

**AN OPTIMIZED MACHINE LEARNING PIPELINE
FOR
DETECTING RACIST COMMENTS WRITTEN IN
SINHALA LANGUAGE**

Chandrajith Priyadarshana Bandara Senadheera

189347N

Dissertation submitted in partial fulfillment of the requirements for the
degree Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

February 2021

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

Name: C.P.B. Senadheera

The supervisor should certify the thesis with the following declaration.

The above candidate has carried out research for the Masters thesis under my supervision.

Name: Dr. Uthayasanker Thayasivam.

Signature of the supervisor:

Date:

ACKNOWLEDGEMENT

I warmly convey my gratitude to my supervisor Dr. Uthayasanker Thayasivam, for his continuous guidance and support. His patience and knowledge inspired me and helped me to improve this study. He always steered me in the right direction whenever he thought I needed it and motivated me to complete my research.

Additionally, I appreciate the support provided by all the other university lecturers of the Department of Computer Science and Engineering, who supported me from the beginning and during the Masters degree.

Finally, I express my gratefulness to my family members for supporting me and believing in me. They kept me going on and this work would not have been possible without their support. Finally I would like to thank my closest friends who helped me in many ways and encouraged me to complete this research successfully.

ABSTRACT

Text classification is one of the core area of machine learning applications and it continuously improving with time. Hate speech classification in e-content became very popular over last few years and search engines, social media were the main interested parties of this area. “Racism content” is a sub part of the “Hate speech” area and many studies were carried out in this area. But surprisingly “Racism text classification in Sinhala language” is not very much popular. This study focus on building a machine learning pipeline for detecting racism comments written in Sinhala language. Sinhala racism text classification techniques can be used to remove unnecessary texts from social media, pages, blogs and many more text sources. This study was done to provide a solution to Sinhala racist text classification problems in social media, but also this study is valid for any text source that contain Sinhala text as Unicode text.

As the initial step, previous similar studies were reviewed and documented. Used techniques and results of similar studies were documented and reviewed. One similar study was selected as the baseline and set the baseline performance measures as the lower margin of the performance target. An architecture for the pipeline was designed and a methodology was selected as the next step. In this methodology, as the initial step, dataset was extracted and preprocessed. Stemming, stop word removal and conversion of text to basic character word were the main preprocessing steps. Features were extracted next and due to less number of racism data, an oversampling method was used to increase the training data. Six machine learning classifiers were selected and those were Random forest, Naïve bayes, SVM, Logistic regression, Ada boost and XGBoost. All the classifiers were trained with oversampled data. This was a major improvement point of the results. In order to increase the performance of these classifiers further, hyperparameter tuning was performed. Ensemble techniques were also used to increase the performance. As the ensemble techniques bagging, boosting and voting was used. After selecting the best classifiers from bagging and boosting, best three classifiers were set as the input to voting classifier to get the final results.

The study shows that each preprocessing steps improves the performance of classifiers and results. Each classifier behave differently in each step. This study highlights these differences and sensitivity of each classifier to various changes. In oversampling step and hyperparameter tuning step, all the classifiers reached to a stable level. Hyperparameter tuning identified as a critical step in Sinhala text classification. Finally the best results were shown by the voting classifier and selected as the best model in the study. The proposed pipeline performed better and highlighted the each classifiers performance differences. The pipeline was able to outperform the baseline with a greater accuracy. The study proves that the racism text classification is possible with selected classifiers and accurate data. Study also proves that a better Sinhala racism text classification pipeline can be built using other classifiers with the help of ensemble techniques and enhanced preprocessing techniques.

TABLE OF CONTENTS

DECLARATION	II
ACKNOWLEDGEMENT	III
ABSTRACT	IV
TABLE OF CONTENTS	V
LIST OF FIGURES	VIII
LIST OF TABLES	IX
LIST OF ABBREVIATIONS	X
LIST OF APPENDICES	XI
CHAPTER 1 - INTRODUCTION	12
1.1 TEXT CLASSIFICATION	12
1.2 SINHALA LANGUAGE	12
1.3 TEXT CLASSIFICATION IN SINHALA	13
1.4 HATEFUL CONTENT IN SOCIAL MEDIA	13
1.5 PROBLEM IDENTIFICATION AND PROPOSED SOLUTION	13
1.6 GOALS AND OBJECTIVES	14
CHAPTER 2 – LITERATURE REVIEW AND BACKGROUND	16
2.1 SIMILAR STUDIES	16
2.2 RESULTS COMPARISON	25
2.3 BACKGROUND	25
2.3.1 Oversampling	25
2.3.2 Classifiers	27
2.3.3 Bagging	30
2.3.4 Boosting	31
2.3.5 Voting	34
CHAPTER 3 – METHODOLOGY AND EXPERIMENT	36
3.1 DATASET	37
3.1.1 Finding the Dataset	37
3.1.2 Preprocessing the dataset	38
3.1.3 Annotate the dataset	42
3.1.4 Train and Test Data	43
3.1.5 Baseline dataset	43
3.1.6 Challenges of dataset selection and preprocessing	44
3.2 FEATURE EXTRACTION	44
3.2.1 Tokenizer	44
3.2.2 TF-IDF Vectorizer	44
3.2.3 N-grams	45
3.2.3 fastText	45
3.2.4 Choose the best feature generation technique	46
3.2.5 SMOTE Oversampling	46
3.2.6 Challenges of feature extraction	46

3.3	CLASSIFICATION	47
3.3.1	K-Fold cross validation.....	47
3.3.1	Hyperparameter tuning	47
3.3.2	Challenges of classification.....	50
3.4	BAGGING.....	51
3.4.1	Bagging classifier.....	51
3.4.2	How bagging classifier applied to this study.....	51
3.4.3	Challenges of bagging.....	51
3.5	BOOSTING	52
3.5.1	Boosting Classifiers.....	52
3.5.2	Hyperparameter tuning	53
3.5.3	Challenges of boosting	54
3.6	VOTING	54
3.6.1	Challenges of voting.....	54
3.7	EVALUATION.....	55
3.7.1	Four main evaluators	55
3.7.2	Accuracy.....	56
3.7.3	Precision.....	56
3.7.4	Recall.....	56
3.7.5	F Measure.....	57
3.7.6	F-beta Measure	57
3.7.7	F2 Measure – Main Performance Measure.....	58
3.7.8	ROC Curve	58
3.7.9	Precision-Recall Curve	59
3.7.10	ROC curve vs. Precision-Recall Curve	60
3.8	BASELINE EXPERIMENTS.....	61
3.8.1	Experiment 1.....	61
3.8.2	Experiment 2.....	61
CHAPTER 4 – RESULTS.....		62
4.1	RESULTS AFTER MAJOR STEPS.....	62
4.1.1	Results after applying TF-IDF Vectorizer	62
4.1.2	Results after stemming.....	63
4.1.3	Results after removing stop words.....	64
4.1.4	Results after text convert to base character.....	65
4.1.5	Results after SMOTE oversampling.....	66
4.1.6	Results after applying N-grams	68
4.1.7	Results after applying fastText	69
4.1.8	Results after hyperparameter tuning	70
4.1.9	Results after bagging	71
4.1.10	Results after voting.....	72
4.1.11	Result comparison with precision recall graph	73
4.1.12	Result comparison with ROC curve.	73
4.2	RESULT IMPROVEMENTS OF EACH CLASSIFIER.....	73
4.2.1	Random forest classifier	74
4.2.2	Naïve Bayes Classifier.....	76
4.2.3	SVM Classifier.....	79
4.2.4	Logistic Regression Classifier.....	81
4.2.5	Ada Boost Classifier	83
4.2.6	XGBoost Classifier	85
4.3	RESULT COMPARISON WITH BASELINE.....	88

CHAPTER 5 – DISCUSSION	89
5.1 SUMMARY.....	89
5.2 FINDINGS.....	90
5.2.1 Apply TF-IDF vectorizer.....	90
5.2.2 Stemming.....	91
5.2.3 Removing stop words.....	91
5.2.4 Text convert to base character.....	91
5.2.5 SMOTE oversample.....	92
5.2.6 N-grams.....	92
5.2.7 fastText.....	93
5.2.8 Hyperparameter tuning.....	93
5.2.9 Apply bagging.....	94
5.2.10 Apply voting.....	94
5.3 EVALUATION.....	95
5.4 WRONG CLASSIFICATIONS.....	95
5.5 BASELINE COMPARISON.....	97
5.5 LIMITATIONS.....	97
5.6 FUTURE WORK.....	97
CHAPTER 6 – CONCLUSION	98
REFERENCES	99
APPENDICES.....	104

LIST OF FIGURES

<i>Figure 2.1: Class imbalance problem</i>	26
<i>Figure 2.2: SMOTE new generated instances</i>	26
<i>Figure 2.3: Hyperplanes in SVM</i>	27
<i>Figure 2.4: Importance of weighted class</i>	28
<i>Figure 2.5: Logistic regression function</i>	30
<i>Figure 2.6: Bagging process</i>	31
<i>Figure 2.7: Boosting process</i>	32
<i>Figure 2.8: XGBoost improvement process</i>	33
<i>Figure 2.9: Voting classifier behavior</i>	35
<i>Figure 3.1: Pipeline for Sinhala racism text classification</i>	36
<i>Figure 3.2: Main flow of the experiment</i>	37
<i>Figure 3.3: Sample hate post in Facebook</i>	38
<i>Figure 3.4: Sample ROC curve</i>	58
<i>Figure 3.5: Sample precision recall curve</i>	60
<i>Figure 4.1: Random forest classifier performance</i>	75
<i>Figure 4.2: Naive bayes classifier performance</i>	78
<i>Figure 4.3: SVM classifier performance</i>	80
<i>Figure 4.4: Logistic regression classifier performance</i>	82
<i>Figure 4.5: Ada boost classifier performance</i>	84
<i>Figure 4.6: XGBoost classifier performance</i>	87

LIST OF TABLES

<i>Table 3.1: Preprocessing steps</i>	39
<i>Table 3.2: Stemming actions</i>	41
<i>Table 3.3: Inter-rater agreement</i>	43
<i>Table 3.4: Main evaluators</i>	55
<i>Table 4.1: Results of TF-IDF vectorizer</i>	62
<i>Table 4.2: Results after stemming</i>	64
<i>Table 4.3: Results after removing stop words</i>	65
<i>Table 4.4: Results after text convert to base character</i>	66
<i>Table 4.5: Results after SMOTE oversample</i>	67
<i>Table 4.6: Results after applying N-grams</i>	68
<i>Table 4.7: Results after applying fastText</i>	69
<i>Table 4.8: Results after hyperparameter tuning</i>	70
<i>Table 4.9: Results after bagging</i>	71
<i>Table 4.10: Results after voting</i>	72
<i>Table 4.11: Result comparison of random forest classifier</i>	74
<i>Table 4.12: Result comparison of naive bayes classifier</i>	76
<i>Table 4.13: Result comparison of SVM classifier</i>	79
<i>Table 4.14: Result comparison of logistic regression classifier</i>	81
<i>Table 4.15: Result comparison of ada boost classifier</i>	83
<i>Table 4.16: Result comparison of XGBoost classifier</i>	85
<i>Table 4.17: Result comparison with baseline</i>	88
<i>Table 5.1: Wrong classifications</i>	96

LIST OF ABBREVIATIONS

Abbreviation	Description
BoW	Bag of Words
CNN	Convolutional Neural Network
IDF	Inverse Document Frequency
PoS	Part of Speech
ROC	Receiver Operating Characteristic
SATC	Semi-Automated Text Classifier
SMOTE	Synthetic Minority Oversampling Technique
TF	Term Frequency
UTF	Unicode Transformation Format
WWW	World Wide Web

LIST OF APPENDICES

Appendix	Description	Page
Appendix – I	Literature review summary part 1	104
Appendix – II	Literature review summary part 2	105
Appendix – III	Literature review summary part 3	106
Appendix – IV	ROC curves comparison	107
Appendix – IV	Precision recall curves comparison	108