CSCI 636: Pattern Recognition & Data Mining I

Project Report on **Categorization of Clinical Trials using Machine Learning and Natural Language Processing**

Michael Welford (Z-1679714) and **A S M Shahadat Hossain** (Z-1907296)

Abstract

This paper applies machine learning and natural language processing techniques to categorize clinical trials. 15 classification models were used to classify clinical trials based on their Altmetric Attention Scores (AAS). Separately numerical and text-based features were used to train and test the models. For the numerical-based models, the Decision Tree classifier was shown to have the best performance of the general classifiers used. The use of ensemble methods increased the performance of the models as expected. The Random Forest and extreme gradient boosting models had the best performance overall with the XGBClassifier tuned with randomized search having the largest ROC/AUC values. Similarly, XGBClassifier outperformed the other classifiers applied on the text features. The highest test accuracy was found 0.945 for the numerical features and 0.801 for the text features.

Keywords: Altmetric Attention Scores, Clinical Trials, Altmetrics, Machine Learning, Natural Language Processing, Text Classification.

1. Introduction

Altmetrics or “alternative metrics" is an increasingly growing topic in the field of the impact of scientific research. Altmetrics are defined as complimentary impact measurements to the usual metrics such as citation counts and h-index and are typically associated with attention in the news and social media websites such as Facebook, Twitter, etc. [1] In this project, we attempt to make models to categorize clinical trials automatically based on their altmetrics with the help of artificial intelligence (AI) techniques such as machine learning and natural language processing.

We hypothesize that words included in the titles and abstracts of the clinical trials have an impact on their Altmetric scores. The research community behind the clinical trials can benefit from this work as they can get an idea of how the Altmetric score they are going to receive for the trial, that is how much the outreach will be of that clinical trial and how aware people are of their work. Better outreach can also inspire more participants to participate in those clinical trials. During the ongoing COVID-19 pandemic situation, it is obvious how significant the clinical trials are for bringing up some life-saving drugs or vaccines [2]. Since we hypothesize that the words or phrases chosen in the titles of clinical trials have an impact on the attention on social media that the trial receives, classifiers are applied on two types of features – numerical features and text features.

Prior studies on the Altmetrics of clinical trials and scientific research mostly determined the correlations between online mentions and citation counts or used trial start year, trial completion year, trial length, phase of the trial, and gender of the participants in order to predict whether a trial's total mentions will be above or below a median value. However, in this study, text features extracted from the titles from the trials are used to determine their effect on the Altmetric Attention Score (AAS) directly. Two classes for the AAS value are produced using the value of the 75th percentile as a threshold. This value is chosen to separate abnormally high scores from typical scores. Some ensemble methods are also used such as Random Forest, bagging, adaptive boosting, and gradient boosting which we believe have never been tried in the prior projects on this dataset.

2. Related Work

Similar research to this project can be divided into two main categories. The first area of research is the study of clinical trials data in order to improve trial accrual or to improve citation counts. In a study to construct a model to predict the number of participants in cancer clinical trials, Iruku et al. [3]. found that industrial sponsored trials had a higher ability to attract an ideal number of participants than institutionally sponsored trials. Also, their model showed that trials attracted more participants if the trial was done in a metastatic setting, or the trial was done by a local PI. Studies by Lara Jr. et al. found that for cancer trials conducted in California the socioeconomic status of possible participants affected their likelihood to sign up to participate [4, 5]. Thiele et al. constructed a model using a Random Forest classifier to determine linkages between 32,106 German clinical trials from ClinicalTrials.gov [6]. They found a decrease in the number of clinical trials in Germany over time and discussed the use of natural language processing (NLP) of clinical trial titles as a potential avenue of study to classify studies more accurately. Mike Thewall and

Kayvan Kousha studied whether studies cited on ClinicalTrials.gov have higher citation counts [7]. In the study, they used the four high impact general medical journals BMJ, the New England Journal of Medicine, The Lancet, and JAMA. They found that there was in fact a positive correlation.

The second area of research is the study of the relationship between Altmetrics (social media attention) and literary impact in scientific journals. Finch et al. studied how the number of online mentions on Twitter are correlated with the publication year and journal impact factor in the field of ornithology [8]. They found positive correlations between the year and Altmetric score, the journal impact factor and the Altmetric score, and the Altmetric score and the number of citations of the studies. Wang et al. found that journals with accounts on Twitter had higher Altmetric Attention Scores (AAS) in the field of neurosurgery [9]. Muñoz-Velandia et al. found similarly that having a Twitter account increased the h-index of an author in endocrinology journals [10]. Veysel Suzan M.D. and Damla Unal M.D. studied the correlations between citation number and AAS for the top 50 cited malnutrition journal articles and found a positive correlation [11]. Hayon et al. found similar correlation between Twitter mentions and citations for the field of urology [12]. Likewise, Daniel J. Lehane and Colin S. Black studied the relationship between Altmetric score and the citation count in high impact critical care medical journals with similar results [13]. However, they did find that articles with high AAS were less likely to have a high citation count. This shows that more work would need to be done to determine the cause of this difference. Anderson et al. studied 818 popular sports medicine articles and found positive correlations between scientific citations and media mentions on Facebook and Twitter [14]. Erskine et al. found beyond just studying the number of Twitter mentions that the type of the mention was also important [15] They found that tweets with article links had the lowest mentions. Links to internet journal clubs talking about articles had the greatest number of mentions.

3. Datasets Used

Two datasets were used for this project. The first dataset is the Altmetric Clinical Trials dataset [16] which can be downloaded from the Altmetric website. It contains 50,330 clinical trials with 45 features. The key features this dataset contains include the Altmetric Scores, the Title, the publication date, various mention types (news, blog, Twitter, peer review, Facebook, Google+, LinkedIn, Reddit, Pinterest, and F1000), the number of Mendeley readers, and the number of Dimensions citations. The second dataset is the Dimensions Clinical Trials dataset [17] which can be obtained from the Dimensions.ai website. It contains 32,728 clinical trials with 57 features. Key features of interest in this dataset include Rank, Trial ID, Title, Abstract, Start Year, Completion Year, Phase, Conditions, Gender, Sponsors/Collaborators, City of Sponsor/Collaborator, State of Sponsor/Collaborator, Collaborating Funders, and 6 category features (FOR (ANZSRC), RCDC, HRCS HC, HRCS RAC, ICRP Cancer Types, and ICRP CSO). The 6 categories of interest correspond to:

- FOR (ANZSRC): Field of Research using the Australian and New Zealand Standard Research Classification
- RCDC: Research, Condition, and Disease Categorization from the NIH
- HRCS HC: Health Research Classification System: Health Categories
- HRCS RAC: Health Research Classification System: Research Activities
- ICRP Cancer Types: International Cancer Research Partnership: Cancer Types
- ICRP CSO: International Cancer Research Partnership: Common Scientific Outline

We an inner join was used to combine the two datasets using the “Trial ID” column from the Dimensions dataset and the “National Clinical Trial ID” column from the altmetric dataset as keys so that the text in the clinical trial titles and abstracts could be used with the altmetric scores.

4. Methodology

First data preprocessing was done, and feature engineering was performed to generate new features. Then several general-purpose machine learning classifiers as well as ensemble methods were applied on the features extracted.

4.1 Data Preprocessing and Numeric Feature Engineering

The data preparation for this project was done using Pandas, a popular Python package. Some new numerical features were generated with the help of other features.

4.1.1 Dimensions Dataset

First, the columns which were completely empty were determined by finding the number of missing values. These columns were then removed. A list of the remaining columns was then used to subset the dataset. Next, a new feature “Numeric Phase” was then engineered by converting the phase number of the clinical trial to an integer. For trials that have two phases at once, the higher numbered phase was used. Another feature “Number of Phases” was constructed by determining the number of simultaneous phases each trial had.

4.1.2 Altmetric Clinical Trials Dataset

Similar to how the Dimensions dataset was processed, the columns that were completely empty were removed by subsetting the corresponding Pandas DataFrame. A statistical description of the resulting dataset was used to determine the 25%, 50% and 75% percentiles of the Altmetric Attention Score (AAS). A histogram of the AAS values for the clinical trials shows that there is significant imbalance in the data with most scores being less than 100 as shown in Figure 1 below. Given the imbalance of these scores, only two class labels were used rather than the originally intended four class labels corresponding to the 4 quartiles. This new class label “AAS Two Class” was then engineered with Class 0 (Group 0) corresponding to trials with AAS values greater than the 75% percentile or abnormally high values and Class 1 (Group 1) corresponding to trials with values less than or equal to the 75% percentile. In this case, the cutoff value used was the 75% percentile in the resultant dataset from joining both datasets which had a value of 7 rather than 5. A feature “Altmetric Title Length” was then constructed by taking the character length of the corresponding values found in the “Title” column for each trial. Finally, a “Total Mentions” feature was constructed as the sum of the media mentions for the corresponding trials. The resulting Pandas DataFrame was then saved as a CSV file.

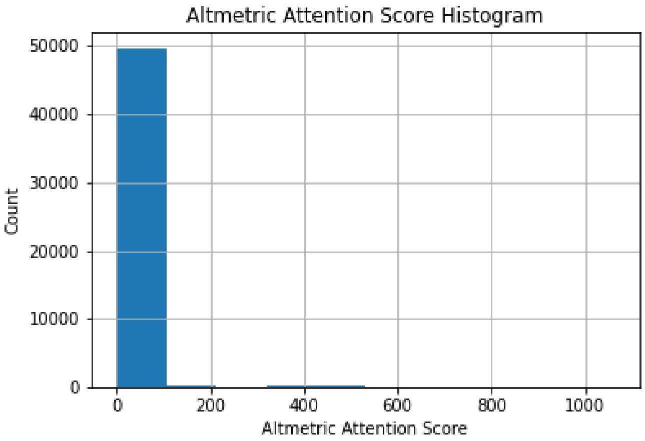


Figure 1: **Altmetric Attention Score Histogram**

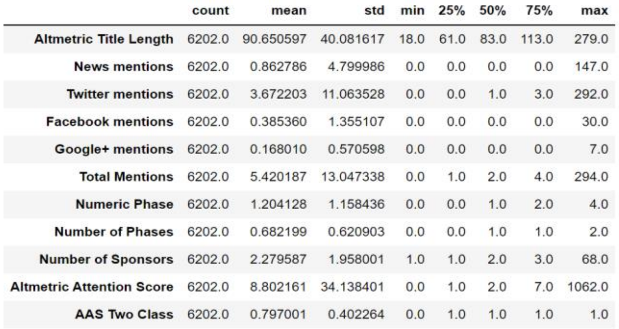
4.1.3 Joined Dataset and Category Feature Engineering

With both the Dimensions and Altmetric datasets processed, an inner join was performed by using the ‘Trial ID’ attribute from the Dimensions dataset and the ‘National Clinical Trial ID’ from the Altmetric dataset. A new feature corresponding to the number of sponsors for each trial was then added. The resulting dataset contained 6202 entries with 46 columns. CSV files were then produced for each of the different category types (FOR (ANZSRC), RCDC, HRCS HC, HRCS RAC, ICRP Cancer Types, and ICRP CSO) by removing entries from the joined dataset that were missing values for the corresponding categories. These could be used in future projects.

4.1.4 Descriptive Statistical Studies of Numerical Features

Table 1 shows the descriptive statistics for the numerical features used for classification in the joined dataset. It was found that, on average, of the media mention types studied, clinical trials are mentioned more on Twitter (3.67) than either the news (0.863), Facebook (0.385), or Google+ (0.168). The social media site that showed the lowest number of mentions was found to be Google+ with an average value of 0.168 mentions for a specific clinical trial. The fact Twitter was found to be used for mentions more than Google+ is likely due to the relative popularities of the two sites. The means of the numeric phase and number of phases were found to be 1.20 and 0.682, respectively. This tells us that the average clinical trial in the dataset is likely in Phase 1 and is not involving multiple phases simultaneously. It was also found that clinical trials usually have around 2 sponsors. (2.28) Since the 25th, 50th, and 75th percentiles are all 0, this tells us that most clinical trials are never mentioned in the news. Since the mean is 0.863, some trials that are mentioned likely correspond to significant treatments.

Table 1: **Descriptive Statistics of Numerical Classification Features**



	count	mean	std	min	25%	50%	75%	max
Altmetric Title Length	6202.0	90.650597	40.081617	18.0	61.0	83.0	113.0	279.0
News mentions	6202.0	0.862786	4.799986	0.0	0.0	0.0	0.0	147.0
Twitter mentions	6202.0	3.672203	11.063528	0.0	0.0	1.0	3.0	292.0
Facebook mentions	6202.0	0.385360	1.355107	0.0	0.0	0.0	0.0	30.0
Google+ mentions	6202.0	0.168010	0.570598	0.0	0.0	0.0	0.0	7.0
Total Mentions	6202.0	5.420187	13.047338	0.0	1.0	2.0	4.0	294.0
Numeric Phase	6202.0	1.204128	1.158436	0.0	0.0	1.0	2.0	4.0
Number of Phases	6202.0	0.682199	0.620903	0.0	0.0	1.0	1.0	2.0
Number of Sponsors	6202.0	2.279587	1.958001	1.0	1.0	2.0	3.0	68.0
Altmetric Attention Score	6202.0	8.802161	34.138401	0.0	1.0	2.0	7.0	1062.0
AAS Two Class	6202.0	0.797001	0.402264	0.0	1.0	1.0	1.0	1.0

After importing the joined dataset, a correlation heatmap was produced for the “Start Year”, “Completion Year”, “Blog mentions”, “Policy mentions”, “Weibo mentions”, “Wikipedia mentions”, “Q&A mentions”, “Video mentions”, “Altmetric Title Length”, “News mentions”, “Twitter mentions”, “Facebook mentions”, “Google+ mentions”, “Total Mentions”, “Numeric Phase”, “Number of Phases”, “Number of Sponsors”, and the “Altmetric Attention Score” features. As shown in Figure 2, the “Altmetric Attention Score” was found to be positively correlated with “Blog mentions” (ρ = 0.43), “News mentions” (ρ = 0.93), “Twitter mentions” (ρ = 0.20), and “Total Mentions” (ρ = 0.55). “Weibo mentions” and “Q&A mentions” were both found to have no correlation with the Altmetric Attention Score. There was a correlation between the number of simultaneous phases and the title length (ρ = 0.30), but this is likely due to the word “phase” being used multiple times in the title and doesn’t add relevant information for classification. Thus, the News mentions, Total mentions, Twitter mentions, Facebook mentions, and Google+ mentions were then used for model classification using numeric features. The Blog mentions feature was not used because of its high correlation with the News mentions feature. Numeric Phase, Number of Phases, Altmetric Title Length, and Number of Sponsors were added as new features.

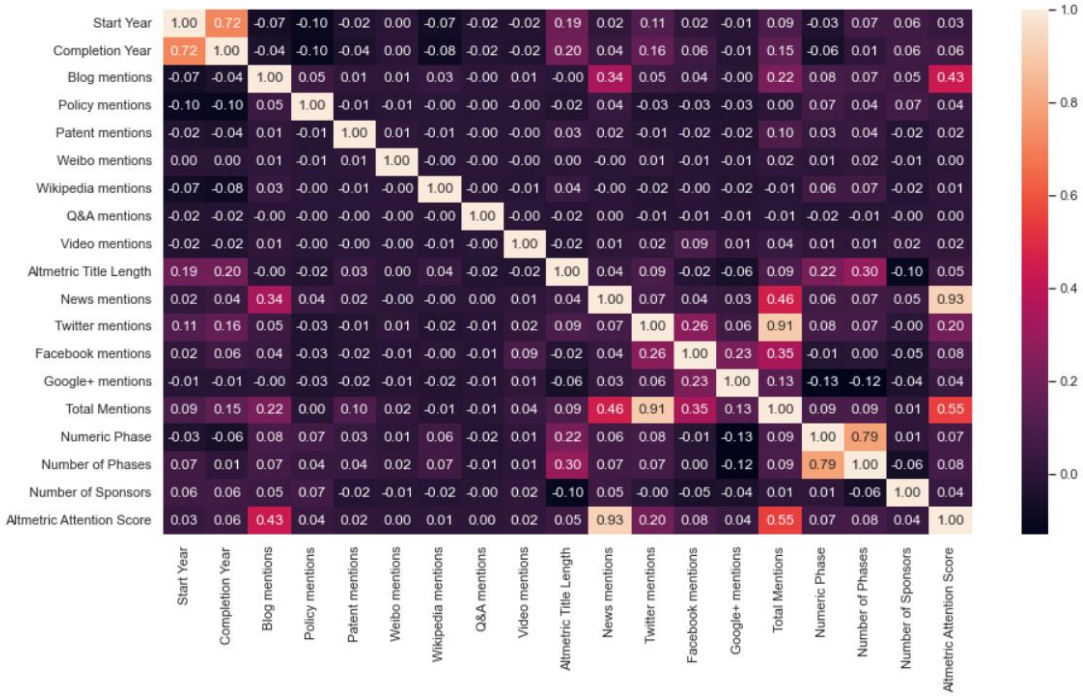


Figure 2: **Joined Dataset Variable Correlations**

4.2 Text Data Pre-processing and Feature Extraction

Before applying the Machine Learning classifiers, the text features need to be converted in numerical forms. An easy approach to do that is to use the unique words in a text as features and the frequency of that word as the value of the respective feature. However, before finding the unique meaningful words from a text, the following steps were performed:

1) **HTML parsing:** Usually this type of parsing helps to pull data out of HTML and XML files. Since most of the texts in the dataset are possibly pulled from HTML documents, this parsing was applied. It extracts data in a hierarchical and more readable format. For this experiment this step was performed with the help of BeautifulSoup [18] Python library.

2) **Removing special characters:** Text data can often contain special characters which do not bear much meaning or significance. Python regular expression library was used to remove all the special characters so that the texts contain letters and numbers only.

3) **Tokenization:** Tokenization is a process to split texts into words. This was done using word_tokenize() function of word_tokenize library which is a part of Natural Language ToolKit (NLTK) [19].

4) **Converting to lower-case:** At this step all the letters in the words were converted to their lower cases. This is required so that the count of words having the same spelling and different cases become uniform. For example, ‘Learn’ and ‘learn’ are the same word but they can be counted twice unless we convert the upper-case letters to their lower cases before counting. A python function lower() was used to convert the upper case letters into their lower cases.

5) **Stopwords removal:** Words that are commonly used in our language for supporting other words but have less significant meaning are considered as stop words. Stop words are filtered out for Natural Language Processing for saving memory and processing time. Search engines are examples where stop words are internally removed before processing the query [20]. In this experiment, stop words listed for the English language in NLTK were removed from the texts. Some of the words listed there were: ‘I’, ‘me’, ‘my’, ‘myself’, ‘we’, ‘it’, ‘what’, ‘is’ etc.

6) **Stemming and Lemmatization:** Stemming is done to remove the last few letters from a word to convert it to another meaningful word while Lemmatization also considers context to convert it to the meaningful base word. For example, after lemmatization the word ‘selection’ will be converted into the word ‘select’. PorterStemmer() and WordNetLemmatizer() python functions from NLTK were used for this step.

Once the above steps were applied on the dataset, the data was ready for text feature extraction. In this experiment text feature extraction was done using the following model.

Term Frequency–Inverse Document Frequency (TF-IDF): TF-IDF is a popular model for text feature extraction which has quite successful applications in text related problems [21]. The TF-IDF score of a word increases with the number of times the word appears in the document and offset by the number of documents containing the same word. It is used such frequently that 83% of text-based recommender systems for digital libraries use this [22]. TF-IDF score of a word is calculated using the following equation:

$$tfidf(t, d, D) = tf(t, d).idf(t, D)$$

Where, *t* is the term, *d* is the document and *D* is the total number of documents in the corpus.

4.3 Data Splitting and Model Production

The joined dataset was then split into feature data and label data. Next, the data was split into 80% training data and 20% test data. The test data was used as a holdout set while the training data was used for 10-fold cross validation. Finally, selected machine learning classifiers from different categories were applied to the joined dataset and their performances were evaluated. Some general purpose and most commonly used classifiers such as Decision Tree (DT), Logistic Regression (LR), Naive Bayes (NB), and Support Vector Machine (SVM) were primarily used. Some ensemble methods [23] were also used such as Random Forest (RF) which is a tree-based model, BaggingClassifier from bagging category, AdaBoostClassifier from adaptive boosting category, and XGBoost from gradient boosting category.

Bagging was used with Decision Tree, SVM, or LR as the base classifier. Random Forest and Adaboost were used just with the Decision Tree classifier. Models with XGBoost were fitted with either the DMatrix classifier or XGBClassifier using the binary: logistic objective hyperparameter.

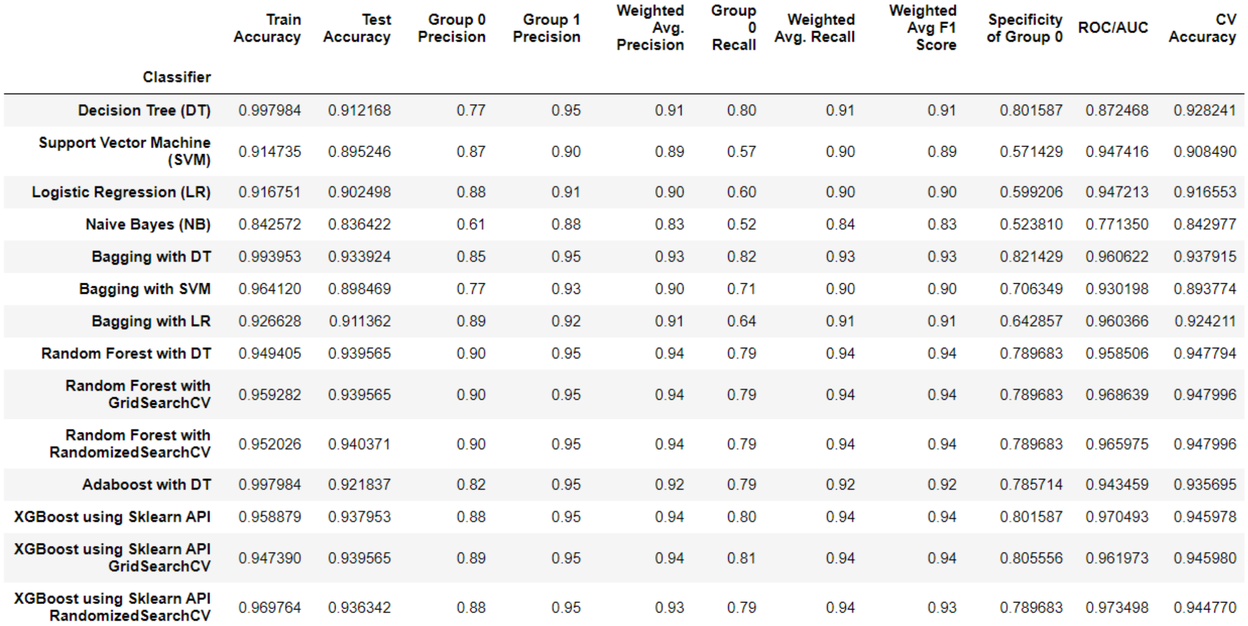
The ensemble methods were chosen because the accuracies mentioned in the prior project on the same dataset were not found satisfactory, and these methods often improve the performance rather than using any standalone classifier. Most of the classifiers used in this project are available in Scikit-learn Python library except the XGBoost is in xgboost library [24]. An XGBoost classifier which uses the Scikit-learn style API was used when comparing models. RandomSearchCV as well as GridSearchCV cross validation and tuning classes which are available in Scikit-learn Python library were used to tune the hyperparameters of Random Forest and XGBoost Scikit-learn API classifiers as these techniques have been proven effective in increasing model accuracy. The max_depth (1,2,4,6,8,10) and n_estimators (100,200,300,1000) hyperparameters were tuned for the Random Forest models using the numeric features, and the learning_rate (0.05 to 1.05), n_estimators (200), and subsample (0.05 to 1.05) hyperparameters were tuned for the XGBClassifier. Between these two techniques, RandomSearchCV usually outperforms GridSearchCV [25].

To evaluate our models, first, confusion matrices were constructed for each model. Next, training accuracies, test accuracies, 10-fold cross validation accuracies, precision for the high AAS valued class (Group 0), weighted average precision, Group 0 recall, weighted average recall, weighted average F1-score, Group 0 specificity, and ROC AUC values were calculated. Finally, ROC curves were plotted to compare models.

5. Results

This section presents the experimental results found upon applying different machine learning models on the numerical features as well as the text features. Results were categorized based on the group of classifiers such as general-purpose classifiers, ensemble methods and so on.

Table 2: **Evaluation Results of Numerical-Based Models**



Classifier	Train Accuracy	Test Accuracy	Group 0 Precision	Group 1 Precision	Weighted Avg. Precision	Group 0 Recall	Weighted Avg. Recall	Weighted Avg F1 Score	Specificity of Group 0	ROC/AUC	CV Accuracy
Decision Tree (DT)	0.997984	0.912168	0.77	0.95	0.91	0.80	0.91	0.91	0.801587	0.872468	0.928241
Support Vector Machine (SVM)	0.914735	0.895246	0.87	0.90	0.89	0.57	0.90	0.89	0.571429	0.947416	0.908490
Logistic Regression (LR)	0.916751	0.902498	0.88	0.91	0.90	0.60	0.90	0.90	0.599206	0.947213	0.916553
Naive Bayes (NB)	0.842572	0.836422	0.61	0.88	0.83	0.52	0.84	0.83	0.523810	0.771350	0.842977
Bagging with DT	0.993953	0.933924	0.85	0.95	0.93	0.82	0.93	0.93	0.821429	0.960622	0.937915
Bagging with SVM	0.964120	0.898469	0.77	0.93	0.90	0.71	0.90	0.90	0.706349	0.930198	0.893774
Bagging with LR	0.926628	0.911362	0.89	0.92	0.91	0.64	0.91	0.91	0.642857	0.960366	0.924211
Random Forest with DT	0.949405	0.939565	0.90	0.95	0.94	0.79	0.94	0.94	0.789683	0.958506	0.947794
Random Forest with GridSearchCV	0.959282	0.939565	0.90	0.95	0.94	0.79	0.94	0.94	0.789683	0.968639	0.947996
Random Forest with RandomizedSearchCV	0.952026	0.940371	0.90	0.95	0.94	0.79	0.94	0.94	0.789683	0.965975	0.947996
Adaboost with DT	0.997984	0.921837	0.82	0.95	0.92	0.79	0.92	0.92	0.785714	0.943459	0.935695
XGBoost using Sklearn API	0.958879	0.937953	0.88	0.95	0.94	0.80	0.94	0.94	0.801587	0.970493	0.945978
XGBoost using Sklearn API GridSearchCV	0.947390	0.939565	0.89	0.95	0.94	0.81	0.94	0.94	0.805556	0.961973	0.945980
XGBoost using Sklearn API RandomizedSearchCV	0.969764	0.936342	0.88	0.95	0.93	0.79	0.94	0.93	0.789683	0.973498	0.944770

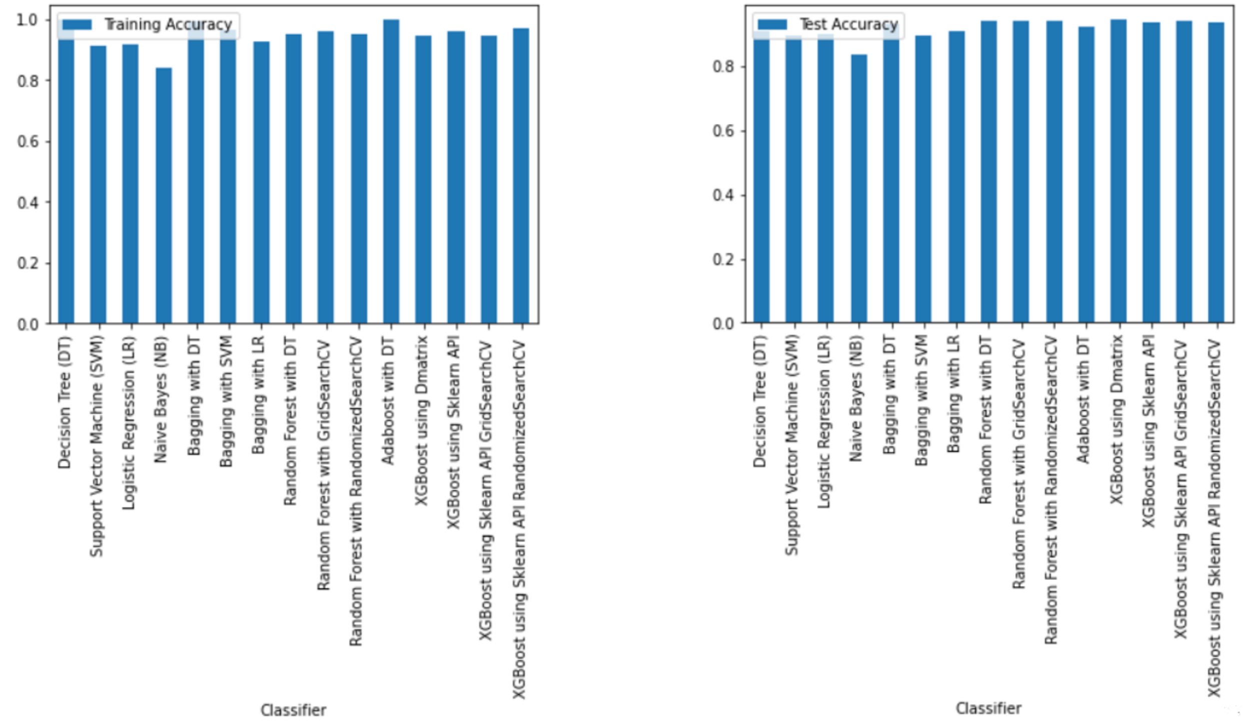


Figure 3: **Training and Test Accuracies for Numerical-Based Models**

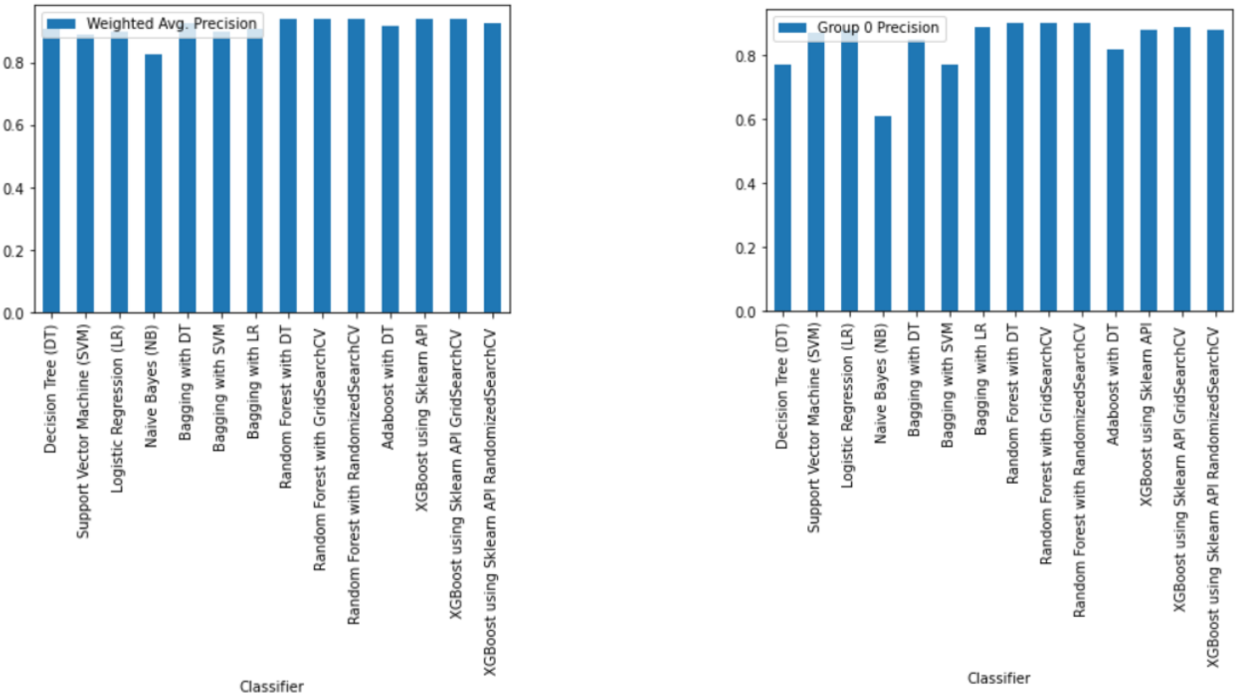


Figure 4: **Weighted Avg. Precision and Group 0 Precision for Numerical-Based Models**

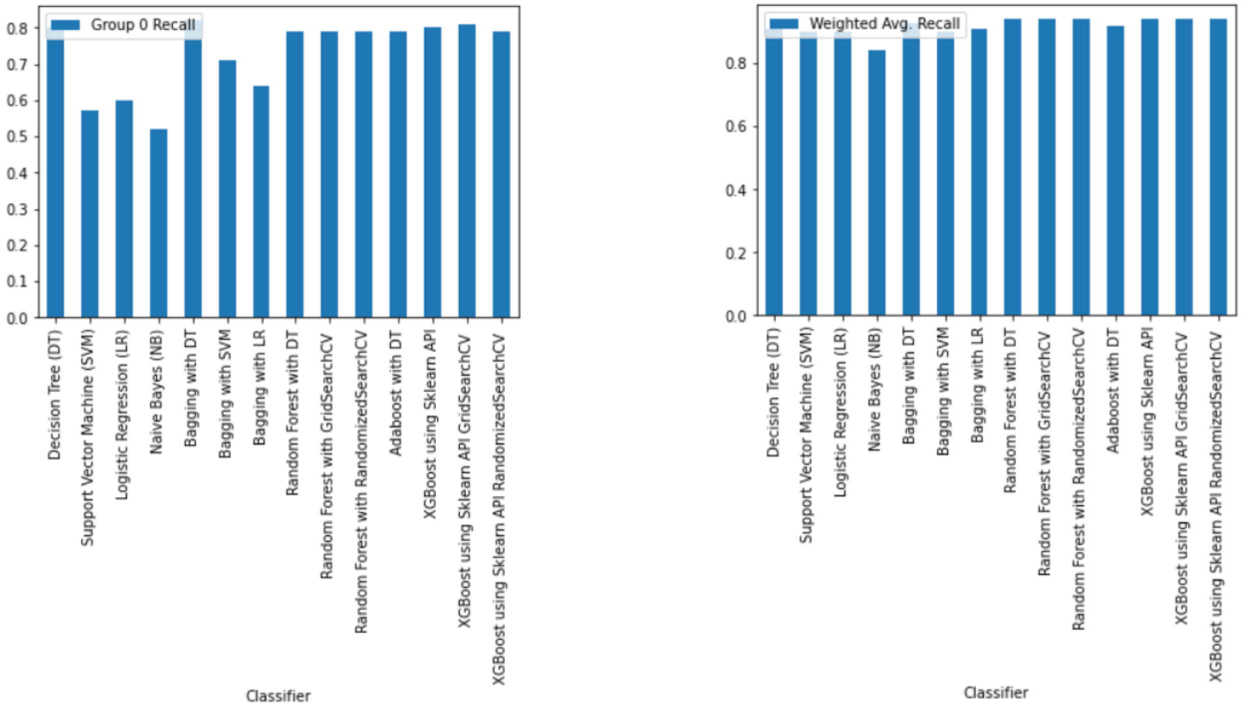


Figure 5: **Group 0 Recall and Weighted Avg. Recall for Numerical-Based Models**

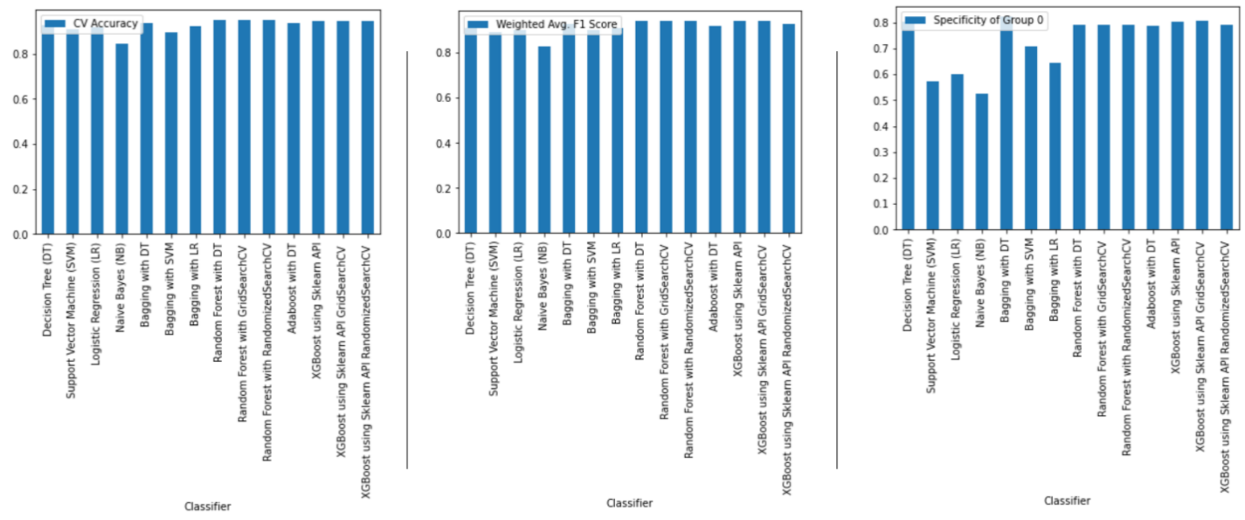


Figure 6: **Cross Validation Accuracy, Weight Avg. F1-Score, and Specificity of Group 0 for Numerical-Based Models**

5.1 General Classifiers

Among the general and commonly used classifiers Decision Tree (DT), Naive Bayse (NB), Support Vector Machine (SVM), and Logistic Regression (LR) were applied. Results found for the numerical models and the text models are as follows:

5.1.1 Results for Numerical-Based Models using General Classifiers

Table 2 shows the results for the numerical-based models. As shown in Table 2, the Decision Tree classifier gave values of 0.998 for the training accuracy, 0.912 for the test accuracy, 0.77 for the group 0 precision, 0.91 for the weighted average precision, 0.80 for the group 0 recall, 0.91 for the weighted average recall, 0.91 for the weighted average F1-score, 0.802 for the specificity of group 0, 0.872 for the area under the ROC curve, and 0.928 for the mean 10-fold cross validation accuracy. The Support Vector Machine classifier gave values of 0.915 for the training accuracy, 0.895 for the test accuracy, 0.87 for the group 0 precision, 0.89 for the weighted average precision, 0.57 for the group 0 recall, 0.90 for the weighted average recall, 0.89 for the weighted average F1-score, 0.571 for the specificity of group 0, 0.947 for the area under the ROC curve, and 0.908 for the mean 10-fold cross validation accuracy. The Logistic Regression classifier gave values of 0.917 for the training accuracy, 0.902 for the test accuracy, 0.880 for the group 0 precision, 0.90

for the weighted average precision, 0.60 for the group 0 recall, 0.90 for the weighted average recall, 0.90 for the weighted average F1-score, 0.599 for the specificity of group 0, 0.947 for the area under the ROC curve, and 0.916 for the mean 10-fold cross validation accuracy. The Naive Bayes classifier gave values of 0.842 for the training accuracy, 0.836 for the test accuracy, 0.61 for the group 0 precision, 0.83 for the weighted average precision, 0.52 for the group 0 recall, 0.84 for the weighted average recall, 0.83 for the weighted average F1-score, 0.524 for the specificity of group 0, 0.771 for the area under the ROC curve, and 0.843 for the mean 10-fold cross validation accuracy.

5.1.2 Results for Text-Based Models using General Classifiers

Upon applying the same set of classifiers on the text features, the minimum training accuracy was found 0.80 for Decision Tree while the maximum was 0.99 for Logistic Regression. On the other hand, the minimum test accuracy was found 0.72 for Logistic Regression where the maximum was 0.79 for Support Vector Machine classifier with linear kernel. Among the general classifiers used Support Vector Machine (SVM) had the highest precision score of 0.74. SVM also showed the highest recall score of 0.99 among the classifiers from this group. The F1 score was found almost similar for all the classifiers. The cross-validation accuracy for SVM was 0.79 which was same for Decision Tree. The area under the ROC curve for SVM was 0.63 which was not the highest but the second highest. The highest value found for the area under ROC curve was found for Naive Bayes classifier which was 0.64 slightly better than the SVM. Based on the overall performance SVM with the linear kernel showed consistently good results for all the metrics considered.

5.2 Results for Ensemble Methods

A number of ensemble methods from each of its category (e. g. Bagging, Boosting) were applied and the results found are presented as follows:

5.2.1 Results for Numerical-Based Models using Bagging

For the numerical-based models, the bagging method with the Decision Tree classifier gave values of 0.994 for the training accuracy, 0.934 for the test accuracy, 0.85 for the group 0 precision, 0.93 for the weighted average precision, 0.82 for the group 0 recall, 0.93 for the weighted average recall, 0.93 for the weighted average F1-score, 0.821 for the specificity of group 0, 0.961 for the area under the ROC curve, and 0.938 for the mean 10-fold cross validation accuracy. The bagging method with the Support Vector Machine classifier gave values of 0.964 for the training accuracy, 0.898 for the test accuracy, 0.77 for the group 0 precision, 0.90 for the weighted average precision, 0.71 for the group 0 recall, 0.90 for the weighted average recall, 0.90 for the weighted average F1-score, 0.706 for the specificity of group 0, 0.930 for the area under the ROC curve, and 0.894 for the mean 10-fold cross validation accuracy. The bagging method with the Logistic Regression classifier gave values of 0.927 for the training accuracy, 0.911 for the test accuracy, 0.89 for the group 0 precision, 0.92 for the weighted average precision, 0.64 for the group 0 recall, 0.91 for the weighted average recall, 0.91 for the weighted average F1-score, 0.643 for the specificity of group 0, 0.960 for the area under the ROC curve, and 0.924 for the mean 10-fold cross validation accuracy.

5.2.2 Results for Text Based Models using Bagging

Bagging classifier was applied with three base classifiers – Logistic Regression, Decision Tree, and Support Vector Machine. Among all these variants, Bagging with the SVM performed consistently better than the others for most of the metrics with a precision of 0.84 and a recall of 0.8. Its F1 score was 0.71 and the area under the ROC curve is 0.67. In this group of classifiers, the largest area under the ROC curve was found 0.67 for the Bagging with Logistic Regression.

5.2.3 Results for Numerical-Based Models using Random Forest

For the numerical-based models, the Random Forest method with the Decision Tree classifier and without hyperparameter gave values of 0.949 for the training accuracy, 0.940 for the test accuracy, 0.90 for the group 0 precision, 0.94 for the weighted average precision, 0.79 for the group 0 recall, 0.94 for the weighted average recall, 0.94 for the weighted average F1-score, 0.790 for the specificity of group 0, 0.958 for the area under the ROC curve, and 0.948 for the mean 10-fold cross validation accuracy. When the model was tuned with grid search, the optimal model (max_depth = 8, n_estimators = 300) gave values of 0.959 for the training accuracy, 0.940 for the test accuracy, 0.90 for the group 0 precision, 0.95 for the weighted average precision, 0.79 for the group 0 recall, 0.94 for the weighted average recall, 0.94 for the weighted average F1-score, 0.790 for the specificity of group 0, 0.969 for the area under the ROC curve, and 0.948 for the mean 10-fold cross validation accuracy. When the model was tuned with randomized search, the optimal hyperparameter values (max_depth = 6, n_estimators = 300) gave values of 0.952 for the training accuracy, 0.940 for the test accuracy, 0.90 for the group 0 precision, 0.94 for the weighted average precision, 0.79 for the group 0 recall, 0.94 for the weighted average recall, 0.94 for the weighted average F1-score, 0.790 for the specificity of group 0, 0.966 for the area under the ROC curve, and 0.948 for the mean 10-fold cross validation accuracy.

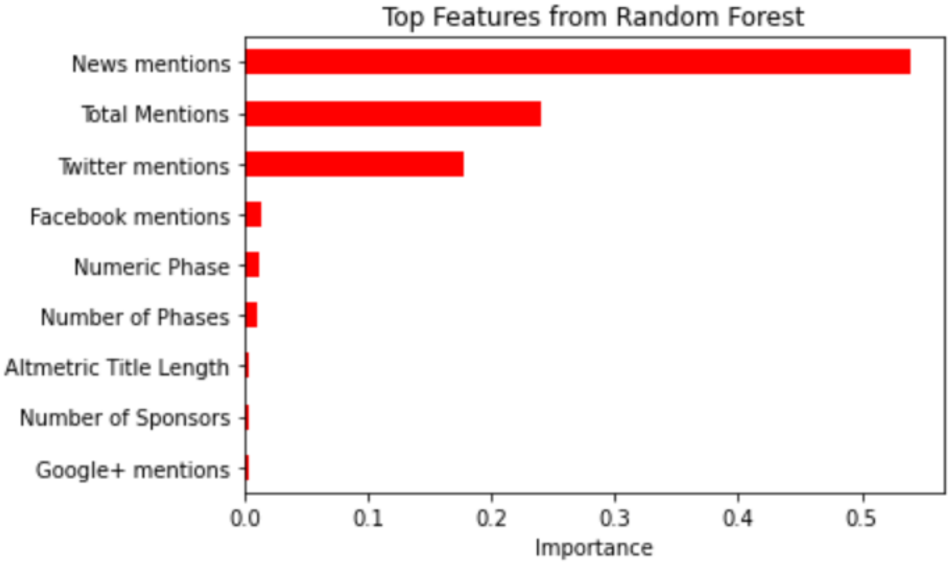


Figure 7: **Top Features from Random Forest**

Figure 7 shows the top features identified by the Random Forest model and they correspond to the fact found from the correlation shown in the heatmap (Figure 2).

5.2.4 Results for Text-Based Models using Random Forest

For this experiment, Random Forest classifier was applied with Decision Tree, GridSearchCV and RandomizedSearchCV. All of them showed very similar results for all the metrics considered except the area under the ROC curve. Their training accuracy was 0.79, test accuracy was 0.79, precision was 0.64, recall was 0.8, and F1 score was 0.71. The cross-validation accuracy was also found to be the same for all of them and that was 0.79. However, the area under the ROC curve found for all three of them were slightly different – Random Forest with Decision Tree had 0.64, Random Forest with GridSearchCV had 0.61 while Random Forest with RandomizedSearchCV had 0.65.

5.2.5 Results for Numerical-Based Models using Adaptive Boosting

For the numerical-based models, Adaboost using Decision Tree as the base classifier gave results of 0.998 for the training accuracy, 0.922 for the test accuracy, 0.82 for the group 0 precision, 0.92 for the weighted average precision, 0.79 for the group 0 recall, 0.92 for the weighted average recall, 0.92 for the weighted average F1-score, 0.786 for the specificity of group 0, 0.942 for the area under the ROC curve, and 0.936 for the mean 10-fold cross validation accuracy.

5.2.6 Results for Text-Based Models using Adaptive Boosting

Adaboost showed a training accuracy of 0.83 which is relatively higher than most of the other classifiers applied. It had a test accuracy of 0.77, cross-validation accuracy of 0.76, precision of 0.71, recall of 0.77, and a F1 score of 0.72. The area under the ROC curve for this classifier was 0.59.

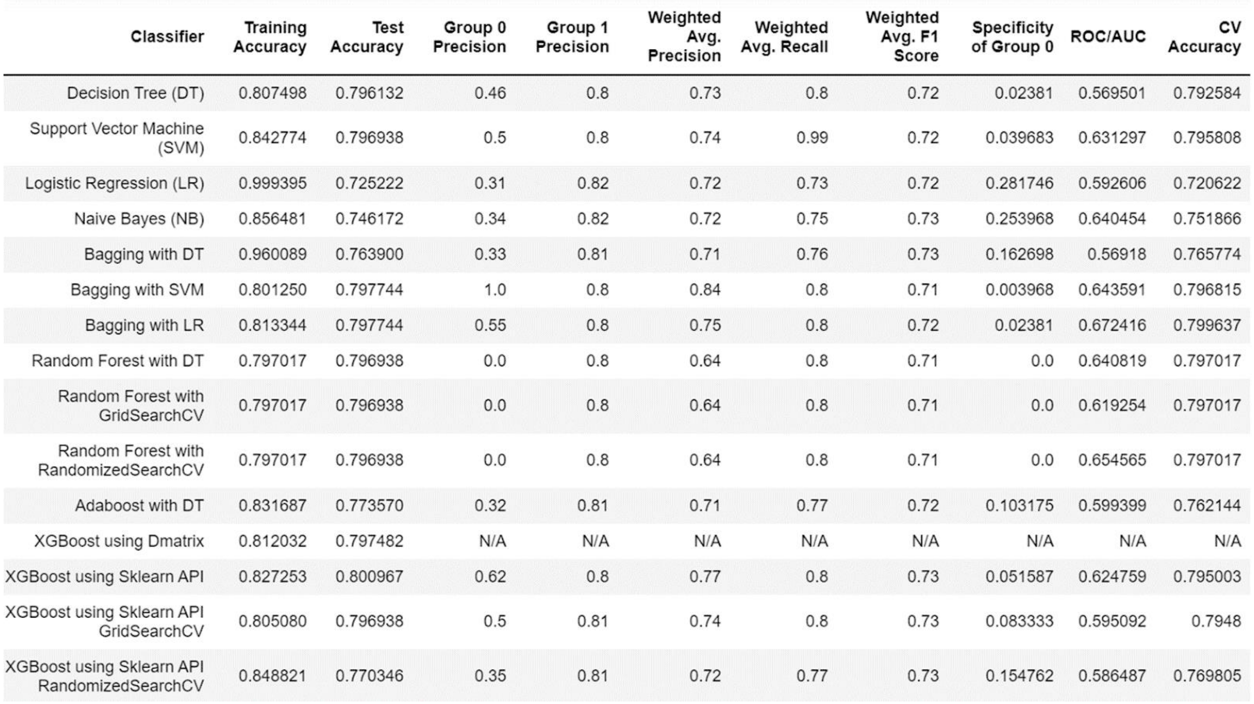
5.2.7 Results for Numerical-Based Models using Gradient Boosting

For the numerical-based models, the XGBoost classifier using the DMatrix, values of 0.948 for the training accuracy and 0.945 for the test accuracy. The remaining metrics were calculated for XGBoost when using the Scikit-learn API with and without hyperparameter tuning. Without tuning, the XGBoost classifier gave values of 0.959 for the training accuracy, 0.940 for the test accuracy, 0.88 for the group 0 precision, 0.94 for the weighted average precision, 0.80 for the group 0 recall, 0.94 for the weighted average recall, 0.94 for the weighted average F1-score, 0.802 for the specificity of group 0, 0.970 for the area under the ROC curve, and 0.946 for the mean 10-fold cross validation accuracy. When using grid search, the optimal classifier (subsample = 0.05, n_estimators = 200, learning_rate = 0.05) gave values of 0.948 for the training accuracy, 0.940 for the test accuracy, 0.89 for the group 0 precision, 0.94 for the weighted average precision, 0.81 for the group 0 recall, 0.94 for the weighted average recall, 0.94 for the weighted average F1-score, 0.806 for the specificity of group 0, 0.962 for the area under the ROC curve, and 0.946 for the mean 10-fold cross validation accuracy. When using randomized search, the optimal classifier (subsample = 0.8, n_estimators = 200, learning_rate = 0.05) gave values of 0.970 for the training accuracy, 0.936 for the test accuracy, 0.88 for the group 0 precision, 0.93 for the weighted average precision, 0.79 for the group 0 recall, 0.94 for the weighted average recall, 0.93 for the weighted average F1-score, 0.790 for the specificity of group 0, 0.973 for the area under the ROC curve, and 0.945 for the mean 10-fold cross validation accuracy.

5.2.8 Results for Text-Based Models using Gradient Boosting

XGBoost was applied using Sklearn API along with GridSearchCV and RandomizedSearchCV. Among them XGBoost with the RandomizedSearchCV had the highest training accuracy 0.84 though it had the lowest test accuracy of 0.77 compared to the others in this group. XGBoost using the Sklearn API without any of the GridSearchCV and RandomizedSearchCV showed the best performance for all other metrics with 0.77 precision, 0.8 recall, and 0.73 F1 score. The area under the ROC curve was found 0.62 for it.

Table 3: **Evaluation Results of Text-Based Models**



Classifier	Training Accuracy	Test Accuracy	Group 0 Precision	Group 1 Precision	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F1 Score	Specificity of Group 0	ROC/AUC	CV Accuracy
Decision Tree (DT)	0.807498	0.796132	0.46	0.8	0.73	0.8	0.72	0.02381	0.569501	0.792584
Support Vector Machine (SVM)	0.842774	0.796938	0.5	0.8	0.74	0.99	0.72	0.039683	0.631297	0.795808
Logistic Regression (LR)	0.999395	0.725222	0.31	0.82	0.72	0.73	0.72	0.281746	0.592606	0.720622
Naive Bayes (NB)	0.856481	0.746172	0.34	0.82	0.72	0.75	0.73	0.253968	0.640454	0.751866
Bagging with DT	0.960089	0.763900	0.33	0.81	0.71	0.76	0.73	0.162698	0.56918	0.765774
Bagging with SVM	0.801250	0.797744	1.0	0.8	0.84	0.8	0.71	0.003968	0.643591	0.796815
Bagging with LR	0.813344	0.797744	0.55	0.8	0.75	0.8	0.72	0.02381	0.672416	0.799637
Random Forest with DT	0.797017	0.796938	0.0	0.8	0.64	0.8	0.71	0.0	0.640819	0.797017
Random Forest with GridSearchCV	0.797017	0.796938	0.0	0.8	0.64	0.8	0.71	0.0	0.619254	0.797017
Random Forest with RandomizedSearchCV	0.797017	0.796938	0.0	0.8	0.64	0.8	0.71	0.0	0.654565	0.797017
Adaboost with DT	0.831687	0.773570	0.32	0.81	0.71	0.77	0.72	0.103175	0.599399	0.762144
XGBoost using Dmatrix	0.812032	0.797482	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
XGBoost using Sklearn API	0.827253	0.800967	0.62	0.8	0.77	0.8	0.73	0.051587	0.624759	0.795003
XGBoost using Sklearn API GridSearchCV	0.805080	0.796938	0.5	0.81	0.74	0.8	0.73	0.083333	0.595092	0.7948
XGBoost using Sklearn API RandomizedSearchCV	0.848821	0.770346	0.35	0.81	0.72	0.77	0.73	0.154762	0.586487	0.769805

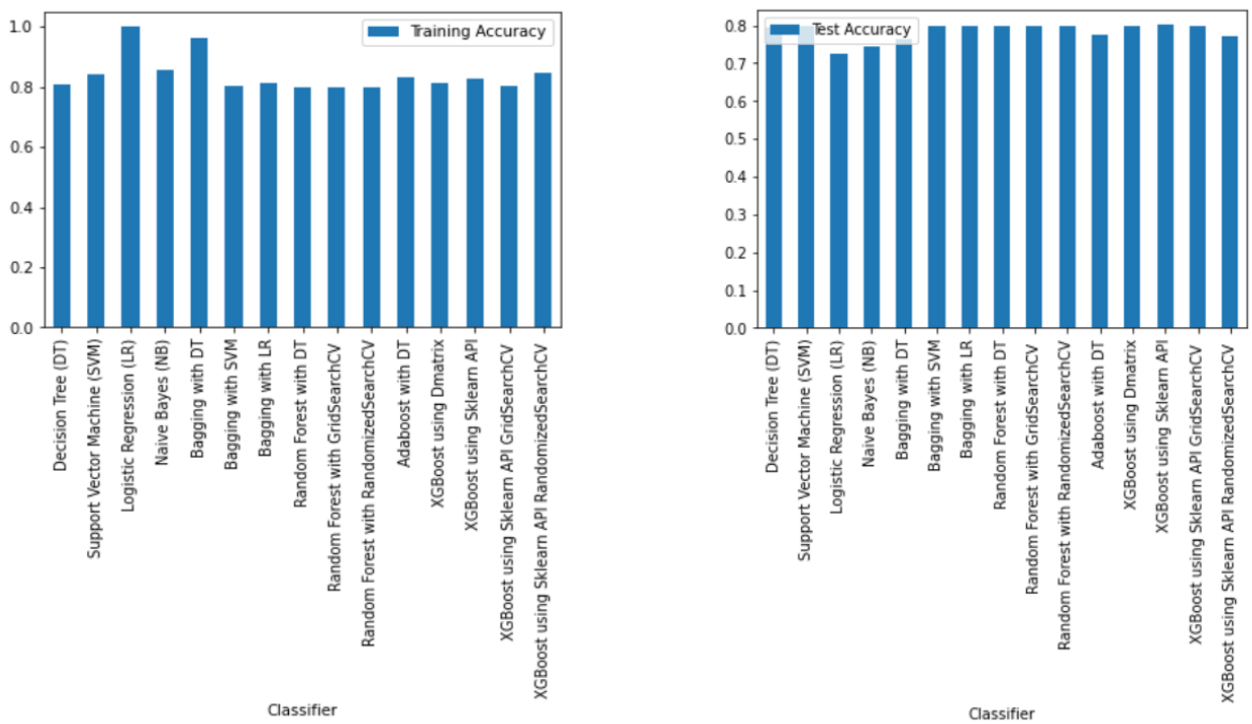


Figure 8: **Training and Test Accuracies for Text-Based Models**

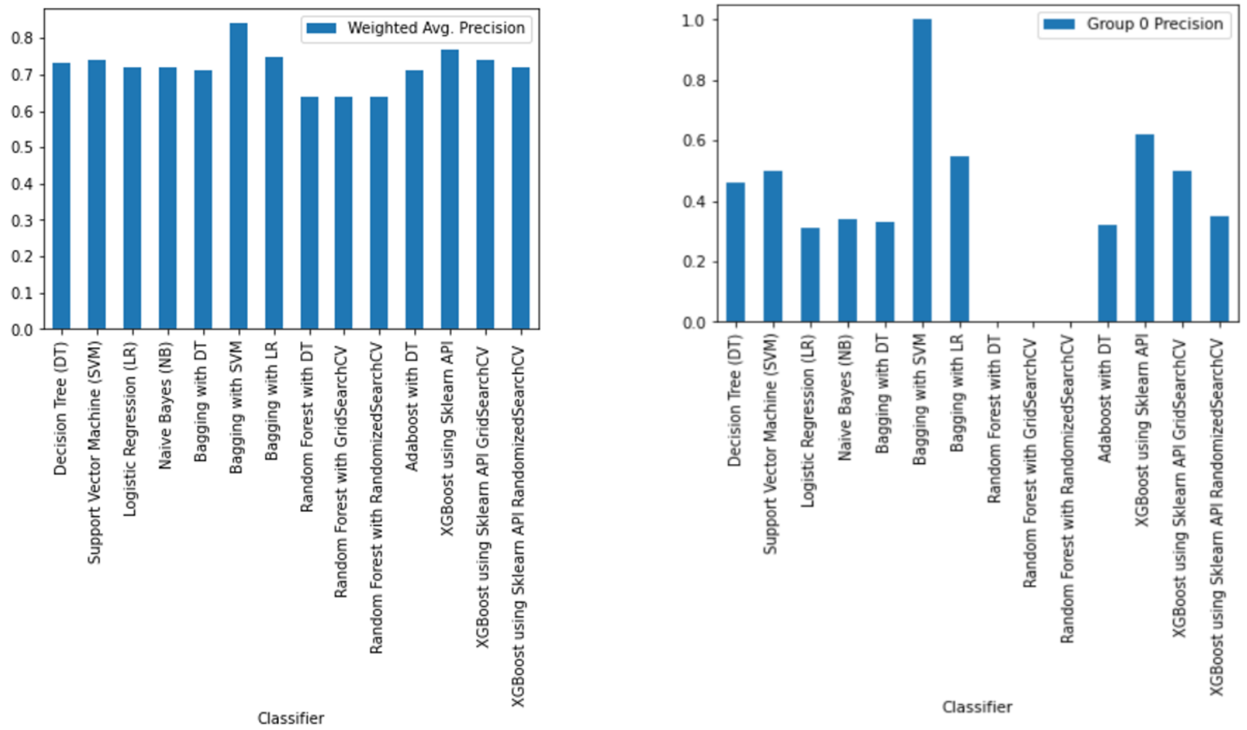


Figure 9: **Weighted Avg. Precision and Group 0 Precision for Text-Based Models**

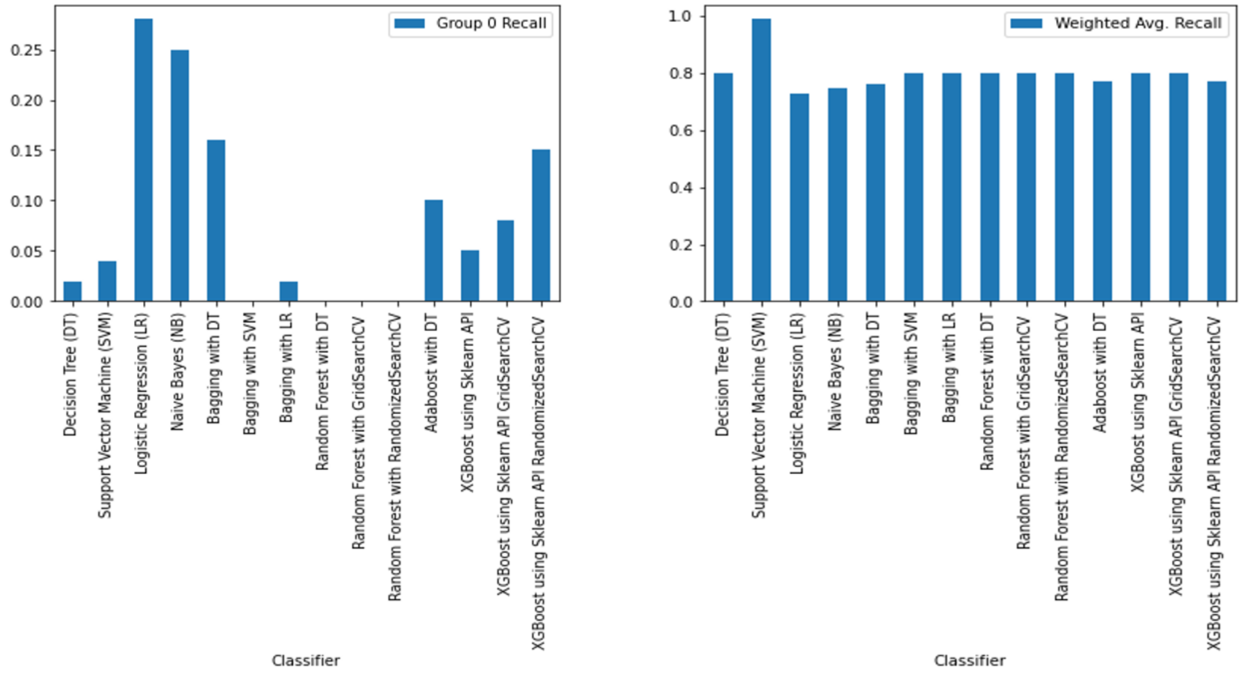


Figure 10: **Group 0 Recall and Weighted Avg. Recall for Text-Based Models**

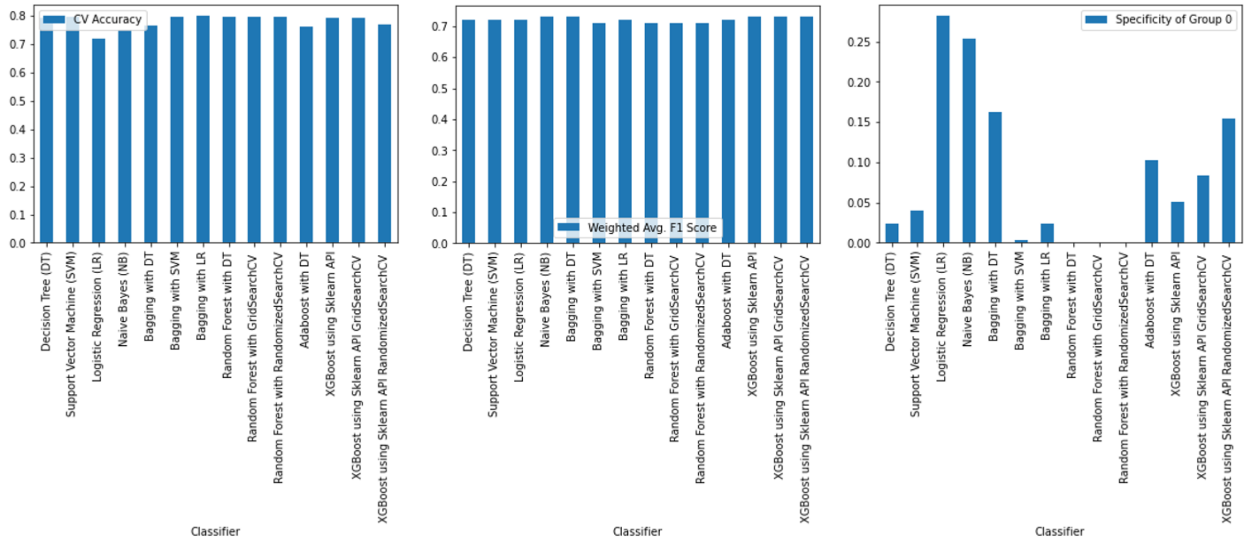


Figure 11: **Cross Validation Accuracy, Weight Avg. F1-Score, and Specificity of Group 0 for Text-Based Models**

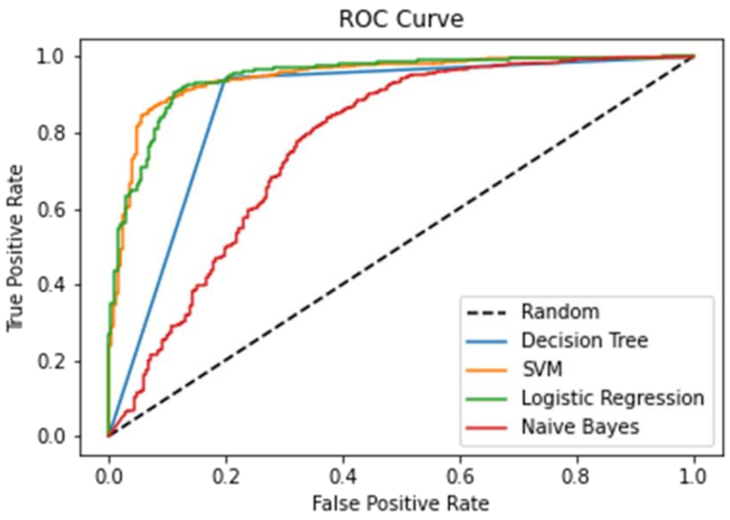


Figure 12: **ROC Curves for Numerical-Based General Classifiers**

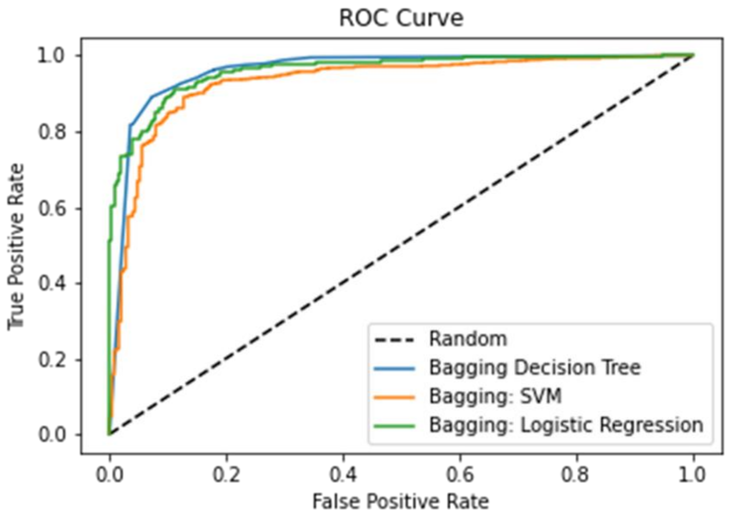


Figure 13: **ROC Curves for Numerical-Based Bagging Classifiers**

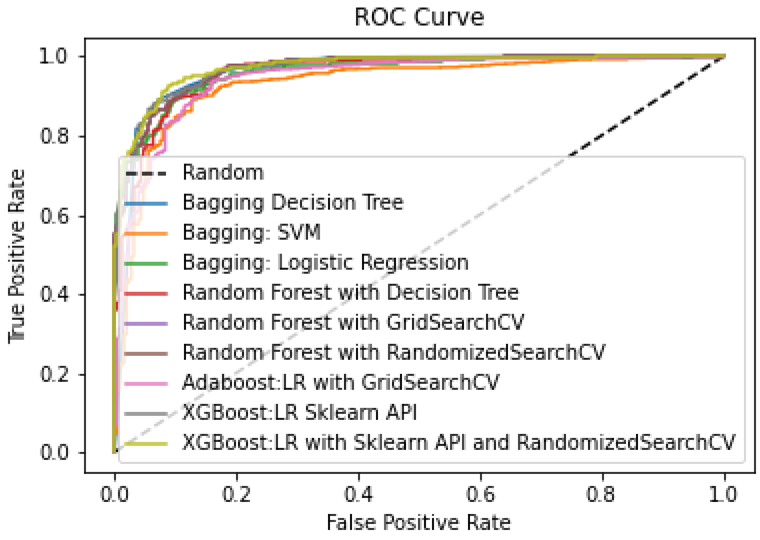


Figure 14: **ROC Curves for Numerical-Based Ensemble Classifiers**

6. Discussion

Figure 3 shows a comparison of the test and training accuracies for the Numerical-Based classifiers. Figure 4 shows a comparison for the weighted average precision and Group 0 precision. Figure 5 shows a comparison for the Group 0 recall and weighted average recall. Figure 6 shows a comparison for the cross-validation accuracy, weighted average F1-Score, and specificity of Group 0. Of the general classifiers, Decision Tree had the best performance with the highest training and test accuracy values shown in Figure 3. This is also reflected in Figure 11 showing the ROC curves for the general classifiers. As we see in Figures 3-6, Naive Bayes gave the worst performance results for the numerical features. This is likely due to the fact that Naive Bayes assumes features are independent from one another.

When comparing the performances for Decision Tree, SVM, and Logistic Regression when the bagging ensemble method was used, it was found that bagging improved the accuracies for all three classifiers. The ROC plot in Figure 12 shows that Decision Tree outperformed SVM and Logistic Regression. The weighted average precision, weighted average recall, weighted average F1-score, and specificity for the class with abnormally high AAS values all increased. Similar trends were found for the ROC/AUC values and cross validation accuracies for Decision Tree and Logistic Regression, however, these values decreased when using SVM. This is likely due to …

When comparing the general Decision Tree model to the Random Forest models, it was found that all metrics except the specificity of Class 0 increased. Tuning the hyperparameters of the Random Forest models increased the training accuracy when randomized search was used as well as the ROC/AUC and cross validation accuracy. No changes were found for the rest of the metrics.

When comparing the general Decision Tree model to the model using the Adaboost method, there was no change in the training accuracy, but an increase in the values of all other metrics except the specificity which decreased.

Comparing the Bagging and Adaboost ensemble methods, showed slightly higher training accuracy for the Adaboost method, but slightly higher test accuracy, weighted average precision, weighted average recall, weighted average F1-score, ROC/AUC, and cross validation accuracy values.

Figure 13 shows the ROC plots for the ensemble methods. It was found that the XGBoost using the Scikit-learn API after using randomized search had the highest ROC/AUC value.

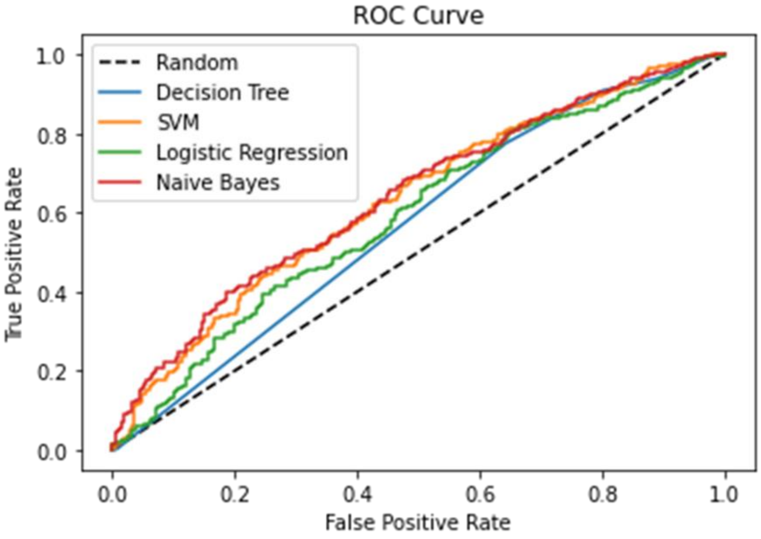


Figure 15: **ROC Curves for Text-Based General Classifiers**

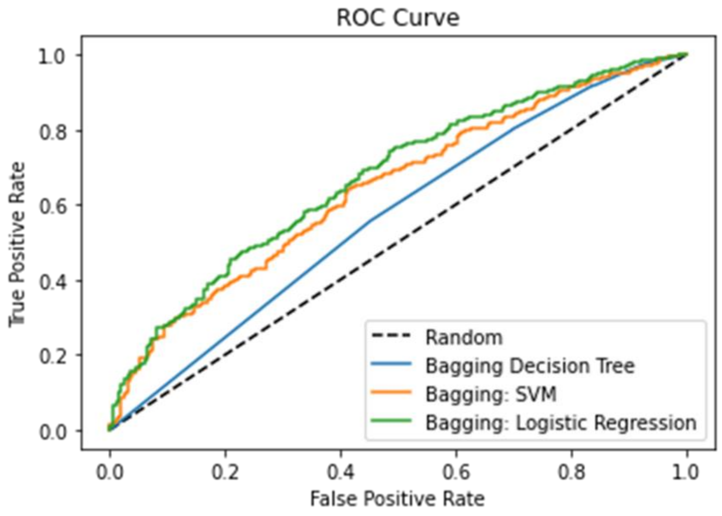


Figure 16: **ROC Curves for Text-Based Bagging Classifiers**

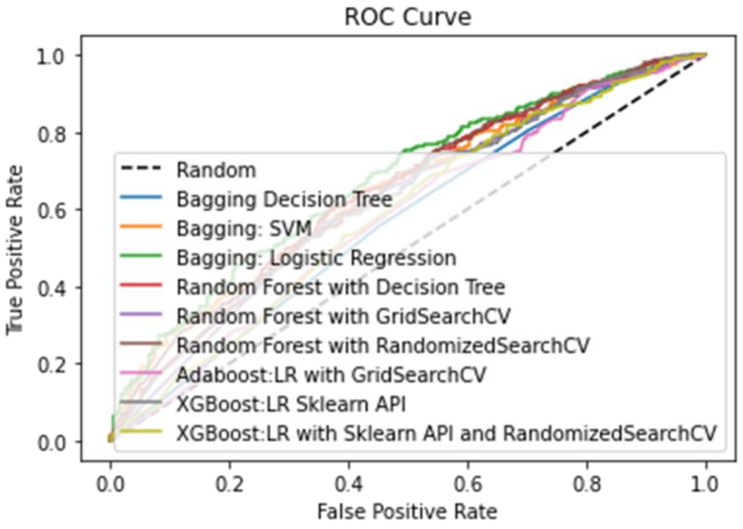


Figure 17: **ROC Curves for Text-Based Ensemble Classifiers**

One of the goals of this work was to investigate how the classifiers perform on the text features for the clinical trials classification. As the results suggest, the classifiers performed better when the numerical features were used. Titles of the clinical trials were used as the text features. Though the classifiers applied on the text-based features could not outperform their performances in the numerical features, it was hoped that at least they would have been better than what was found. One of the possible reasons behind that can be the weak correlation between a title and the Altmetric score of a clinical trial which might have made some of the classifiers perform so poorly.

Data imbalance can be another issue that was present in the original dataset. Besides, having more samples of the text data could result in better performance. Some classifiers showed 0 recall for one class even after tuning the hyperparameters and trying multiple approaches. Recall for some classifiers were also found poor in some earlier works on the same dataset. One assumption is that the titles of the clinical trials had a smaller number of overlapping i.e. common words among them which ultimately contributed negatively while applying text-based approach. Though some classifiers showed reasonable performance on the text data, it could do better if more training data could be used. As it seems from the experimental results is more datasets on this area are required for training better models for automatic categorization of clinical trials.

7. Conclusion and Future Work

For the numerical-based models, it was found that Decision Tree showed the best performance of general classifiers. It was also found that the use of ensemble models increased test accuracy by about 2%. The Random Forest models were shown to increase precision and recall values when compared to the general Decision Tree classifier. The bagging models with Decision Trees were found to give the largest specificity for the class with high AAS values. Finally, the XGBClassifier tuned with randomized search gave the largest ROC/AUC values overall.

For the text-based models, Support Vector Machine using the linear kernel showed the best performance among the general classifiers. It was found that the use of ensemble models increased the test accuracy by about 1%. The performance of the untuned Random Forest classifier was similar with both the GridSearchCV and RandomizedSearchCV tuning methods. The Logistic Regression model was found to give the largest specificity for the class with high AAS values. Finally, the overall performance of XGBoost was the best among the Ensemble methods for when using text features.

There are several possible directions that can be done to expand this work. For example, topic clustering of the titles and abstracts could be performed. The text features of the data could be used to make classification models to predict the known trial categories described above. The original datasets could also be expanded with newer trials in hopes of improving the accuracy of the models discussed. Finally, more complex text extract methods could be used in order to limit the number of features used.

References

[1] Wikipedia contributors, "Altmetric," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Altmetric&oldid=1011169199 (accessed December 10, 2021).

[2] Covid-19 Clinical Trials Website

https://covid19.nih.gov/treatments-and-vaccines/clinical-trials

[3] Iruku, Praveena, Martin Goros, Jonathan Gelfond, Jenny Chang, Susan Padalecki, Ruben Mesa, and Virginia G. Kaklamani. "Developing a model to predict accrual to cancer clinical trials: data from an NCI designated cancer center." Contemporary clinical trials communications 15 (2019): 100421.

[4] Lara Jr, Primo N., Debora A. Paterniti, Christine Chiechi, Corinne Turrell, Claudia Morain, Nora Horan, Lisa Montell et al. "Evaluation of factors affecting awareness of and willingness to participate in cancer clinical trials." Journal of Clinical Oncology 23, no. 36 (2005): 9282-9289.

[5] Umutyan, Ari, Christine Chiechi, Laurel A. Beckett, Debora A. Paterniti, Corinne Turrell, David R. Gandara, Sharon W. Davis, Ted Wun, Moon S. Chen Jr, and Primo N. Lara Jr. "Overcoming barriers to cancer clinical trial accrual: impact of a mass media campaign." Cancer: Interdisciplinary International Journal of the American Cancer Society 112, no. 1 (2008): 212-219.

[6] Thiele, Christian, Gerrit Hirschfeld, and Ruth von Brachel. "Clinical trial registries as Scientometric data: A novel solution for linking and deduplicating clinical trials from multiple registries." Scientometrics (2021): 1-18.

[7] Thelwall, Mike, and Kayvan Kousha. "Are citations from clinical trials evidence of higher impact research? An analysis of ClinicalTrials. gov." Scientometrics 109, no. 2 (2016): 1341-1351.

[8] Finch, Tom, Nina O'Hanlon, and Steve P. Dudley. "Tweeting birds: online mentions predict future citations in ornithology." Royal Society Open Science 4, no. 11 (2017): 171371.

[9] Wang, Justin, Naif M. Alotaibi, George M. Ibrahim, Abhaya V. Kulkarni, and Andres M. Lozano. "The spectrum of altmetrics in neurosurgery: the top 100 “trending” articles in neurosurgical journals." World neurosurgery 103 (2017): 883-895.

[10] Muñoz-Velandia, Oscar Mauricio, Daniel Gerardo Fernández-Ávila, Daniela Patino-Hernandez, and Ana María Gómez. "Metrics of activity in social networks are correlated with traditional metrics of scientific impact in endocrinology journals." Diabetes & Metabolic Syndrome: Clinical Research & Reviews 13, no. 4 (2019): 2437-2440.

[11] Suzan, Veysel, and Damla Unal. "Comparison of attention for malnutrition research on social media versus academia: Altmetric score analysis." Nutrition 82 (2021): 111060.

[12] Hayon, Solomon, Hemantkumar Tripathi, Ian M. Stormont, Meagan M. Dunne, Michael J. Naslund, and Mohummad M. Siddiqui. "Twitter mentions and academic citations in the urologic literature." Urology 123 (2019): 28-33.

[13] Lehane, Daniel J., and Colin S. Black. "Can altmetrics predict future citation counts in critical care medicine publications?." Journal of the Intensive Care Society 22, no. 1 (2021): 60-66.

[14] Anderson, P. Sage, Aubrey R. Odom, Hunter M. Gray, Jordan B. Jones, William F. Christensen, Todd Hollingshead, Joseph G. Hadfield et al. "A case study exploring associations between popular media attention of scientific research and scientific citations." PloS One 15, no. 7 (2020): e0234912.

[15] Erskine, Natalie, and Sharief Hendricks. "The Use of Twitter by Medical Journals: Systematic Review of the Literature." Journal of medical Internet research 23, no. 7 (2021): e26378.

[16] 2015. Altmetric Data is Now Available for ClinicalTrials.gov Study Records. Altmetric. Retrieved December 9, 2019 from https://www.altmetric.com/blog/altmetric-data-is-now-available-for-clinicaltrials-gov-study-records/

[17] Why did we build Dimensions | Dimensions. Dimensions. Retrieved December 9, 2019 from https://www.dimensions.ai/why-dimensions/

[18] Zheng, Chunmei, Guomei He, and Zuojie Peng. "A Study of Web Information Extraction Technology Based on Beautiful Soup." *J. Comput.* 10, no. 6 (2015): 381-387.

[19] Loper, Edward, and Steven Bird. "Nltk: The natural language toolkit." *arXiv preprint cs/0205028* (2002).

[20] Wikipedia contributors, "Stop word," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Stop_word&oldid=1057431778 (accessed December 11, 2021).

[21] Bafna, Prafulla, Dhanya Pramod, and Anagha Vaidya. "Document clustering: TF-IDF approach." In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 61-66. IEEE, 2016.

[22] Beel, Joeran, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. "Research paper recommender system evaluation: a quantitative literature survey." In Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, pp. 15-22. 2013.

[23] González, Sergio, Salvador García, Javier Del Ser, Lior Rokach, and Francisco Herrera. "A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities." *Information Fusion* 64 (2020): 205-237.

[24] Brownlee, Jason. "XGBoost with Python." *Machine Learning Mastery* (2019).

[25] Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." *Journal of machine learning research* 13, no. 2 (2012).