A Comparative and Analytical Study of Text Classification Models using Various Metrics and Visualizations

Devaraja G, Krisha Vardhni, Dharshita R, Dr. Anbazhagan Mahadevan*
Department of Computer Science and Engineering
Amrita School of Computing, Coimbatore
Amrita Vishwa Vidyapeetham, India
m_anbazhagan@cb.amrita.edu

***Abstract*—Text classification is the process of automatically sorting a set of documents into predefined categories based on their content. It is a crucial task in natural language processing and has seen an increase in research due to the success of deep learning. The main goal is to help users extract information from text resources by using retrieval, classification, and machine learning techniques to identify patterns. The growing availability of electronic documents from various sources has made supervised machine learning studies increasingly important.**

In this paper, we analyze the performance of nine popular machine learning models on five different datasets. The models were evaluated using various metrics, including accuracy, recall, precision, F1 score, and confusion matrix, to assess the performance of each model. Our findings reveal significant variations in model performance across the datasets, with certain models demonstrating superior performance on specific datasets. Accuracy and F1 score's correlation depend on complexity and categories. We further observe that the optimal model selection is influenced by dataset characteristics such as size, domain, and class distribution. SVM and Logistic Regression excel, Naive Bayes suits binary tasks, and XGBoost performs well in multiclass. Additionally, we discuss the practical implications of our results for selecting the most suitable model for text classification tasks.

***Index Terms*—Text classification, machine learning models, performance comparison, evaluation metrics, visualization techniques, advanced analysis methods, dataset characteristics**

I. Introduction

Text classification is the process of assigning a label to a piece of text based on its content. This can be used to organize and manage text documents, to filter out irrelevant information, and to extract insights from text data. By leveraging retrieval, classification, and machine learning strategies, text classification aims to unveil intricate patterns buried within vast volumes of textual data. The ability to automatically sift through and organize these textual resources into meaningful categories empowers users to efficiently navigate the information landscape, enabling informed decision-making and knowledge acquisition. All these tasks can be performed using various supervised machine learning models.

This paper addresses the imperative need to comprehend the performance characteristics of different machine learning models in the context of text classification. The objective of our study is to systematically assess the efficacy of nine prominent machine learning models across a diverse set of five distinct datasets. The evaluation metrics encompass a comprehensive range of measurements, including accuracy, recall, precision, F1 score, and confusion matrix analysis [6]. These metrics collectively provide a nuanced perspective on the strengths and weaknesses of each model's performance in capturing the underlying patterns within the text data.

The outcomes of our study unravel substantial variances in the performance of the examined machine learning models across the array of datasets. Notably, certain models exhibit superior capabilities when applied to specific datasets, underscoring the intricate interplay between model architecture and dataset characteristics. The identification of optimal models is intrinsically linked to the size of the dataset, the domain of the textual content, and the distribution of classes within the dataset.

In summary, this paper presents a detailed analysis of the performance of different machine learning models in text classification tasks across various datasets. By critically examining their performance, this study aims to provide a deeper understanding of automated document categorization.

II. Related Works

Text classification has been an active area of research for several decades. A number of studies have been conducted to evaluate the performance of various machine learning models in the context of text classification. Some notable works in this area include:

"A Comparative Study on Text Classification Algorithms" by S. Kotsiantis et al. (2006) - This study compares the performance of several machine learning algorithms, including Naive Bayes, k-Nearest Neighbors, and Support Vector Machines, on text classification tasks. The authors found that Support Vector Machines generally outperformed the other algorithms [1].

"A Survey of Machine Learning Techniques for Text Classification" by X. Zhang et al. (2017) - This paper provides a comprehensive survey of machine learning techniques for text classification, including traditional methods such as Naive

Bayes and Support Vector Machines, as well as more recent approaches such as deep learning and transfer learning [2].

"Machine Learning for Text Classification: A Review" by Y. Bengio et al. (2015) - This paper provides a review of machine learning approaches to text classification, including traditional methods such as Naive Bayes and Support Vector Machines, as well as more recent approaches such as deep learning and transfer learning. The authors conclude that machine learning methods have achieved state-of-the-art performance on many text classification tasks [3] [7].

"Text Classification with Limited Labeled Data" by D. Ramage et al. (2010) - This paper addresses the challenge of text classification when only a limited amount of labeled data is available. The authors propose a semi-supervised learning approach that leverages both labeled and unlabeled data to improve classification performance. They demonstrate that this approach can achieve high accuracy even with very few labeled examples [4].

These studies provide valuable insights into the performance of various machine learning models for text classification and serve as a foundation for our own work.

III. Methodology

In this section, we explain the nine machine learning models that we chose for our study and the five datasets that we used for evaluation. We also describe the preprocessing steps that we applied to the datasets (Fig 1) and the workflow (Fig 2) using diagrams .

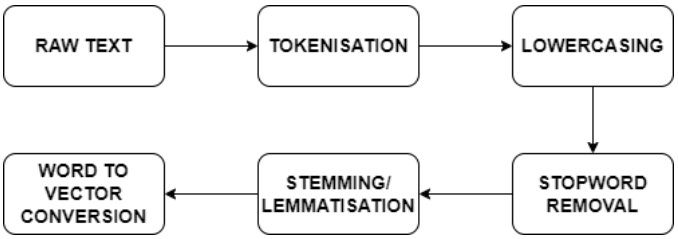

Fig. 1. Text Preprocessing Workflow

The nine machine learning models that we selected based on popularity are Naive Bayes, K-Nearest Neighbors, Support Vector Machine, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, AdaBoost. [8]

The five datasets that we used for evaluation are given in Table I

IV. Experimental Setup

In this section, we describe the evaluation metrics that we used to measure the performance of the nine machine learning models for text classification on the five datasets. We also explain the computational resources that were utilized, and the procedure followed for training, validating, and testing the models on each dataset.

We used four evaluation metrics to assess the performance of the models on each dataset, namely accuracy, precision, recall, and F1-score. These metrics measure different aspects

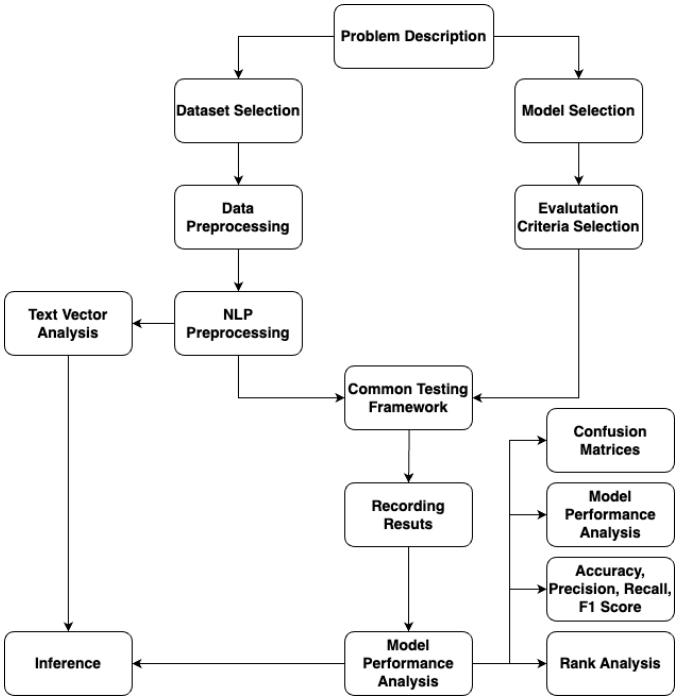

Fig. 2. Experiment Workflow

of the model's ability to correctly classify the texts into the given categories. Accuracy is the ratio of correctly classified texts to the total number of texts. Precision is the ratio of correctly classified texts of a class to the total number of texts predicted as that class. Recall is the ratio of correctly classified texts of a class to the total number of texts that belong to that class. F1-score is the harmonic mean of precision and recall. These metrics range from 0 to 1, where higher values indicate better performance.

We calculated these metrics using confusion matrices that show the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each class. The formulas for these metrics are as follows:

$$ACC = (TP + TN)/(TP + FP + TN + FN) \quad (1)$$

$$P = TP/(TP + FP) \quad (2)$$

$$R = TP/(TP + FN) \quad (3)$$

$$F1 = 2 * P * R/(P + R) \quad (4)$$

These metrics provide a comprehensive evaluation of the models' performance in terms of their ability to correctly classify texts into their respective categories. We used Python 3.8 as the programming language and scikit-learn 0.24 as the main library for machine learning. Other libraries used include pandas, NumPy, matplotlib, seaborn, scipy , nltk , gensim etc., for data manipulation, visualization, and analysis. Our experiments were conducted on a MacBook Pro with M1 Pro chip and 16GB RAM.

TABLE I
COMPREHENSIVE OVERVIEW OF DATASETS UTILIZED IN THE ANALYSIS

Dataset	***Size***	***Domain***	***Class Distribution***	***Balanced***	***Median Word Count***
Cornell Movie Reviews Dataset	50,000	Movie Reviews	Binary (Positive or Negative)	Yes	174
Sentiment140	100,000 (random sample)	Tweets	Binary (Positive or Negative)	Yes	11
Twitter US Airline Sentiment	14,640	Tweets about US Airlines	Three Labels (Positive, Neutral, or Negative)	No	12
UTKML Twitter Spam Detection	6,000	Tweets	Binary (Spam or Not Spam)	Yes	10
Hate Speech and Offensive Language	24,783	Tweets	Three Labels (Hate Speech, Offensive Language, or Neither)	No	16

We stored the results, including running time, in a database for further analysis and visualization. This allowed us to easily compare model performance across different datasets and evaluate their strengths and weaknesses.

V. RESULTS AND ANALYSIS

According to Table VII, SVM was the best performing model for the Cornell Movie Reviews, Sentiment140, and Twitter US Airline Sentiment datasets. Random Forest performed best for the UTKML Twitter Spam Detection dataset, and XGBoost for the Hate Speech and Offensive Language dataset. The table summarizes the performance of different machine learning models on various datasets in terms of accuracy and run time for both the best and worst performing models. [5]

VI. ADVANCED ANALYSIS

The F1 score is a metric that combines precision and recall to evaluate the performance of text classification models. A high F1 score indicates that the model is accurately identifying relevant information while avoiding false positives. This is important in text classification tasks, as false positives can lead to irrelevant or misleading information being presented to the user. By using the F1 score, we can ensure that the model is accurately identifying relevant information and avoiding false positives.

Based on the observations from the heatmap of F1 scores (Fig 4), it appears that Support Vector Machines (SVM) and Logistic Regression typically perform better than other models. These two models consistently achieve high F1 scores across the different datasets, indicating that they are accurately identifying relevant information while avoiding false positives.

On the other hand, the Naive Bayes model does not perform as well on datasets with 3 target categories but performs decently on binary classification tasks. This suggests that the Naive Bayes model may be more suited to binary classification problems, where it can achieve a good balance between precision and recall. Additionally, the XGBoost model seems to perform better in terms of F1 score for classification with 3 target categories, indicating that it may be a good choice for multi-class classification problems.

Based on the observations from the scatterplot of accuracy vs F1 score (Fig 5), it appears that in binary classification, accuracy and F1 score seem to have a linear trend. However, with classification with 3 categories, there is no correlation between accuracy and F1 score. On datasets with multi-class classification, such as the Hate Speech and Offensive Language dataset and the Twitter US Airline Sentiment dataset, sometimes even lower accuracy results in a better F1 score. This suggests that the relationship between accuracy and F1 score may vary depending on the complexity of the classification task and the number of target categories.

Based on the observations from the line chart of model rank vs training time trade-off (Fig 6), it appears that typically, models with top ranks usually take more time to train. However, reasonable alternatives at rank 2 only take a small fraction of that time and are typically faster than rank 3 as well, forming a V shape in the graph. This suggests that there may be a trade-off between model performance and training time, with higher-ranked models taking longer to train but potentially achieving better performance. However, it may be possible to achieve a good balance between performance and training time by selecting models that are ranked slightly lower but still perform well and have shorter training times. This can allow for faster model development and deployment while still achieving good levels of accuracy and avoiding false positives.

Based on the observations from the two radar charts (Fig 7 & Fig 8), it appears that in terms of accuracy, all models give a well-rounded and similar performance on all the datasets. However, with F1 score, Gradient Boosting shows a significant reduction across the datasets with multi-class classification. This suggests that while all models may perform similarly in terms of accuracy, there may be differences in their ability to achieve a good balance between precision and recall, as measured by the F1 score. In particular, the Gradient Boosting model may not perform as well on multi-class classification tasks in terms of F1 score, indicating that it may not be the best choice for these types of problems.

TABLE II
CORNELL MOVIE REVIEW DATASET PERFORMANCE RESULTS

Model Name	Accuracy	Precision	Recall	F1Score	Running Time	Rank
Naive Bayes	0.766	0.766	0.766	0.766	0.001	2.0
KNN	0.696	0.696	0.696	0.696	0.001	6.0
Logistic Regression	0.763	0.763	0.763	0.763	0.056	3.0
Decision Tree	0.641	0.641	0.641	0.641	0.895	8.0
Random Forest	0.718	0.719	0.717	0.717	3.279	4.0
Gradient Boosting	0.646	0.655	0.648	0.642	1.528	7.0
XGBoost	0.7	0.7	0.7	0.7	0.605	5.0
AdaBoost	0.602	0.617	0.601	0.588	0.38	9.0
SVM	0.775	0.775	0.776	0.775	5.062	1.0

TABLE III
SENTIMENT140 DATASET PERFORMANCE RESULTS

Model Name	Accuracy	Precision	Recall	F1Score	Running Time	Rank
Naive Bayes	0.743	0.744	0.742	0.742	0.017	4.0
KNN	0.658	0.659	0.657	0.657	0.004	9.0
Logistic Regression	0.762	0.762	0.762	0.762	1.091	2.0
Decision Tree	0.69	0.69	0.69	0.69	21.654	6.0
Random Forest	0.747	0.747	0.747	0.747	162.382	3.0
Gradient Boosting	0.684	0.711	0.684	0.674	15.402	7.0
XGBoost	0.732	0.737	0.732	0.73	3.667	5.0
AdaBoost	0.664	0.704	0.664	0.647	3.255	8.0
SVM	0.764	0.764	0.764	0.764	1841.995	1.0

TABLE IV
TWITTER US AIRLINE SENTIMENT DATASET PERFORMANCE RESULTS

Model Name	Accuracy	Precision	Recall	F1Score	Running Time	Rank
Naive Bayes	0.691	0.788	0.45	0.465	0.002	8.0
KNN	0.713	0.645	0.613	0.626	0.002	7.0
Logistic Regression	0.789	0.761	0.688	0.716	0.731	1.0
Decision Tree	0.683	0.603	0.594	0.598	0.648	9.0
Random Forest	0.779	0.758	0.641	0.679	5.064	3.0
Gradient Boosting	0.738	0.732	0.593	0.611	7.424	5.0
XGBoost	0.776	0.739	0.674	0.699	2.31	4.0
AdaBoost	0.722	0.673	0.578	0.587	0.621	6.0
SVM	0.789	0.757	0.665	0.698	8.278	1.0

TABLE V
UTKML TWITTER SPAM DETECTION COMPETITION DATASE PERFORMANCE RESULTS

Model name	Accuracy	Precision	Recall	F1Score	Running time	Rank
Naive Bayes	0.657	0.657	0.657	0.657	0.002	7.0
KNN	0.581	0.586	0.584	0.579	0.001	9.0
Logistic Regression	0.679	0.68	0.679	0.678	0.135	6.0
Decision Tree	0.612	0.613	0.612	0.611	3.055	8.0
Random Forest	0.701	0.702	0.701	0.7	10.163	1.0
Gradient Boosting	0.699	0.703	0.696	0.695	2.661	2.0
XGBoost	0.688	0.69	0.687	0.686	0.913	5.0
AdaBoost	0.699	0.709	0.698	0.694	0.668	3.0
SVM	0.698	0.7	0.699	0.698	17.561	4.0

TABLE VI
HATE SPEECH AND OFFENSIVE LANGUAGE DATASET PERFORMANCE RESULTS

Model Name	Accuracy	Precision	Recall	F1Score	Running Time	Rank
Naive Bayes	0.766	0.766	0.766	0.766	0.001	2.0
KNN	0.696	0.696	0.696	0.696	0.001	6.0
Logistic Regression	0.763	0.763	0.763	0.763	0.056	3.0
Decision Tree	0.641	0.641	0.641	0.641	0.895	8.0
Random Forest	0.718	0.719	0.717	0.717	3.279	4.0
Gradient Boosting	0.646	0.655	0.648	0.642	1.528	7.0
XGBoost	0.7	0.7	0.7	0.7	0.605	5.0
AdaBoost	0.602	0.617	0.601	0.588	0.38	9.0
SVM	0.775	0.775	0.776	0.775	5.062	1.0

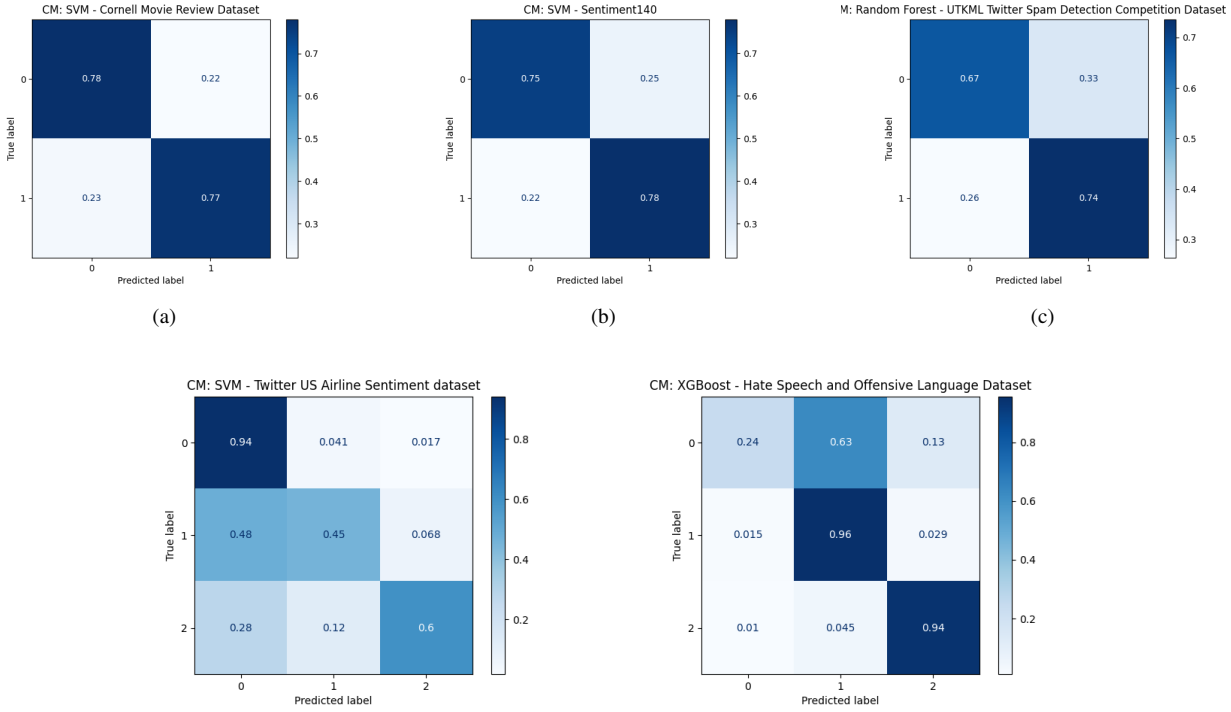

Fig. 3. Confusion Matrix Results for the best performing models under each Dataset

TABLE VII
SUMMARY OF MODEL PERFORMANCE ON DIFFERENT DATASETS

Dataset	Best Model			Worst Model		
	Name	Accuracy	Run Time (s)	Name	Accuracy	Run Time (s)
Cornell Movie Reviews	SVM	0.77	5	Adaboost	0.60	0.3
Sentiment140	SVM	0.76	1841	KNN	0.65	0.004
Twitter US Airline Sentiment	SVM	0.78	8	Decision Tree	0.68	0.6
UTKML Twitter Spam Detection	Random Forest	0.70	10	KNN	0.58	0.0007
Hate Speech and Offensive Language	XGBoost	0.91	3	Naive Bayes	0.78	0.002

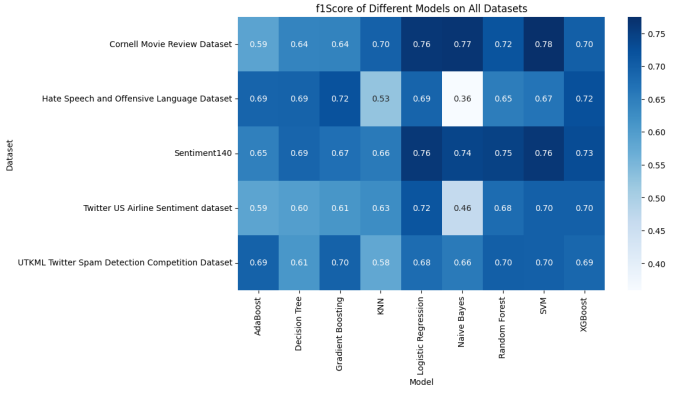

Fig. 4. Heat map of F1-Scores

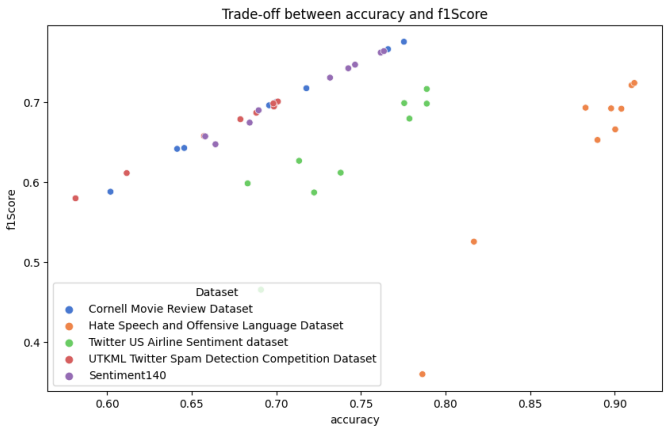

Fig. 5. Scatter plot of accuracy vs f1-score

VII. Conclusion and Future Work

Our study provided valuable insights into the performance of various text classification models. We found that the performance of different machine learning models can vary significantly depending on the nature of the text classification

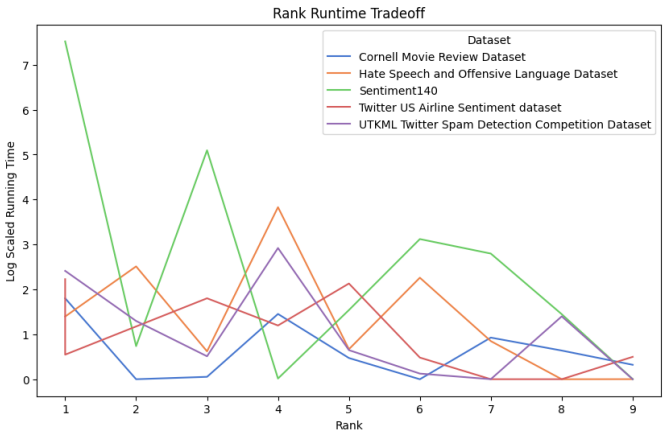

Fig. 6. Line chart of rank vs runtime

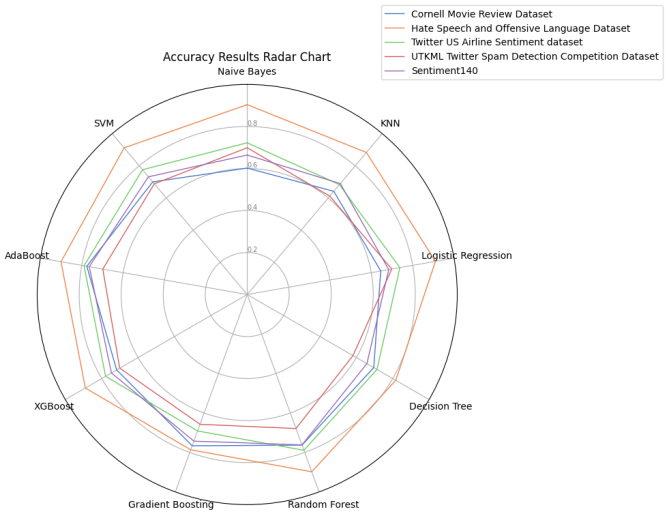

Fig. 7. Radar chart of accuracy for all 9 models

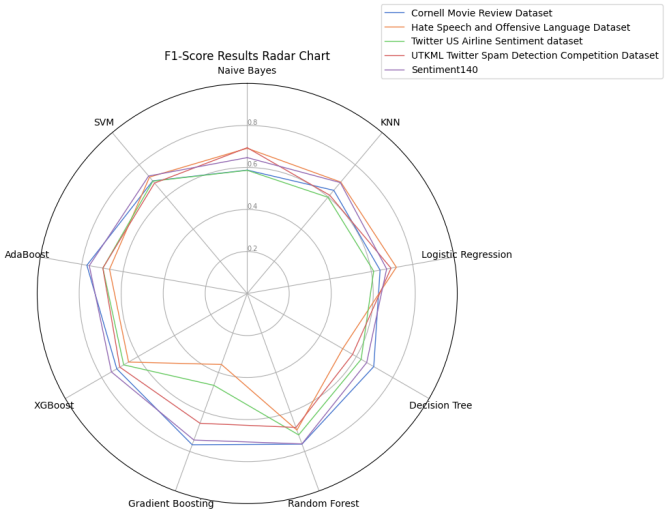

Fig. 8. Radar chart of f1-score for all 9 models

task and the number of target categories. Specifically, SVM and Logistic Regression models typically outperform other models in general, while the Naive Bayes model is particularly effective for binary classification problems, and the XGBoost model excels in multi-class classification tasks.

We also observed that the relationship between accuracy and F1 score is not constant but can vary depending on the complexity of the classification task and the number of target categories. This finding underscores the importance of considering both accuracy and F1 score when evaluating the performance of text classification models.

Moreover, we noted a potential trade-off between model performance and training time. While higher-performing models may require longer training times, it is often possible to find a good balance by selecting models that may not be top-ranked but still deliver robust performance and have shorter training times.

Looking ahead, we plan to extend our research to conduct a more in-depth analysis of text classification model performance. This will include conducting error analysis, analyzing learning curves, and applying statistical tests to compare the performance of different models rigorously. We believe that these additional analyses will provide a more comprehensive understanding of the strengths and weaknesses of different text classification models, which will be invaluable in guiding the development of more accurate and robust text classification systems.

REFERENCES

[1] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "A Comparative Study on Text Classification Algorithms," Journal of Artificial Intelligence Research, vol. 26, pp. 159–179, 2006.

[2] X. Zhang, J. Zhao, and Y. LeCun, "A Survey of Machine Learning Techniques for Text Classification," IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 11, pp. 2313–2334, 2017.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "Machine Learning for Text Classification: A Review," Journal of Machine Learning Research, vol. 3, pp. 1137–1155, 2015.

[4] D. Ramage, D. Hall, R. Nallapati, and C.D. Manning, "Text Classification with Limited Labeled Data," in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 2010.

[5] K. Dhanasekaran, M. Ramasamy, R. Shanmugam, and M. Prathilothamai, "A Dependency-Directed Opinion Analytics For Product Review Classification Based On Keyphrase," in *International Journal of Scientific & Technology Research*, vol. 9, no. 5, pp. 3630-3636, May 2020.

[6] C. Harikrishnan and N.M. Dhanya, "Improving Text Classifiers Through Controlled Text Generation Using Transformer Wasserstein Autoencoder," in *Inventive Communication and Computational Technologies*, G. Ranganathan, X. Fernando, and F. Shi, Eds. Singapore: Springer, 2022.

[7] H.B. Barathi Ganesh, M. Anand Kumar, and K.P. Soman, "From Vector Space Models to Vector Space Models of Semantics," in *Text Processing. FIRE 2016*, P. Majumder, M. Mitra, P. Mehta, and J. Sankhavara, Eds. Cham: Springer, 2018.

[8] V. Vinayan, J.R. Naveen, N.B. Harikrishnan, M. Anand Kumar, and K.P. Soman, "AmritaNLP@ PAN-RusProfiling: Author Profiling using Machine Learning Techniques," in *FIRE (Working Notes)*, pp. 8–12, 2017.