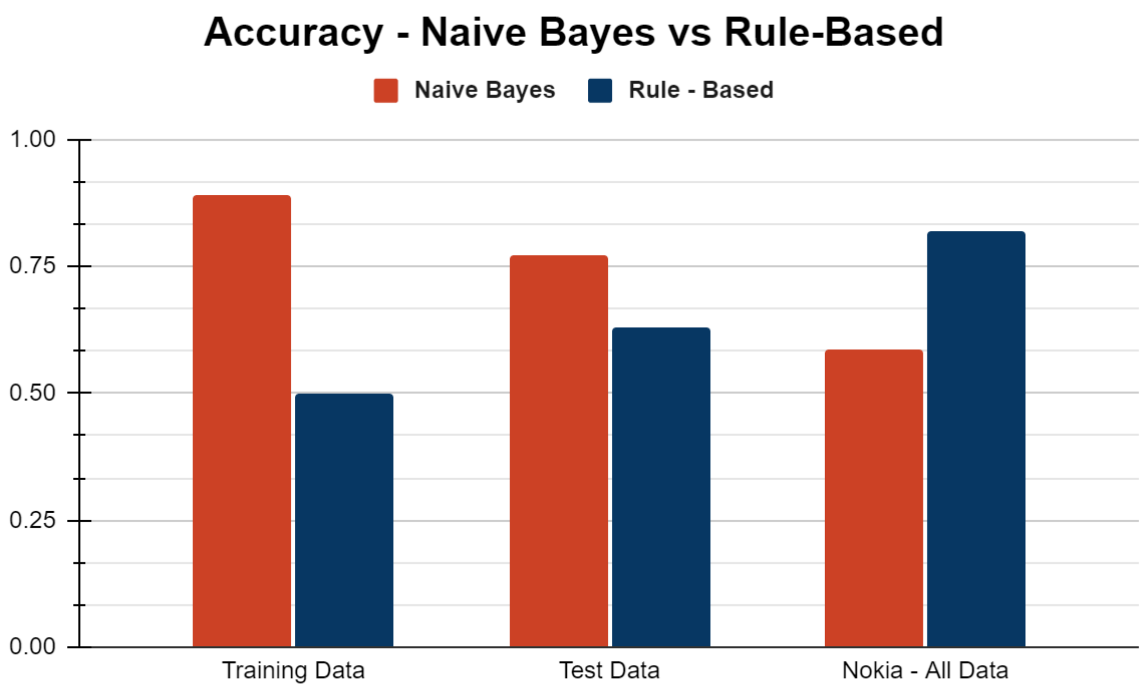
The reason for the contrasting performance metrics across datasets is 3-fold. Firstly, the model could be overfitted with the training data with the hyperparameters tuned to the 'Movies' dataset, making the model struggle to generalise. Secondly, a small training set, as in this case, decreases the sparsity of the feature space, i.e. number of words. Finally, the assumption of word independence in the probabilistic model is impractical to be applied to datasets.

Applying cross-validation and regularization would stop the model from overfitting. A larger dataset could also help the model in increasing the feature space. The threshold tuning in the testing phase allows the calibration of the model's performance by tuning a balance between precision and recall.

Rule-Based Approach

The Rule-based approach involves predicting the sentiment of the sentence by comparing the sum of sentiments in a word with a threshold.

	Training Data	**Test Data**	**Nokia - All Data**
Accuracy	0.501	0.631	0.819
Precision-Pos	0.500	0.600	0.816
Precision-Neg	0.822	0.700	0.833
Recall - Pos	0.998	0.820	0.956
Recall - Neg	0.007	0.434	0.5
F1 Score - Pos	0.666	0.693	0.881
F1 Score - Neg	0.015	0.536	0.625



[Step 5.2] It can be observed that the rule-based model performs poorly compared to the Naive Bayes approach in the 'Movies' dataset whereas it outperforms in the 'Nokia' dataset. As the Naive Bayes approach is trained on the Movies Dataset, it performs well in the testing of the same context. In the rule-based approach, it can be concluded that the complexity of the rules is over-generalized than the Bayesian approach and hence performs better on unseen data. (Appendix 1.2)

Algorithm Improvements

[Step 5.3] To improve the rule-based approach in sentiment analysis, 4 improvements are suggested

1. ***Affin (Python package)***- Affin performs sentiment analysis on a wordlist-based approach. The sentiment of every word is defined in the lexicon aiding in computing the overall sentiment of the sentence. Affin's score method also accumulates and provides a sentiment score based on a complete sentence.

	Training Data	**Test Data**	**Nokia - All Data**
Accuracy	0.512	0.531	0.439
Precision-Pos	0.505	0.523	0.974

Precision-Neg	0.782	0.8	0.348
Recall - Pos	0.989	0.988	0.204
Recall - Neg	0.038	0.046	0.987
F1 Score - Pos	0.679	0.684	0.337
F1 Score - Neg	0.073	0.088	0.514

2. ***Textblob (Python Package)*** - TextBlob is a Python text-processing package with an object-oriented approach. The polarity sub-method in the sentiment methods of the Textblob object provides the sentiment value of the word(s). The lexicon in this Python package is extracted from the corpus of NLTK.

	Training Data	**Test Data**	**Nokia - All Data**
Accuracy	0.610	0.642	0.770
Precision-Pos	0.574	0.607	0.777
Precision-Neg	0.715	0.736	0.731
Recall - Pos	0.850	0.862	0.940
Recall - Neg	0.373	0.409	0.375
F1 Score - Pos	0.685	0.712	0.851
F1 Score - Neg	0.490	0.526	0.495

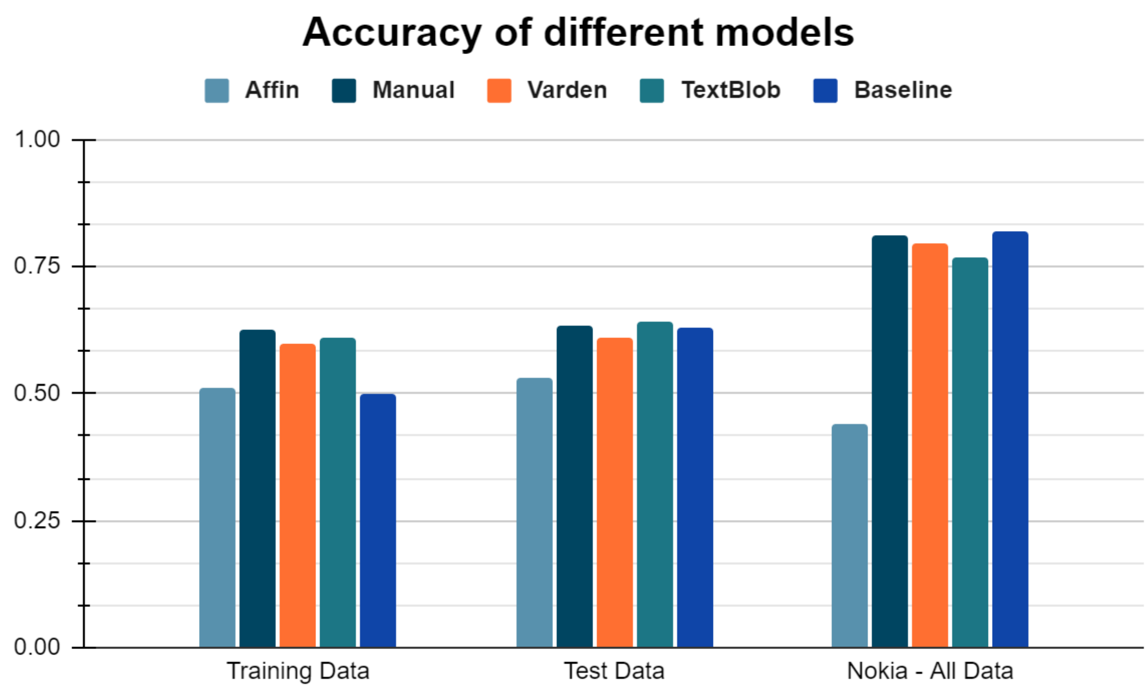
3. ***Vader (Lexicon Docment) -*** Vader is a Python package that computes the score for positive, negative, and neutral sentiment using the Vader lexicon. The lexicon from the repository is considered to perform rule-based analysis.

	Training Data	**Test Data**	**Nokia - All Data**
Accuracy	0.598	0.612	0.796
Precision-Pos	0.567	0.585	0.794

Precision-Neg	0.682	0.685	0.809
Recall - Pos	0.824	0.840	0.956
Recall - Neg	0.375	0.370	0.425
F1 Score - Pos	0.671	0.690	0.868
F1 Score - Neg	0.484	0.481	0.557

4. ***Manual Lexicon*** - Additional rules have been added such as considering negation terms (A list of negation words is used to switch the sign of the score), exclamation marks, intensifiers, and diminishers (Dictionary with scores).

	Training Data	**Test Data**	**Nokia - All Data**
Accuracy	0.627	0.633	0.812
Precision-Pos	0.593	0.609	0.817
Precision-Neg	0.700	0.685	0.788
Recall - Pos	0.805	0.803	0.950
Recall - Neg	0.451	0.454	0.512
F1 Score - Pos	0.683	0.693	0.875
F1 Score - Neg	0.549	0.546	0.621



Accounting for the accuracy metric, it can be observed that Affin has scored sub-par compared to the baseline model. Even in the case of the Nokia dataset, the scores stayed consistent in the 50% range indicating that the scoring in the Afinn lexicon is over-generalized.

The accuracy of using Textblob has indicated a small improvement from the baseline model. This is due to the massive lexicon sourced from the NLTK Python library. In the case of Varden, where the lexicon is used over the Python package, it projects a clear increase in performance throughout the datasets. This is indicative of considering contextual information in the dataset.

The manual lexicon is also shown to have a significant increase in accuracy proving the increase in the complexity of rules by considering negation and intensifiers facilitates better performance.

Another important trend to be noted is the contrasting higher performance of all rule-based algorithms in the Nokia dataset over the Movies dataset compared to the Naive Bayesian approach. (Appendix 1.3)

Useful Words

[Step 4.2] The most useful function ranks words based on the probability of the word being positive over the negative (predictive power). It can be inferred from printing the positive and negative 100 terms that after a threshold (35) the words are incorrectly quantified under positive or negative. (Appendix 1.4)

A simple return function of the head and the tail and iterating it in Sentiment Dictionary indicates that a majority of the words are not present. This is because the probability of the word comes from the list of words across the training data sentences.

Algorithm Inefficiency

[Step 6.4] The errors in the Nokia dataset are printed (Appendix 1.5) in the test Bayes and test dictionary function considering the least accuracy. The following observations can be drawn

1. A large number of sentences with contrasting sentiments containing conjunctions ('but',' although') invert the sentiment of the sentence. As the sentences are broken down into independent words, the ability to capture the sentiment has decreased.
2. The model is unable to process the context of the sentence. Even when the sentence indicates a negative sentiment due to a feature, it is considered to be positive of the overall subject.
3. Subjectivity issues can be noticed where the sentiment is not projected on the required subject. It is vital to identify the subject and then its corresponding sentiments in the sentence structure.
4. Complex sentence structure due to the addition of multiple phrases could is leading to erroneous predictions. The model is not adaptable to predict the sentiment of a larger culmination of word sentiments.
5. Negation, intensity of adjectives, and exclamation points have been observed to take a major percentage in the list of errors. Adding complexity to the rules in the test dictionary function considering these factors would improve the model.

Appendix

1.1 - Probabilistic model

The statistical approach involves the model being built with the help of the Naive Bayes Theorem which is given by,

$$P(Positive|word) \ = \ \frac{P(Word|Positive) * P(Positive)}{P(Word)}$$

Each of the conditional probabilities of a word being positive or negative is assumed to be independent and hence given by,

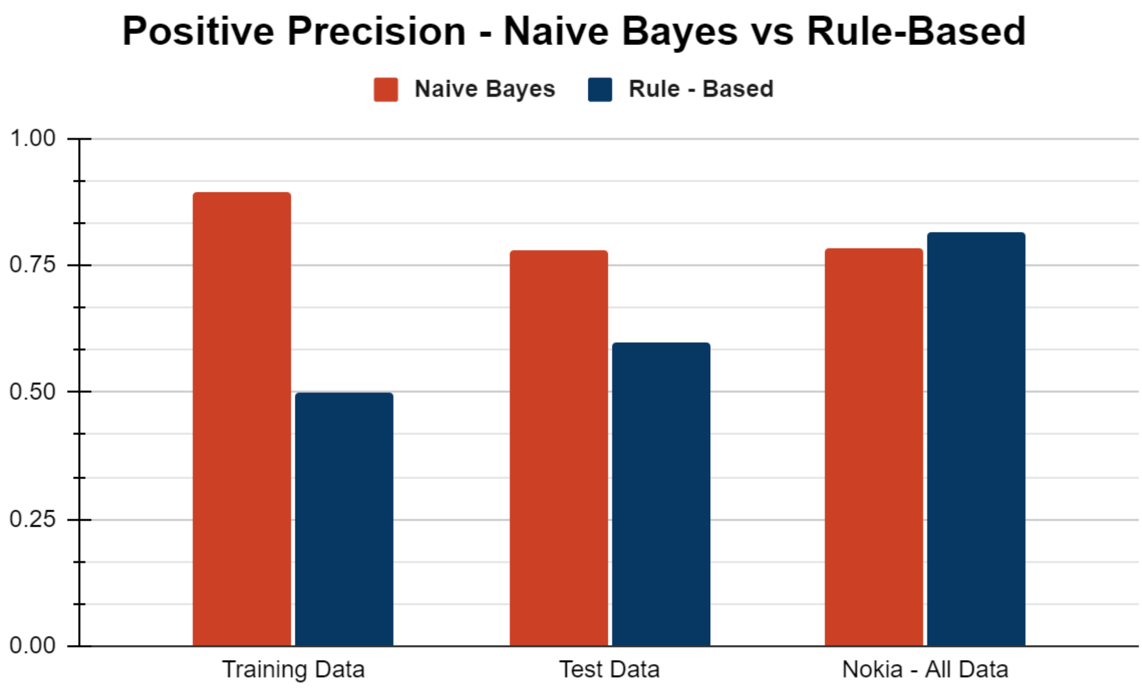
$$P_{positive}(Word|Positive) \ = \ \prod_{i=0}^{N} P(Word_i|Positive)$$

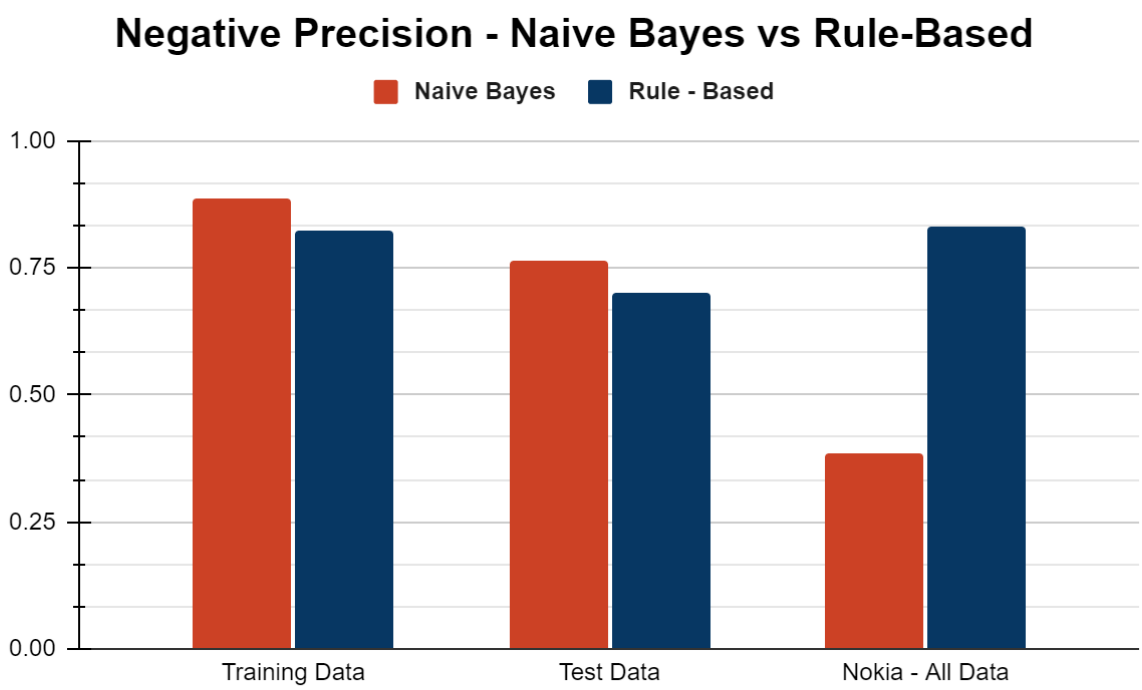
$$P_{negative}(Word|Negative) \ = \ \prod_{i=0}^{N} P(Word_i|Negative)$$

The probability of a word is broken down into the sum of mutually exclusive conditional probabilities of the word given positive and negative. The modified naive Bayes equation is given by,

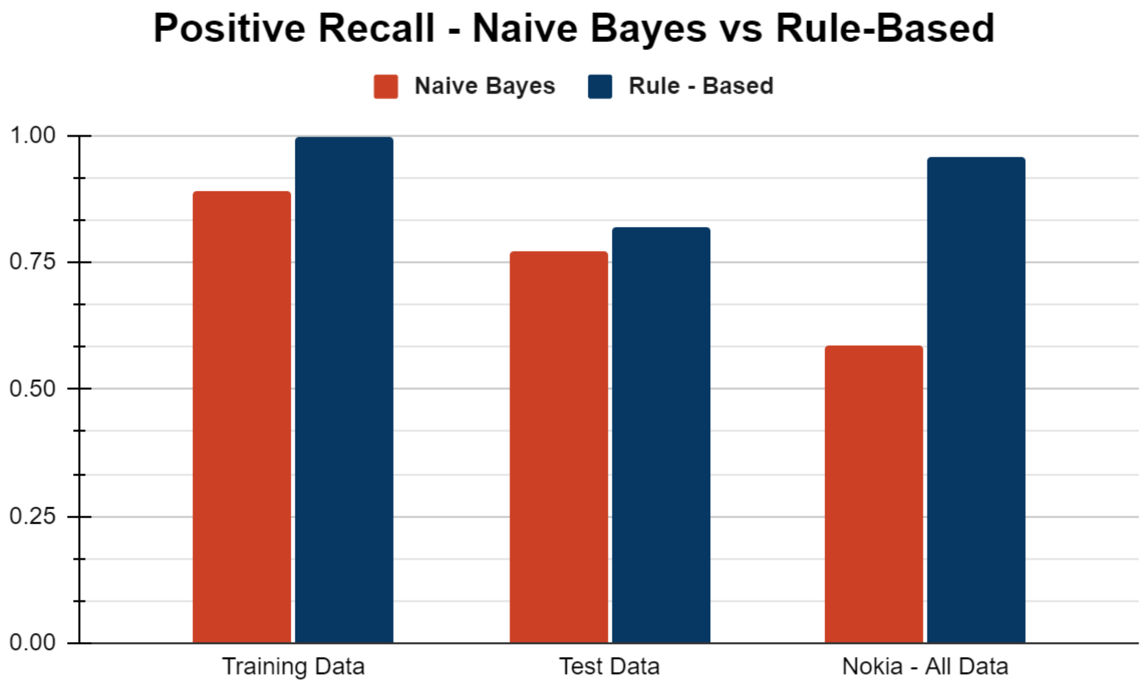
$$P(Positive|word) \ = \ \frac{P_{positive}(Word|Positive)}{P_{positive}(Word|Positive) + P_{negative}(Word|Negative)}$$

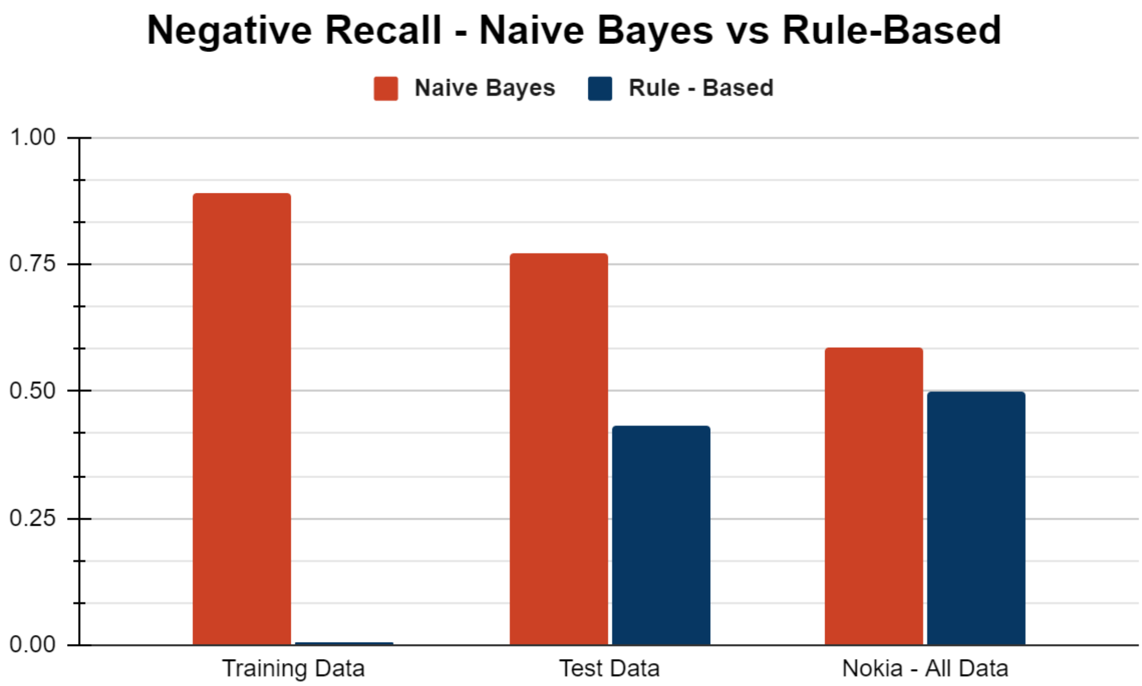
1.2 - Naive Bayes vs Rule-Based

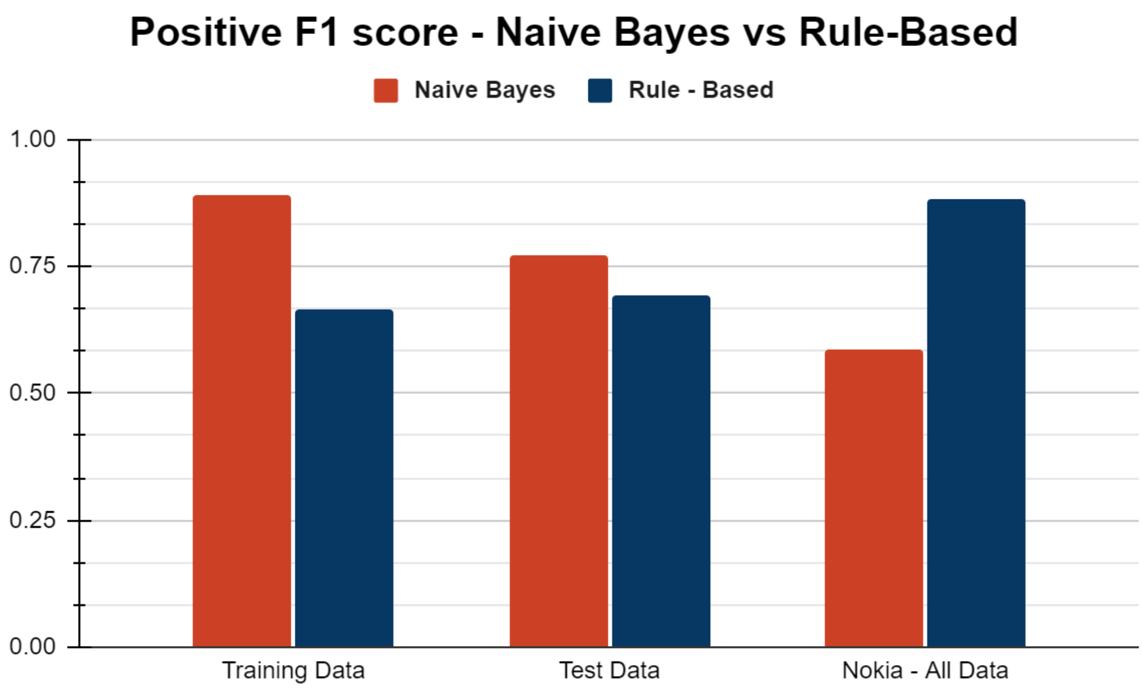


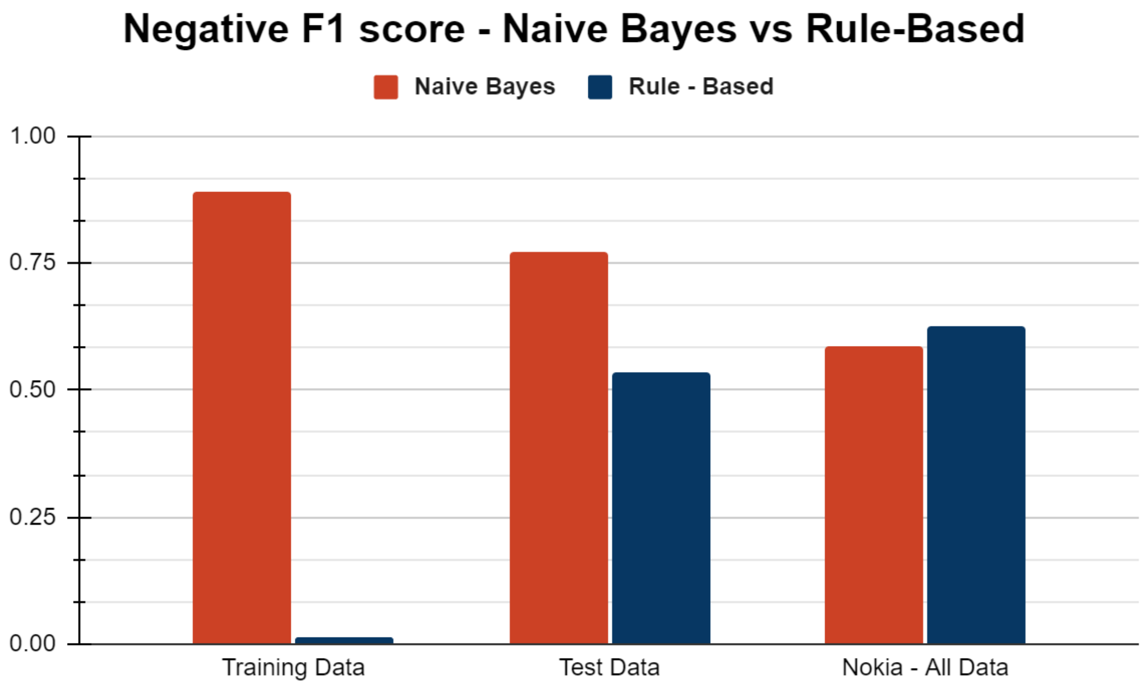


It can be observed that the Naive Bayes outperforms the Rule-based classifier in the ‘Movies’ data in both positive and negative precision. This is due to the classifier being trained on the same dataset. In the case of unseen data, such as the Nokia dataset, the Rule-based classifier performs optimally compared to the statistical counterpart with the ability to generalize apart from the training data.



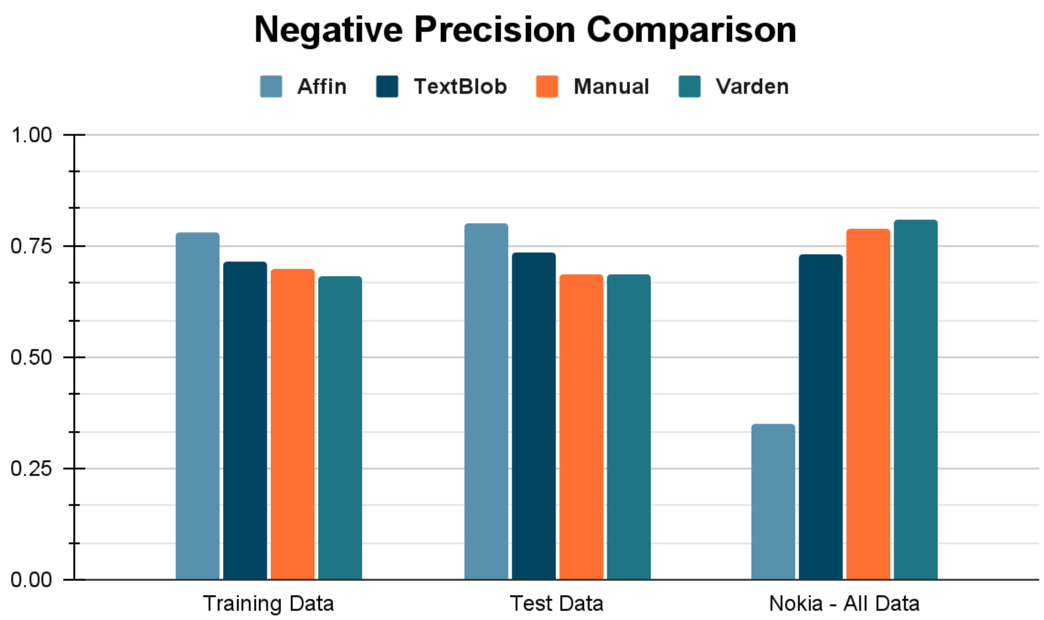


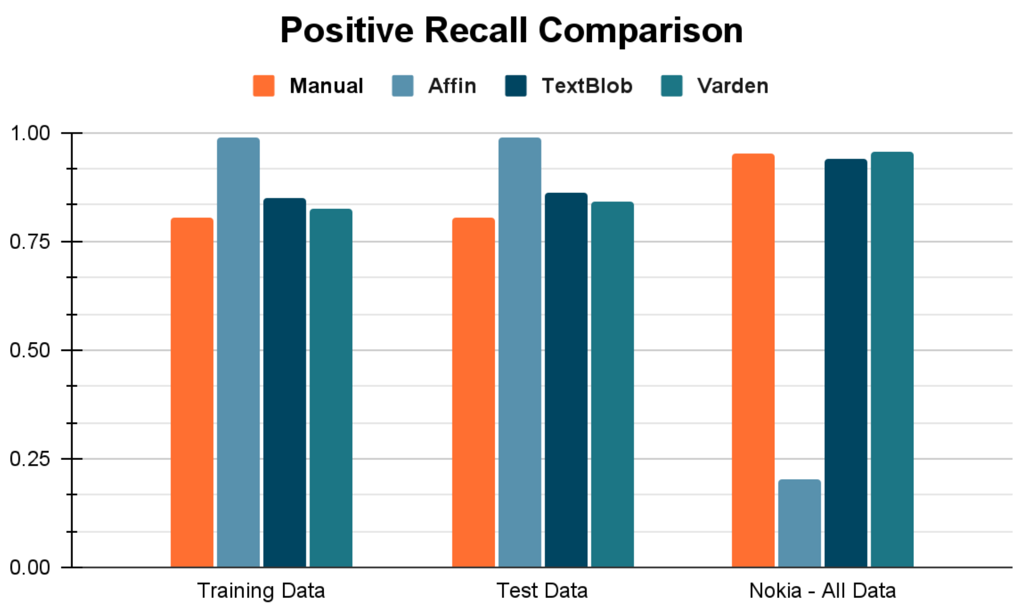


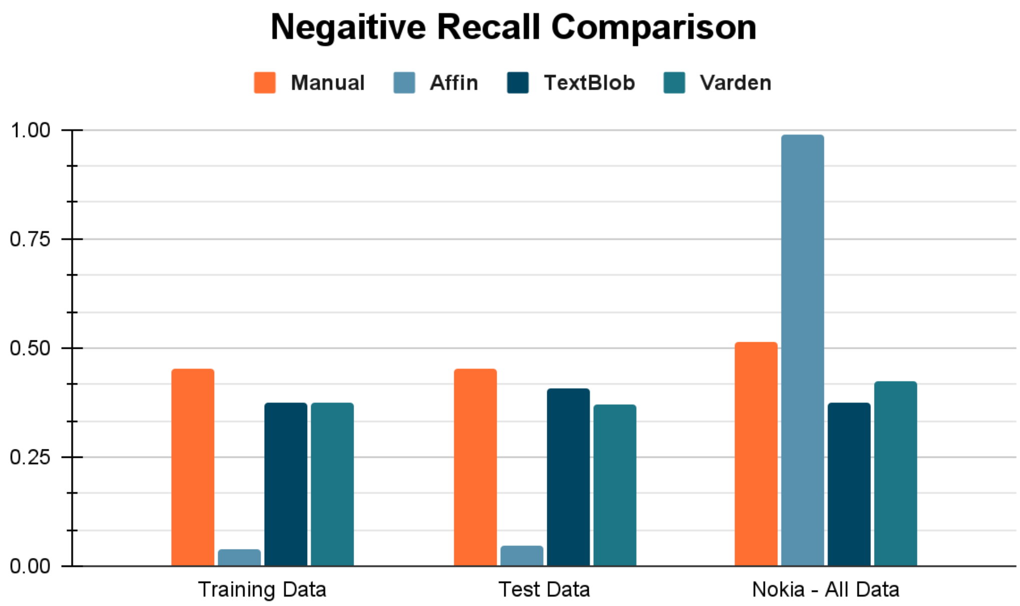


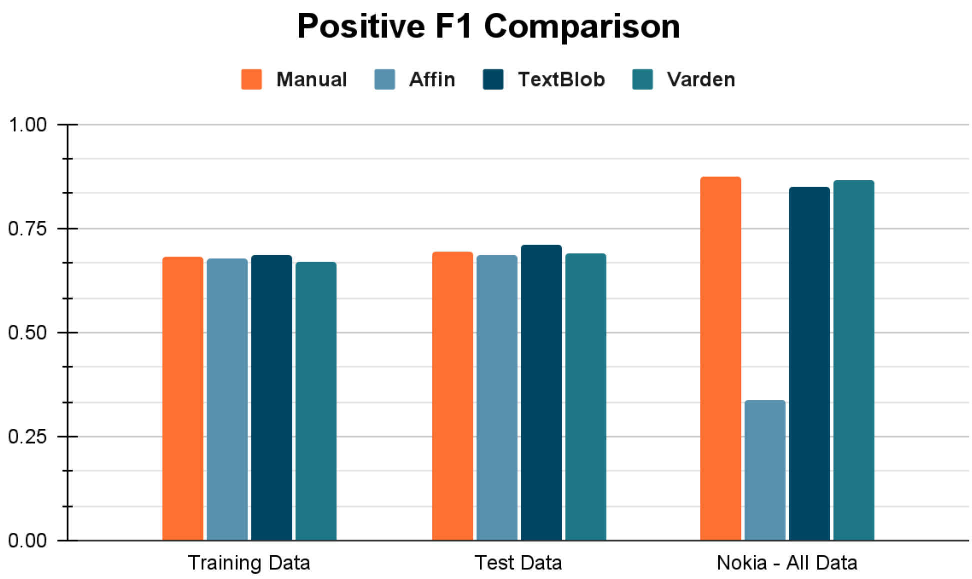
1.3 - Algorithm Improvements

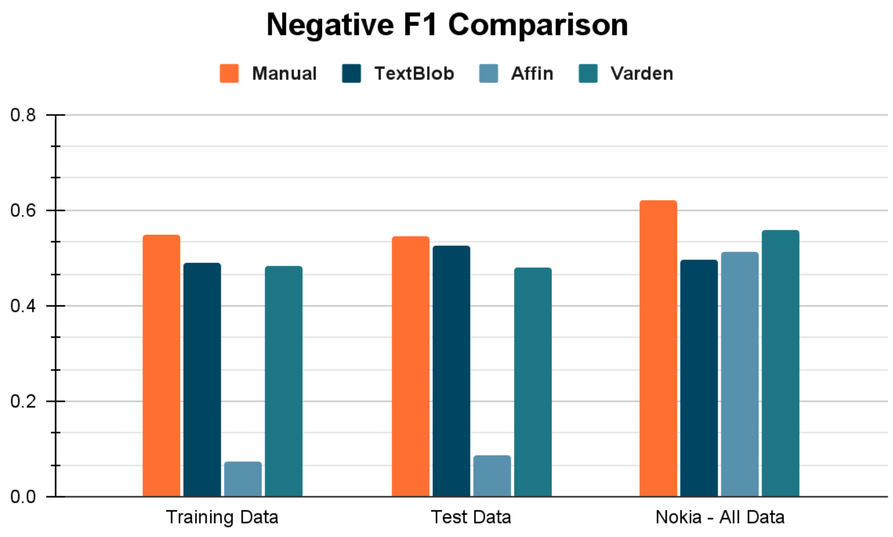
The following graphs present the comparisons of other performance measures from accuracy such as positive and negative precision, positive and negative recall, and positive and negative recall


Negative Precision Comparison
Affin
TextBlob
Manual
Varden
1.00
0.75
0.50
0.25
0.00
Training Data
Test Data
Nokia - All Data


Positive Recall Comparison
Manual
Affin
TextBlob
Varden
1.00
0.75
0.50
0.25
0.00
Training Data
Test Data
Nokia - All Data


Negaitive Recall Comparison
Manual
Affin
TextBlob
Varden
1.00
0.75
0.50
0.25
0.00
Training Data
Test Data
Nokia - All Data


Positive F1 Comparison
Manual
Affin
TextBlob
Varden
1.00
0.75
0.50
0.25
0.00
Training Data
Test Data
Nokia - All Data



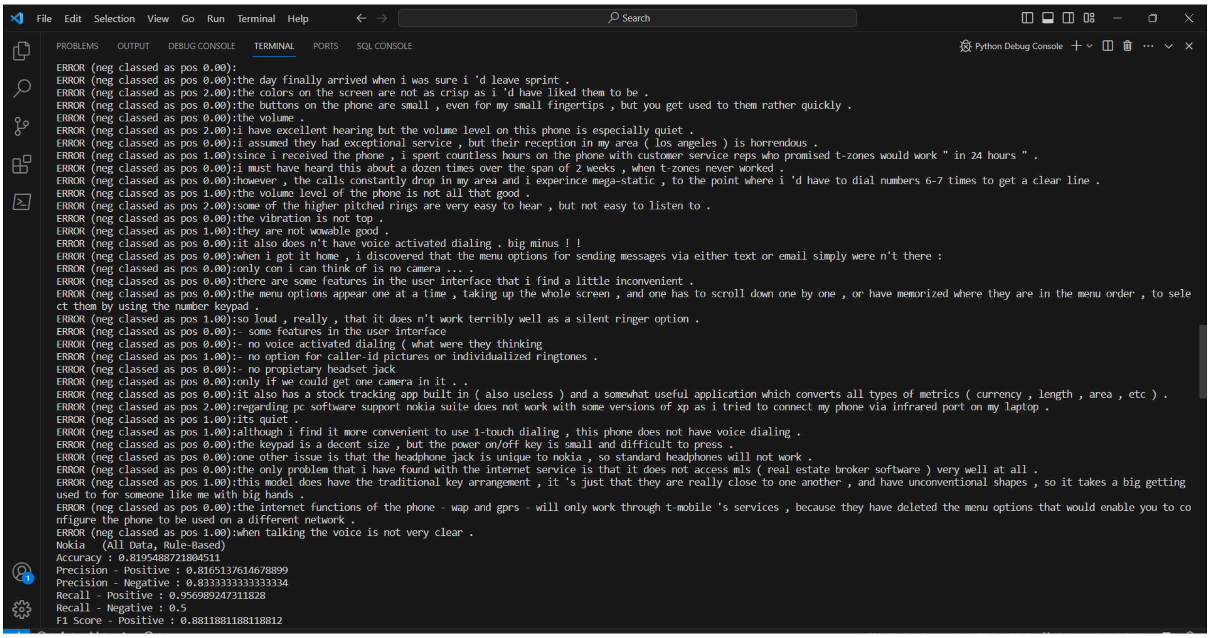
1.4 - Helpful Words

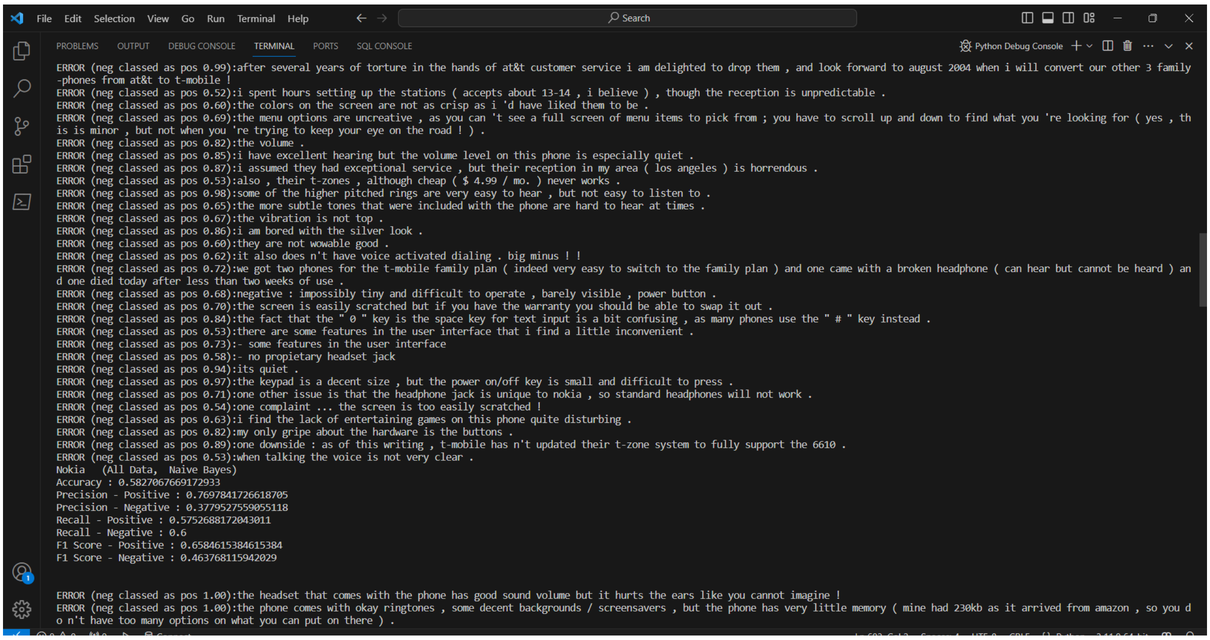
```
NEGATIVE:
['boring', 'generic', 'badly', 'unfunny', 'routine', 'mediocre', 'lame', 'poorly', 'mindless', 'stale', 'pointless', 'unless', 'annoying', 'tiresome', 'disguise', 'apparently', 'offensive', 'stupid', '
dull', 'empty', 'harvard', 'car', 'bore', 'junk', 'excuse', 'pass', 'amateurish', 'banal', 'cliché', 'wasted', 'fatal', 'shoot', 'seagal', 'uninspired', 'numbers', 'stealing', 'meandering', 'animal', '
chan', 'kung', 'trite', 'paper', 'product', 'incoherent', 'plodding', 'lifeless', 'supposed', 'waste', 'bother', 'inept', 'cable', 'missed', 'unintentional', 'ballistic', 'stiff', 'crap', 'conceived',
'ill', 'busy', 'unintentionally', 'hollow', 'ask', 'witless', 'pow', 'pathetic', 'writers', 'pile', 'crass', 'comparison', 'choppy', 'clumsily', 'leaden', 'disaster', 'relentlessly', 'lousy', 'sara', '
inane', 'unnecessary', 'tuxedo', 'pinocchio', 'impostor', '51', 'flat', 'tv', 'skip', 'halfway', 'wannabe', 'muddled', 'store', 'serving', 'horrible', 'obnoxious', 'smug', 'sadly', 'handful', 'flash',
'putting', 'commentary', 'six', 'vapid']


POSITIVE:
['conduct', 'simplicity', 'spite', 'hopeful', 'lane', 'subversive', 'despair', 'harrowing', 'moore', 'desperation', 'intoxicating', 'joyous', 'haunting', 'portrait', 'evocative', 'masterful', 'hearts',
 'lovers', 'format', 'understands', 'poem', 'unflinching', 'deadpan', 'depiction', 'splendid', 'integrity', 'exquisitely', 'current', 'nuanced', 'sobering', 'ingenious', 'delightfully', 'superbly', 're
sist', 'frailty', 'pianist', 'portrayal', 'sadness', 'absorbing', 'touching', 'intimate', 'thoughtful', 'warm', 'unique', 'undeniably', 'bourne', 'wrenching', 'russian', 'jealousy', 'transcends', 'brea
thtaking', 'sly', 'deft', 'gradually', 'richly', 'answers', 'delightful', 'startling', 'd', 'resonant', 'sides', 'heartbreaking', 'timely', 'grown', 'record', 'smarter', 'martha', 'captures', 'tour', '
captivating', 'spare', 'wry', 'wonderful', 'playful', 'iranian', 'detailed', 'lively', 'polished', 'respect', 'nicely', 'vividly', 'heartwarming', 'provides', 'tender', 'gem', 'wonderfully', 'deftly',
'realistic', 'mesmerizing', 'delicate', 'chilling', 'vivid', 'refreshingly', 'riveting', 'refreshing', 'affecting', 'powerful', 'inventive', 'unexpected', 'engrossing']
Ratio of words in powerlist present in the sentimentDictionary:  0.1644360496492175
```

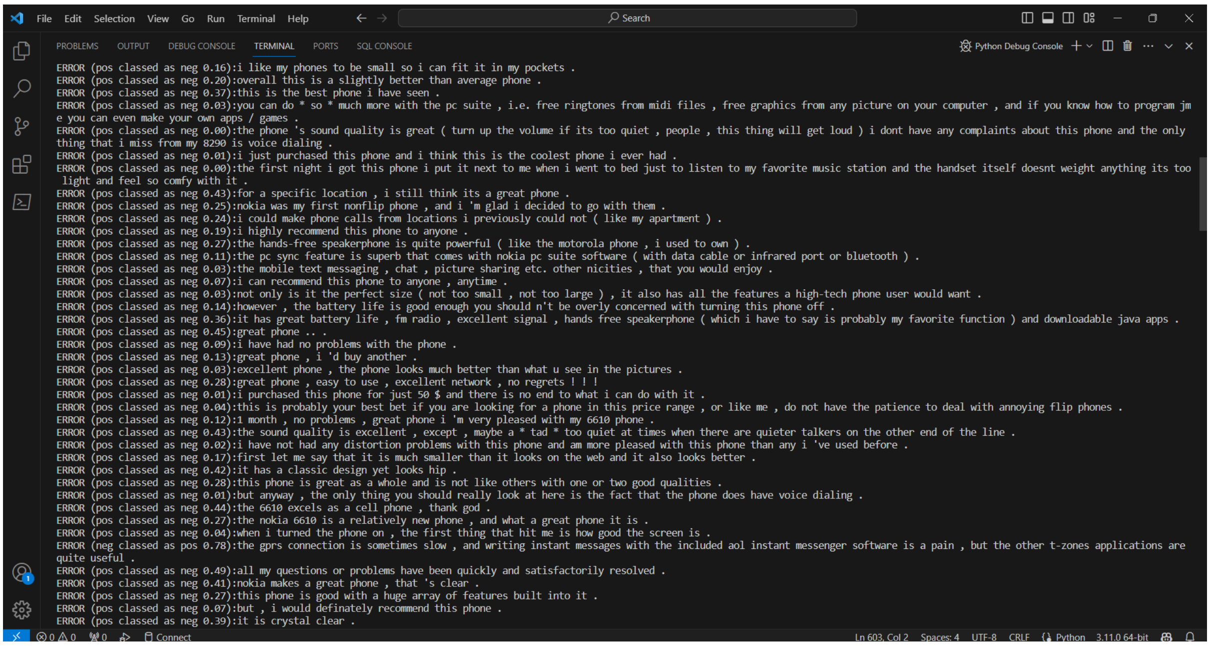
NEGATIVE:
['unfunny', 'mediocre', 'generic', 'badly', 'poorly', 'routine', 'boring', 'pointless', 'mindless', 'shoot', 'stale', 'bore', 'stupid', 'apparently', 'annoying', 'tiresome', 'unless', 'dull', 'chan', 'harvard', 'holes', 'save', 'offensive', 'product', 'disguise', 'junk', 'meandering', 'animal', 'tired', 'wasted', 'plodding', 'uninspired', 'pinocchio', 'pretentious', 'stealing', 'waste', 'amateurish', 'horrible', 'banal', 'trite', 'lousy', 'conceived', 'supposed', 'ill', 'inept', 'cable', 'kung', 'pathetic', 'sadly', 'incoherent', 'preachy', 'seagal', 'sara', 'inane', 'plain', 'tv', 'store', 'endeavor', 'fatal', 'unintentional', 'pile', 'comparison', 'produce', 'stiff', 'leaden', 'lifeless', 'numbingly', 'tuxedo', 'serving', 'soggy', 'benigni', 'impostor', 'lame', 'flat', 'coherent', 'numbers', 'halfway', 'unintentionally', 'wannabe', 'muddled', 'fails', 'unfocused', '100', 'sink', 'ask', 'pseudo', 'settles', 'arts', 'relentlessly', 'witless', 'pow', 'wilson', 'handful', 'missed', 'writers', 'looked', 'crass', 'vapid', 'schneider', 'thrown']

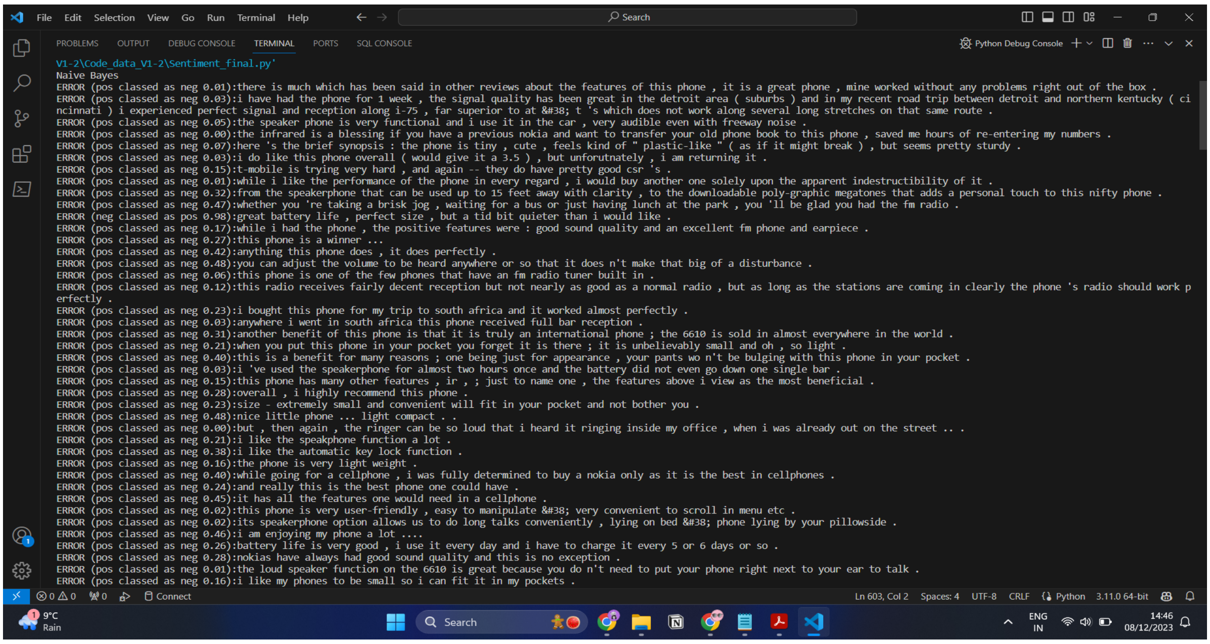
POSITIVE:
['secretary', 'delicious', 'smartly', 'nuanced', 'sobering', 'washington', 'quietly', 'marvel', 'richly', 'conduct', 'subversive', 'simplicity', 'gently', 'straightforward', 'ramsay', 'portrayal', 'intoxicating', 'color', 'masterful', 'joyous', 'intimate', 'vivid', 'beauty', 'grandeur', 'lane', 'hearts', 'undeniably', 'startling', 'buoyant', 'bourne', 'd', 'lovers', 'gut', 'potent', 'droll', 'treat', 'timely', 'grown', 'poem', 'unflinching', 'breathtaking', 'splendid', 'ingenious', 'explores', 'answers', 'spite', 'resist', 'gradually', 'rewarding', 'sadness', 'smarter', 'touching', 'unique', 'wrenching', 'format', 'honesty', 'tour', 'understands', 'scale', 'transcends', 'frailty', 'sly', 'literary', 'hopeful', 'changing', 'evocative', 'martha', 'resonant', 'unexpected', 'polished', 'captivating', 'tender', 'provides', 'sides', 'lively', 'jealousy', "world's", 'captures', 'playful', 'iranian', 'spare', 'respect', 'vividly', 'heartwarming', 'wonderfully', 'detailed', 'realistic', 'wonderful', 'wry', 'mesmerizing', 'chilling', 'powerful', 'warm', 'gem', 'ages', 'refreshingly', 'riveting', 'inventive', 'refreshing', 'engrossing']

1.5 - Error messages - Naive Bayes (Test Bayes), Rule-Based (Test Dictionary)









Citations

1. Finn Årup Nielsen, "A new ANEW: evaluation of a word list for sentiment analysis in microblogs", Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages. Volume 718 in CEUR Workshop Proceedings: 93-98. 2011 May. Matthew Rowe, Milan Stankovic, Aba-Sah Dadzie, Mariann

Hardey (editors)
https://github.com/fnielsen/afinn
2. Loria, S., 2018. textblob Documentation. *Release 0.15*, 2
https://textblob.readthedocs.io/
3. Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.
https://github.com/cjhutto/vaderSentiment