Sentiment Analysis of English Texts

Amanaganti Chethan Reddy
230102117
I.I.T Guwahati

Satyaki Ray
230123080
I.I.T Guwahati

Ponnekanti Bipan Chandra
230102072
I.I.T Guwahati

***Abstract*—This project aims to obtain a dataset of movie reviews and apply different Machine Learning algorithms to analyze and classify them. It explored text classification accuracy while using two different classifiers, namely Naive Bayes and SVM for classifying balanced dataset. The results revealed that the SVM has a better accuracy level than Naive Bayes.**

***Index Terms*—Sentiment Analysis, Machine Learning, Natural Language Processing, SVM, Hyperparameter Tuning, Naive Bayes**

I. Introduction

Over the years, people have expressed their opinions on various matters over the internet. Detecting the true emotion of what a person is trying to convey, is very useful, especially to companies, to obtain valuable customer feedback. In this project, we performed sentiment analysis of text reviews on an IMDB Movie reviews dataset obtained from Kaggle and classified them into two categories - positive and negative.

II. Experiment Setup

A. Data Cleaning

We obtained an IMDB Movie reviews dataset from Kaggle. It consists of more than 50000 tweets, labelled either positive or negative. The distribution of the data set is as follows:

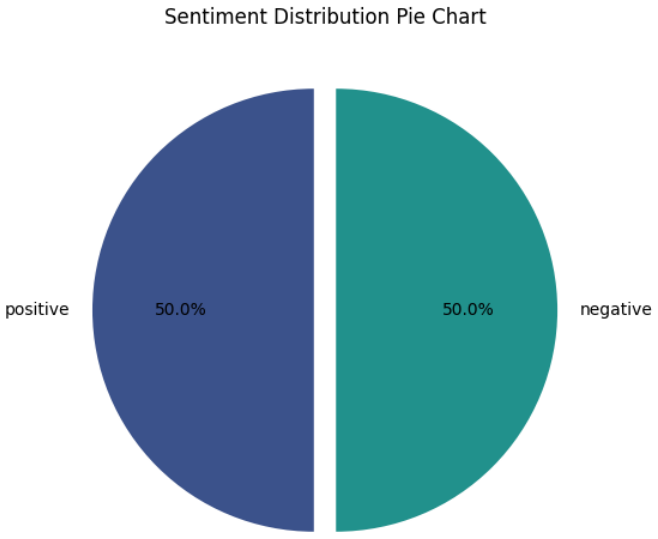


Fig. 1.

B. Data Preprocessing

Reviews were cleaned and preprocessed in the following way:

1) We started by removing all the HTML tags, special characters, and URL references from the data set using Python regular expressions.
2) Then, we used Python's WordNetLemmatizer() and word_tokenize() functions to split the text into a sequence of tokens and lemmatize the text into its base form.
3) Next, we removed all stopwords like is, as, was etc... from the lemmatized text.
4) Finally, we vectorized the lemmatized text using TF-IDF.

	review	clean_text	lemmatized_text	sentiment
0	One of the other reviewers has mentioned that ...	one of the other reviewers has mentioned that ...	one reviewer mention watch oz episode hook rig...	positive
1	A wonderful little production. The...	a wonderful little production br br the filmin...	wonderful little production br br film techniq...	positive
2	I thought this was a wonderful way to spend ti...	i thought this was wonderful way to spend time...	think wonderful way spend time hot summer week...	positive
3	Basically there is a family where a little boy...	basically there is family where little boy jak...	basically family little boy jake think zombie ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	petter mattei love in the time of money is vis...	petter mattei love time money visually stunnin...	positive

C. Dataset Training

The dataset was divided into two parts. The first part contains 80% of the total number of reviews of the dataset, and is used to train the model to classify the data under one attribute(training set). The remaining 20% of reviews were used to classify review attributes as positive or negative, i.e., test set.

D. Data Classifying

- Using the train_test_split() function, the dataset is divided and classified into two parts (training and test). We used 80% of the dataset to train the dataset and the last 20% to evaluate the model.
- Different machine learning algorithms are used for training the dataset (Naive Bayes and SVM(linear kernel)).
- For testing the model, the accuracy_score operator is utilized to measure the performance of the model. Classification Report and confusion matrix are also added.

III. Results and Analysis

The predicted values are obtained by using the trained model on the test set. The accuracy score is determined by comparing the predicted values to actual values. The SVM model gave better accuracy score than Naive Bayes.

A. Naïve Bayes

The accuracy of the model is 86.68%

The classification report is as follows:

```
Classification Report:
              precision    recall  f1-score   support

    Negative       0.88      0.85      0.86      4961
    Positive       0.86      0.88      0.87      5039

    accuracy                           0.87     10000
   macro avg       0.87      0.87      0.87     10000
weighted avg       0.87      0.87      0.87     10000
        Accuracy: 86.68%
```

Fig. 2.

The confusion matrix is as follows:

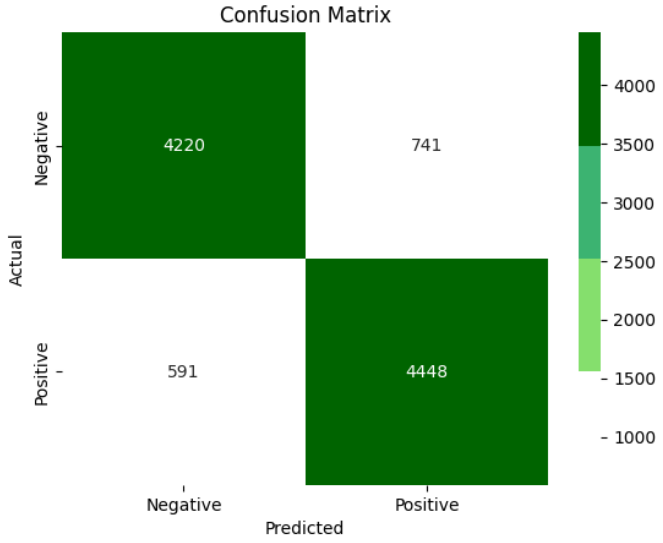


Fig. 3.

Naïve Bayes gives better accuracy with a balanced dataset because

a) Independent Assumption:: Naïve Bayes assumes that features are independent, which in most cases works well if the dataset is balanced.

b) Class prior probability:: Naïve Bayes calculates the probability of each class based on their proportion in the dataset. In the balanced dataset, class probabilities are closer, so the classifier is not biased towards any one class.

If the dataset is not balanced the Naïve Bayes can give biased results towards the class with the most data points.

B. SVM(linear kernel)

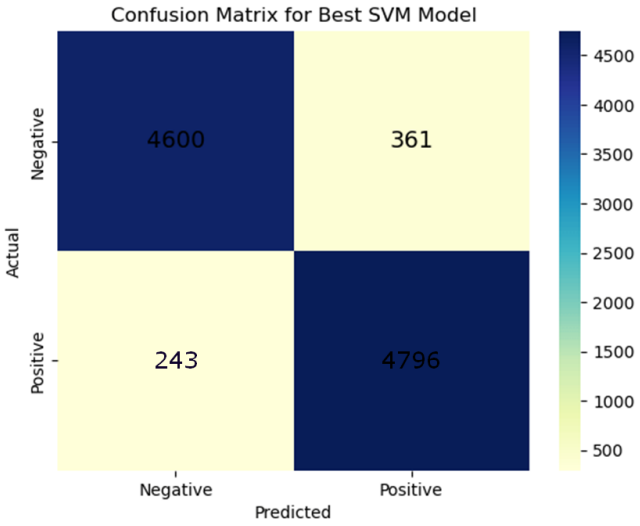
The accuracy of the model is 93.43%

The classification report is as follows:

```
Classification Report:
              precision    recall  f1-score   support

    Negative       0.94      0.93      0.93      4961
    Positive       0.93      0.94      0.94      5039

    accuracy                           0.93     10000
   macro avg       0.93      0.93      0.93     10000
weighted avg       0.93      0.93      0.93     10000

Accuracy: 93.43%
```

Fig. 4.

The confusion matrix is as follows:



SVM gives better accuracy with a balanced dataset because

a) Optimal Hyperplane:: SVM aims to find the hyperplane that maximally separates the classes. When the dataset is balanced, SVM can learn a well-defined boundary that separates the classes effectively, without being biased towards the majority class.

b) Balanced Support Vectors:: In balanced datasets,the vectors from classes contribute equally in defining the boundary.

If the dataset is unbalanced then SVM may place the hyperplane incorrectly, leading to misclassifications.

IV. Hyper-parameter tuning

A. Naive Bayes

a) Parameter and range selection: : The hyperparameter for our Naive Bayes model was the smoothing parameter α. We chose α from 20 randomly generated numbers on a logarithmic scale of 0.001 to 10.

b) Randomized Search: : Python's RandomizedSearchCV() function was employed and the no. of iterations was limited to 10 to make a faster and more efficient search through the given hyperparameter space.

c) Cross-Validation: : We kept a 5-fold cross-validation strategy, ensuring higher accuracy of the model and reduce overfitting.

d) Best Parameters: :

Best alpha: 0.20691380811147903
Best cross-validation accuracy: 0.8675499999999999
Best Parameters: 'alpha': 0.20691380811147903
Best Cross-Validation Score: 0.8675499999999999

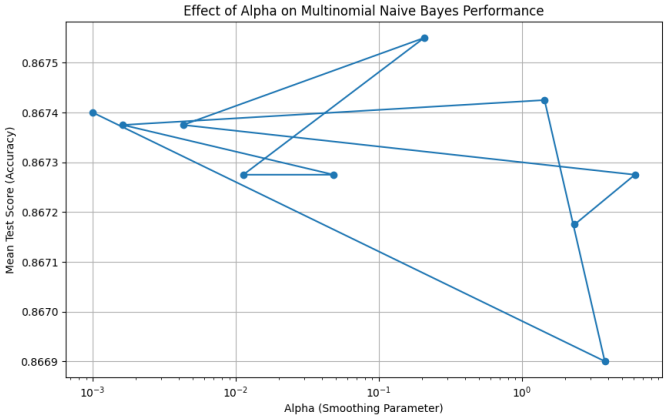


Fig. 5.

B. Support Vector Machine(SVM)

a) : First, we used Randomized Search along with Stratified K-fold technique on 20% of the data(10,000 samples) to select our kernel and gamma. We kept it to only 20% to avoid time limit and computational challenges.

Cross-Validation Results for Each Hyperparameter Combination:

	param_C	param_gamma	param_kernel	mean_test_score	std_test_score
0	7.590802	auto	linear	0.8461	0.010156
1	3.768696	auto	sigmoid	0.6603	0.037536
2	12.07317	auto	poly	0.5083	0.005938
3	9.016655	auto	poly	0.5083	0.005938
4	1.261672	auto	sigmoid	0.6603	0.037536
5	6.774172	auto	sigmoid	0.6603	0.037536
6	14.261452	auto	rbf	0.6603	0.037536
7	1.228232	auto	sigmoid	0.6603	0.037536
8	16.748853	auto	rbf	0.6603	0.037536
9	0.115575	auto	sigmoid	0.6603	0.037536

Fig. 6.

b) : Based on the above results, we found out linear kernel to be the best among all.

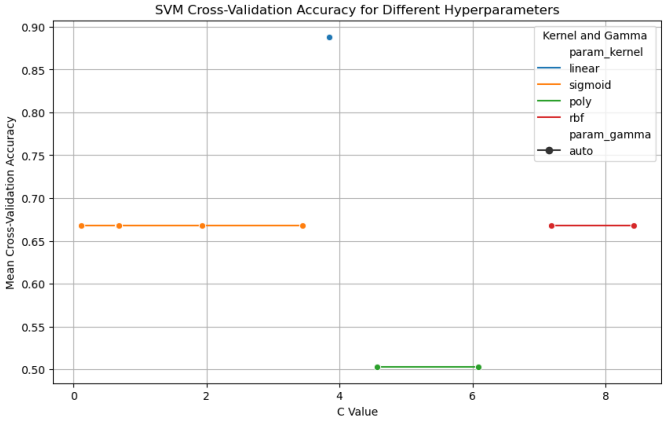


Fig. 7.

c) : Now, since we narrowed down the choice of our kernel, we again initialised RandomizedSearchCV and incorporated it with stratified k-fold cross validation, for tuning the hyperparameter for our linear kernel - C(regularization parameter).

Cross-Validation Results for Each Hyperparameter Combination:

	param_C	param_gamma	param_kernel	mean_test_score	std_test_score
0	3.845401	auto	linear	0.88824	0.002959
1	9.607143	auto	linear	0.87774	0.002953
2	7.419939	auto	linear	0.88122	0.003262
3	6.086585	auto	linear	0.88356	0.002880
4	1.660186	auto	linear	0.89512	0.002704
5	1.659945	auto	linear	0.89508	0.002656
6	0.680836	auto	linear	0.89814	0.002110
7	8.761761	auto	linear	0.87824	0.002842
8	6.11115	auto	linear	0.88352	0.002959
9	7.180726	auto	linear	0.88126	0.003248
10	0.305845	auto	linear	0.89674	0.002066
11	9.799099	auto	linear	0.87740	0.002952
12	8.424426	auto	linear	0.87894	0.003047
13	2.223391	auto	linear	0.89352	0.002924
14	1.91825	auto	linear	0.89446	0.003069

Fig. 8.

d) : Best parameters were those for which, std_test_score was low(i.e. less deviation) and mean_test_score was high i.e. higher accuracy.

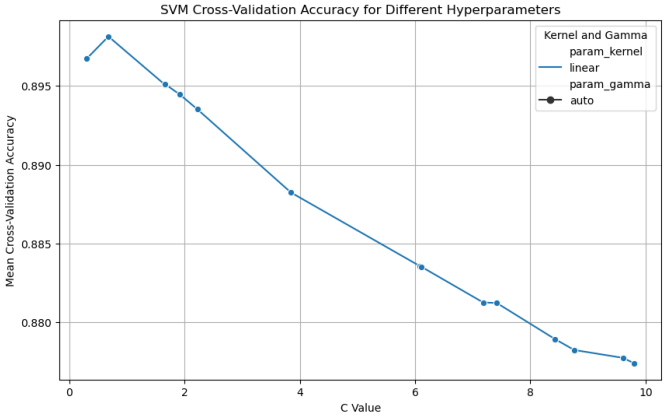


Fig. 9.

e) : Final hyperparameters:
Best Parameters: {'C': 0.6808361216819946, 'gamma': 'auto', 'kernel': 'linear'}

V. CONCLUSION

This project has proven to show the efficient usage of Machine Learning algorithms like Naïve Bayes and SVM in performing sentiment analysis of reviews. The project also heavily relies on the use of Natural Language Processing with Python libraries like NLTK. In this scenario, classifiers perform well on balanced datasets. Also, it was shown that the SVM had a greater accuracy than Naive Bayes in predicting sentiments. However, an issue is that the unbalanced datasets do not tend to perform well in our project, and our models are not reliable in this case. Also another problem may be the use of English idioms or phrases, and sarcastic feedbacks, where the user actually means the opposite of what he says. Our model may wrongly classify such reviews. To overcome these, advanced pre-trained transformers like BERT may be used. Besides, stronger hyperparameter tuning techniques along with the use of Deep Learning models may be further performed to improve accuracy in sentiment classification. Data balancing and tuning the classifier for unbalanced data remains an important step towards enhancement of outcomes concerning the domain of classification, irrespective of the application domain.

REFERENCES

[1] Kaggle Dataset Link: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[2] Arwa A. Al Shamsi, Reem Bayari, Said A. Salloum, "Sentiment Analysis in English Texts,"Advances in Science Technology and Engineering Systems Journal, January 2021