Modeling the Experimental Part

1. **Data used**

For the experiment we used the IMDb Movie Reviews with the following characteristics

* No. reviews: 50.000
* Classes: 2 (1 – positive, 0 – negative)
* Proportion: +/- = 26.097/23.903
* Format: CSV – 2 columns ‘review’ and ‘sentiment’

The text was all then prepared by lowering all capital letters and deleting any symbols and numbers.

Then we calculated the waits accordingly with our goals of study.

Bag-of-Words = the number of times each word appears in a review

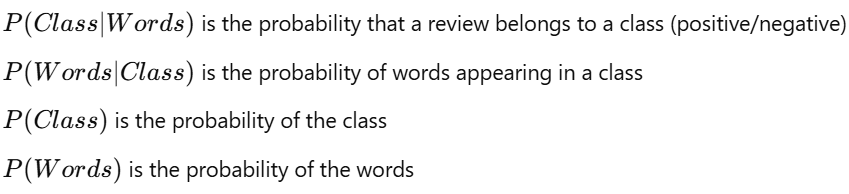
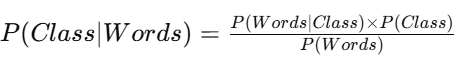
TF-IDF = the frequency of a word in a review \* the frequency of documents containing that word

1. **Experiments**

I compared the performance of the Naïve-Bayes algorithm in both data representations

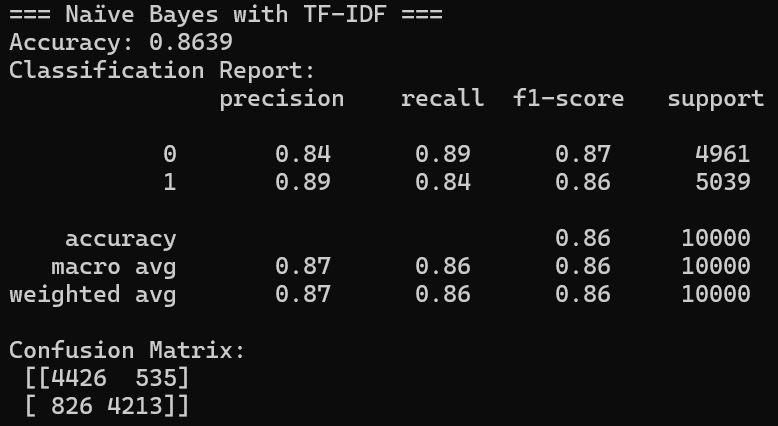
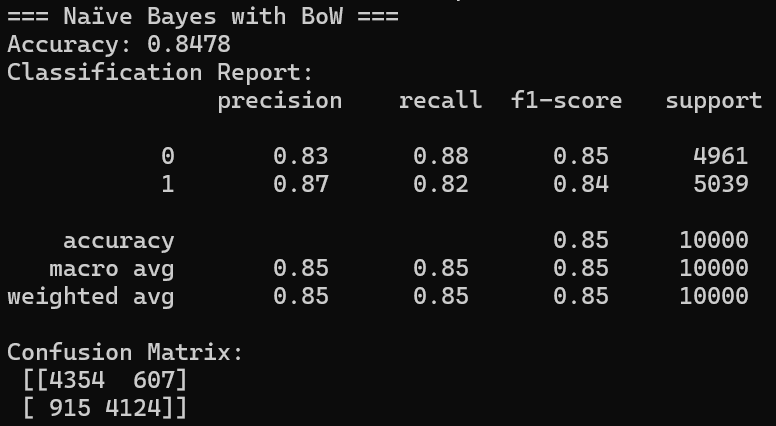
1. Dividing the data set 4:1 = training data : testing data
2. Weight extractions using CountVectorizer and TfidfVectorizer respectively
3. Training
4. Evaluation based on following metrics: Accuracy, Precision, Recall, F1-Score, Confusion Matrix
5. **Mathematic Model**

Naïve-Bayes works by calculating the probabilities that an array of words is in each class. The class with the greatest probability is selected. The probabilities are calculated as follows:



1. **Validation of Results**

After the experiment we obtained the following results:



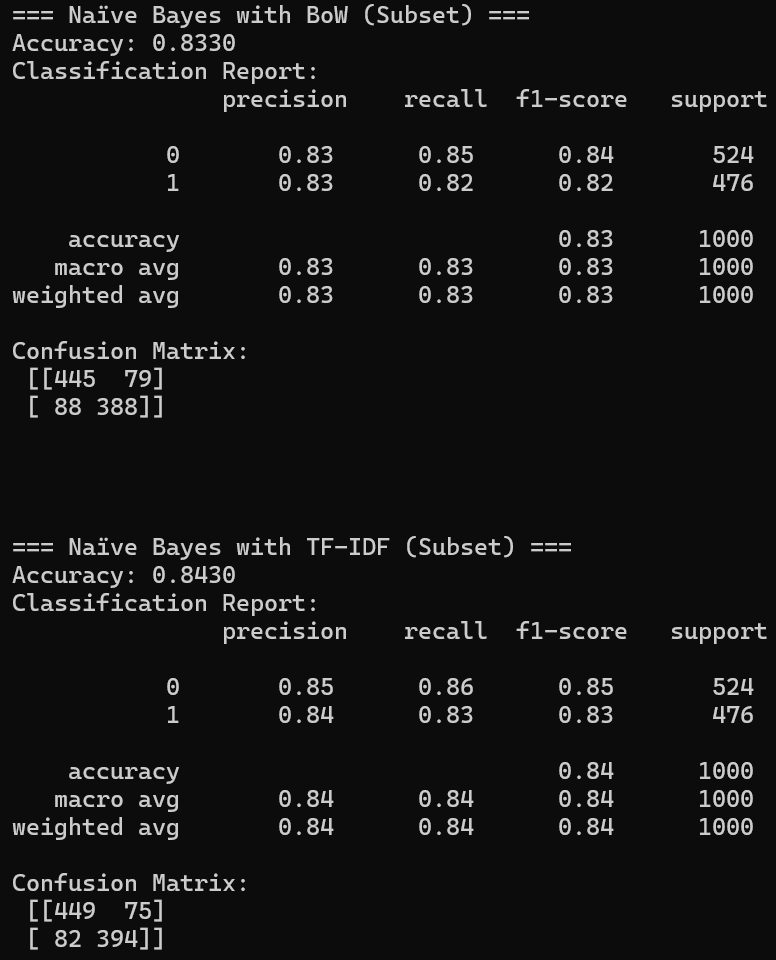
The results show that the TF-IDF representation led to better performance compared to BoW, with higher accuracy and better Precision and Recall scores for both classes. It’s not a huge difference but it’s better.

1. **Link to git Repository Containing All of the Implemented Code**

<https://github.com/Mariyoyay/Sentiment-Movie-Analysis-Naive-Bayes>

Case Study on Initial Data

Initial experiments were conducted on a subset of 5,000 reviews to validate the methodology. The purpose of this smaller-scale experiment was to test the effectiveness of the preprocessing steps and feature extraction techniques before applying them to the full dataset.



TF-IDF improves model consistency even on small datasets, resulting in a slightly better accuracy and balanced performance across precision and recall. The results indicate that TF-IDF captures word importance more effectively than BoW, leading to improved classification performance.

The difference in accuracy between BoW and TF-IDF is noticeable but not drastic, suggesting that while TF-IDF is better, BoW remains a competitive alternative.

These first results reinforce the importance of feature extraction methods in text classification tasks and validate our approach before proceeding to full-scale testing.

Related Work

In this study, we evaluated the performance of the Naïve Bayes classifier using two feature extraction methods—Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF)—on the IMDb Movie Reviews dataset. Our results showed that the TF-IDF representation slightly bested BoW, achieving an accuracy of 86.39% compared to 84.78%.

To put in context our findings, we compared them with existing literature that employed similar methodologies:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Study | Feature Extraction | Classifier | Accuracy | Link of Reaserch |
| Narayanan et al. (2013) | BoW + Bigram | |  | | --- | | Naïve Bayes | | |  | | --- | | 88.80% | | https://arxiv.org/abs/1305.6143 |
| Li (2023) | BoW | Naïve Bayes | 84.80% | https://www.atlantis-press.com/article/125994502.pdf |
| Dewi & Chen (2022) | TF-IDF | Complement Naïve Bayes | 75.00% | https://www.researchgate.net/publication/366126882 |
| This Study | BoW | Naïve Bayes | 84,78% |  |
| This Study | TF-IDF | Naïve Bayes | 86.39% |  |

Our accuracy aligns closely with that reported by Li (2023), who achieved 84.80% using a BoW representation with a Naïve Bayes classifier. Narayanan et al. (2013) reported a better accuracy of 88.80% by incorporating Bigrams into the BoW model, suggesting that including bi-gram features could enhance model performance. In contrast, Dewi & Chen (2022) achieved a lower accuracy of 75.00% using a TF-IDF representation with a Complement Naïve Bayes classifier, indicating that the choice of classifier and feature extraction method impacts performance a lot more.