Machine Learning End Semester Project Presentation

News Article Classification

Group: Classifiers

Yesha Ajudia - AU1841078

Kartavi Baxi - AU1841079

Harsh Kakasaniya - AU1841085

Vimarsh Soni - AU1841121

Introduction

➢ Project based on a supervised Machine Learning Text Classification model

➢ Aim to predict the category of a given news article from the predefined set of categories

➢ Clean & process the data to ensure no distortion to model

➢ Learn the patterns & correlations in the data

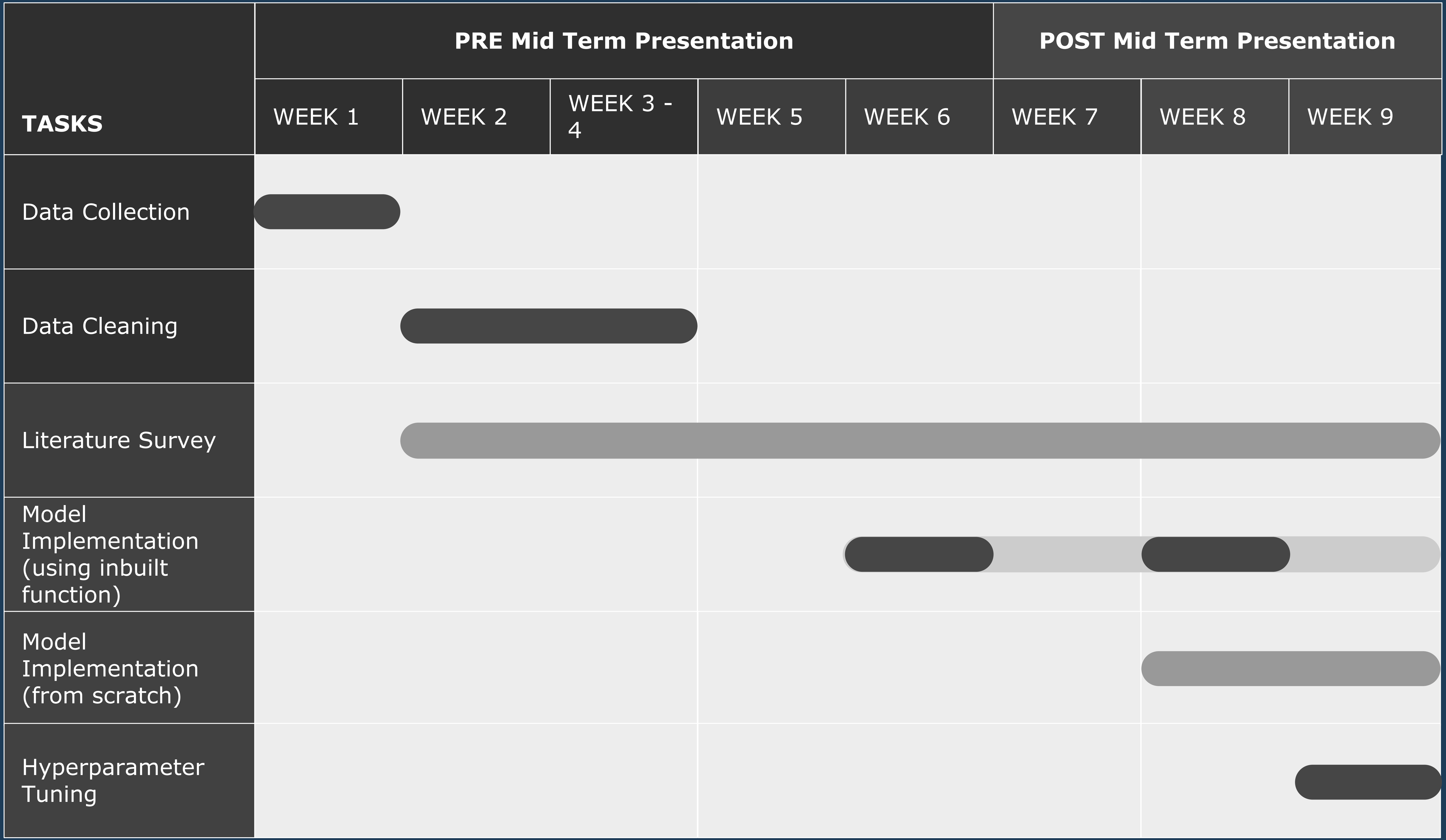
➢ Implement the right machine learning model

➢ Optimize the algorithm

Problem Statement

- Increased digitization
- Concept of E-News
- People prefer to read articles/news, sorted by categories
- Classifying news articles category-wise
- Classification based on keywords in the article
- Keywords defined based on number of occurrences or presence of the word

GANTT chart



TASKS
PRE Mid Term Presentation
POST Mid Term Presentation
WEEK 1
WEEK 2
WEEK 3 - 4
WEEK 5
WEEK 6
WEEK 7
WEEK 8
WEEK 9
Data Collection
Data Cleaning
Literature Survey
Model Implementation (using inbuilt function)
Model Implementation (from scratch)
Hyperparameter Tuning

Existing Body of Work

1. **Text Document Classification Algorithms**
   Rocchio algorithm, Boosting and bagging algorithms, etc [1]

2. **Machine Learning Techniques**
   Naive Bayes classifier, K-nearest neighbor classifiers, support vector machine, neural networks [2]

3. **Work done on Naive Bayes**
   Simple probabilistic classifier, successfully applied to document classification, comparison with other algorithms [3] [4]

4. **Two models of Naive Bayes**
   Multivariate Bernoulli Model and the Multinomial Model [5]

Existing Body of Work

5

Smoothing Techniques

Laplace smoothing, Dirichlet smoothing, Absolute Discounting [8]

6

Variants of Naive Bayes

Complement Naive Bayes, Weight-normalized Complement Naive Bayes, Transformed Weight-normalized Complement Naive Bayes [6] [7]

7

Language

English, Turkish, Arabic, etc.

Approach

Data Cleaning & Preprocessing

- Short forms to full forms, Remove extra characters other than the alphabets
- Convert to lower case, Stop words removal and Lemmatization

Label Encoding and Data Splitting

- Encoding the class labels
- Train - Test Split

Feature Extraction

- TF-IDF vectorizer with Uni-grams and Bi-grams
- Numeric form of features by transforming

Approach

- *Classification* — Multinomial Naive Bayes Classifier
- *Hyperparameter Tuning* — Laplace Smoothing evaluated using K-fold Cross Validation — Return optimal value of hyperparameter
- *Train the Model and Test* — Model Fitting — Prediction on test data

Final Results

Confusion Matrix

Without Hyperparameter Tuning

Predicted

Actual

```
[[ 97,   0,   3,   0,   2],
 [  1,  76,   0,   0,   0],
 [  2,   0,  82,   0,   0],
 [  0,   0,   0, 102,   0],
 [  0,   0,   0,   0,  80]]
```

With Hyperparameter Tuning

Predicted

Actual

```
[[ 98,   0,   2,   0,   2],
 [  1,  75,   0,   0,   1],
 [  1,   0,  83,   0,   0],
 [  0,   0,   0, 102,   0],
 [  0,   0,   0,   0,  80]]
```

Final Results

Classification Report

Without Hyperparameter Tuning

	precision	recall	f1-score	support
0	0.97	0.95	0.96	102
1	1.00	0.99	0.99	77
2	0.96	0.98	0.97	84
3	1.00	1.00	1.00	102
4	0.98	1.00	0.99	80
accuracy			0.98	445
macro avg	0.98	0.98	0.98	445
weighted avg	0.98	0.98	0.98	445

```
Accuracy of model on testing data is 0.9820224719101124
F1 Score of model on testing data is 0.9823857228125256
Log loss of model on testing data is 0.306335081018442
```

With Hyperparameter Tuning

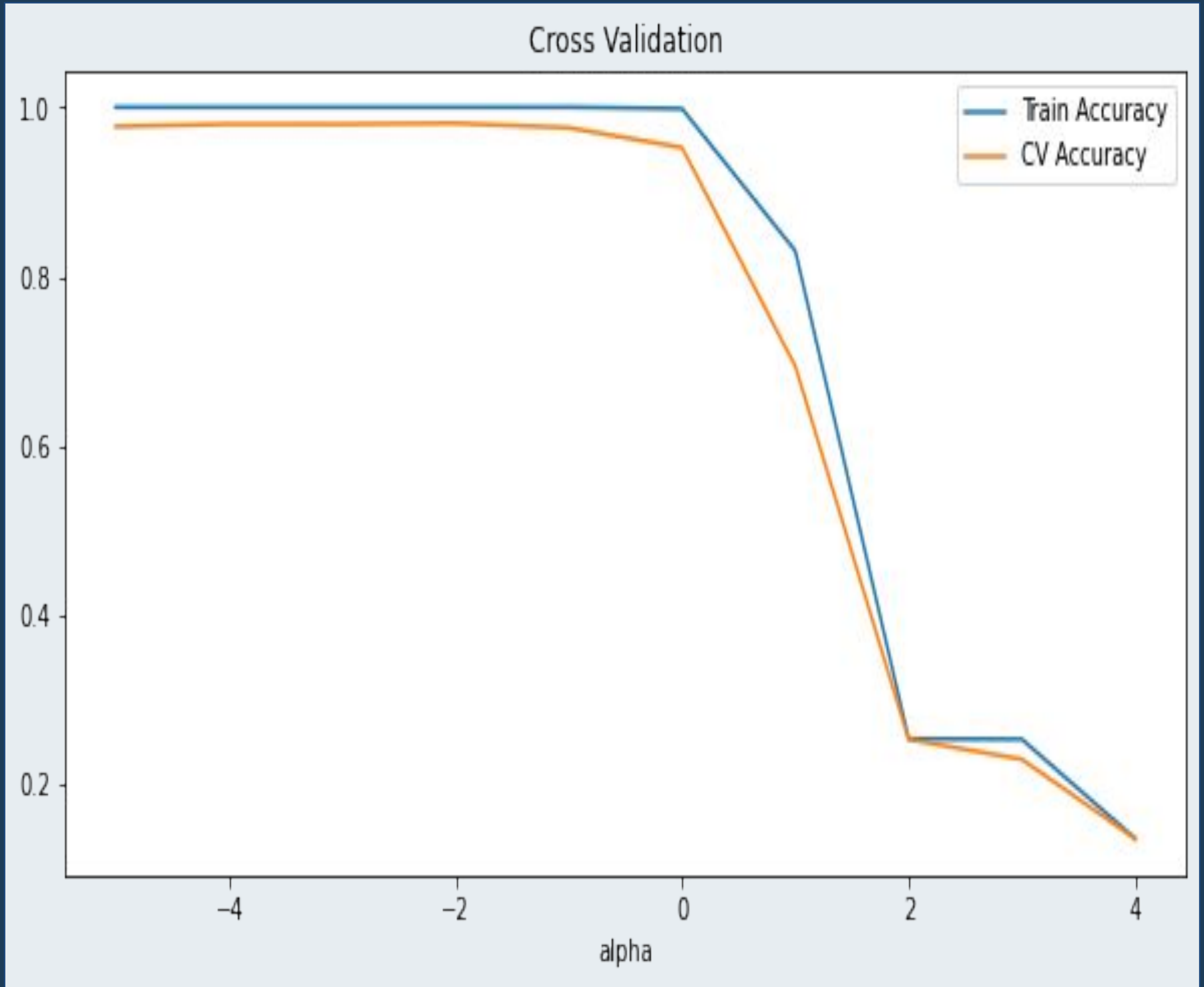
	precision	recall	f1-score	support
0	0.98	0.96	0.97	102
1	1.00	0.97	0.99	77
2	0.98	0.99	0.98	84
3	1.00	1.00	1.00	102
4	0.96	1.00	0.98	80
accuracy			0.98	445
macro avg	0.98	0.98	0.98	445
weighted avg	0.98	0.98	0.98	445

```
Accuracy of model on testing data is 0.9842696629213483
F1 Score of model on testing data is 0.9841965495401453
Log loss of model on testing data is 0.104879285470453 5
```

Increase in accuracy = 0.2%

Decrease in log loss = 0.2

Final Results



- Y-axis : Training and cross validation accuracy for different values of ***α***
- X-axis : $\log_{10}$ values of ***α***

Conclusion

- ★ All the necessary steps are taken for News Article Classification
- ★ Optimal Value of Hyper-parameter obtained is α = 0.01
- ★ This gives an accuracy of 98.43%.
- ★ As α increases, the training & cross validation accuracy decreases.
- ★ Tuning with different values of k in k-fold cross validation didn’t affect the value of α

```
For {'alpha': 1e-05}  acc of Train data is 1.0 and acc of CV data is 0.9775937880440704
For {'alpha': 0.0001}  acc of Train data is 1.0 and acc of CV data is 0.9770382453545505
For {'alpha': 0.001}  acc of Train data is 1.0 and acc of CV data is 0.9788231016338521
For {'alpha': 0.01}  acc of Train data is 1.0 and acc of CV data is 0.9792620391762096
For {'alpha': 0.1}  acc of Train data is 1.0 and acc of CV data is 0.9767168311904312
For {'alpha': 1}  acc of Train data is 0.9975205530154344 and acc of CV data is 0.9568729338514949
For {'alpha': 10}  acc of Train data is 0.8594179084964557 and acc of CV data is 0.7455858541874809
For {'alpha': 100}  acc of Train data is 0.2530901387822964 and acc of CV data is 0.2523111922161643
For {'alpha': 1000}  acc of Train data is 0.2713654757658291 and acc of CV data is 0.25239526671651014
For {'alpha': 10000}  acc of Train data is 0.11122527554374526 and acc of CV data is 0.110850341371371
Best Parameter is  {'alpha': 0.01}
Best F1 Score is  0.9792620391762096
```

Role of each member

Post Mid Term Presentation

	Yesha Ajudia	Kartavi Baxi	Vimarsh Soni	Harsh Kakasaniya
Literature Review	✓	✓	✓	✓
Implement Vectorizer	✓	✓	✓	✓
Explore Orange	✓			
Inbuilt Model Implementation	✓	✓	✓	✓
Model Implementation from scratch	✓	✓	✓	
Hyperparameter Tuning	✓	✓	✓	

References

[1] Kowsari, Kamran, et al. "Text classification algorithms: A survey." Information 10.4 (2019): 150.

[2] Patra, Anuradha, and Divakar Singh. "A survey report on text classification with different term weighing methods and comparison between classification algorithms." International Journal of Computer Applications 75.7 (2013).

[3] Vijayan, Vikas K., K. R. Bindu, and Latha Parameswaran. "A comprehensive study of text classification algorithms." 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017.

[4] McCallum, Andrew, and Kamal Nigam. "A comparison of event models for naive bayes text classification." AAAI-98 workshop on learning for text categorization. Vol. 752. No. 1. 1998.

[5] Aggarwal, Charu C., and ChengXiang Zhai. "A survey of text classification algorithms." Mining text data. Springer, Boston, MA, 2012. 163-222.

[6] Rennie, Jason D., et al. ”Tackling the poor assumptions of naive bayes text classifiers.” Proceedings of the 20th international conference on machine learning (ICML-03). 2003.

[7] Kibriya, Ashraf M., et al. ”Multinomial naive bayes for text categorization revisited.” Australasian Joint Conference on Artificial Intelligence.Springer, Berlin, Heidelberg, 2004.

[8] Indriani, Fatma, and Dodon T. Nugrahadi. ”Comparison of Naive Bayes smoothing methods for Twitter sentiment analysis.” 2016 International Conference on Advanced Computer Science and Information Systems(ICACSIS). IEEE, 2016.

THANK YOU!