IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

COMPARE THE PERFORMANCE OF BORDA RANK AGGREGATION AND OTHER FEATURE SELECTION METHODS FOR TEXT SENTIMENT CLASSIFICATION

[1]Biraj Lahkar, [2]Dr. Jaibir Singh

[1]Research Scholar, [2]Associate Professor

[1]Department of Computer Science

[1]OPJS University, Churu, Rajasthan, India

Abstract: In data mining, the filtering feature selection approach is a well-known feature selection strategy. When the dimension of features is big, the filtering approach outperforms other feature selection methods in many circumstances. Because there are so many filtering methods described in prior work, how to choose an acceptable filtering method for a particular text data collection has become a "selection problem." Because determining the appropriate filtering method is frequently intractable in practise, this research proposes a different approach. We present a framework for feature selection that combines the results of several filtering approaches. In fact, rank aggregation, or the process of creating a superior rank list from many rank lists, is a hot issue being researched across many fields. In this research, we describe FR-Borda, a new feature selection approach based on the presented framework and Borda rank aggregation techniques. The novel feature selection approaches obtain better or equivalent results to conventional filtering methods in an empirical evaluation using tweeter text data, demonstrating the usefulness of our system.

***Index Terms* - Data Mining, Feature Selection, Filtering Approaches, Rank Aggregation.**

1. INTRODUCTION

Sentiment analysis, often known as opinion mining, is the process of collecting subjective information from source materials, such as views, feelings, and attitudes about a certain entity. It's an interdisciplinary study field that integrates natural language processing, text mining, and computational linguistics tools and methodologies [1]. Opinions play a significant role in the decision-making process as both factors and influences. In numerous domains, such as management sciences, political science, economics, and other social sciences disciplines, determining people's perceptions about a certain event may be incredibly essential [2]. The Web provides a large and ever-expanding source of information for obtaining opinions/sentiments about a specific topic, product, event, or person [3].

The process of sentiment analysis may be modelled as a classification issue. Sentiment analysis may be done at various levels of granularity. Sentiment analysis may be separated into three tiers based on the levels: document-level, sentence-level, and aspect-level sentiment analysis [4]. Document-level sentiment classification attempts to determine an entire document's overall sentiment orientation, such as a review text, assuming that each document contains information about a single entity. Sentiment analysis at the sentence level seeks to distinguish between subjective and objective sentences. The sentiment orientation of subjective statements is also discovered in sentence-level sentiment analysis. The classification of review papers at the document or sentence level of granularity does not convey all of the opinions about distinct aspects of a given item. As a result, aspect-level sentiment analysis is concerned with categorising attitudes based on specific features/aspects of things [3, 4].

Machine learning-based methods and lexicon-based methods are the two types of methodologies utilised in sentiment categorization. Decision tree classifiers, rule-based classifiers, probabilistic classifiers, and Naïve Bayes classifiers, as well as linear classifiers like Support Vector Machines and neural networks, have all been widely used in text and web mining applications [5]. Text characteristics are identified and chosen in order to process the data set of text documents. Text-based approaches for describing the data set in sentiment classification include term presence and term frequency, part of speech, opinion words, phrases, and negations [5]. The machine learning algorithm is then trained on a labelled data set, and a supervised classification model is built. Lexicon-based sentiment analysis approaches, on the other hand, use a sentiment lexicon (a collection of sentiment words) to establish an entity's sentiment orientation [3, 4].

Filtering, wrapper, embedding, and hybrid approaches are the four types of feature selection methods currently presented in the literature [1, 2, 4, 5, 6, 7]. When we pick the best filtering method for a certain feature selection job, one problem known as "Selection Difficulties" appears. To the best of our knowledge, this problem remains theoretically unsolved. Many academics prefer to test current filtering algorithms empirically before establishing heuristic recommendations. Because of the complexities of data attributes, counsel that works for one data set may not work for another. Many other communities, in fact, are dealing with a similar situation [8]. Building a robust model with good classification performance necessitates the selection of an acceptable representation and the deployment of an efficient feature selection strategy. The enormous dimensionality and irrelevancy of text data make text mining a challenging area [5]. As previously stated, sentiment analysis is a subfield of text mining, and the challenges faced in sentiment analysis data sets are similar to those found in text mining data sets. As a result, feature selection for sentiment analysis becomes an essential research topic. A fundamental difficulty in data mining is feature selection, which involves selecting a subset of the available characteristics to represent the samples. It is commonly believed that a well-chosen feature subset may successfully avoid both the curse of dimensionality and overfitting. Moreover, feature selection can help to minimise computational complexity [8]. This work provides a rank aggregation feature selection strategy for categorising text sentiment data that takes these concerns into consideration. Filter-based feature selection approaches, such as Chi Square (CHI), Gain Ratio (GR), Correlation Based Feature Selection (CFS), Information Gain (IG), OneR, ReliefF, Symetric Uncertainty (SU) have been effectively used because to their simplicity and relatively good performance [3, 4, 5]. The different base method ranking lists are then combined into a single ranking list using Borda Rank Aggregation Algorithms [3]. Rank aggregation, the process of creating a superior rank list from many rank lists, has been studied in a variety of fields. A feature fusion (selection) framework is created by combining filtering methods with rank aggregation approaches. Based on the framework, one novel feature selection approache FR-Borda is provided. The outcomes of the experiments suggest that the fusion framework may identify resilient and useful features [8].

The remainder of this work is divided into the following sections: Section 2 summarises related work on sentiment analysis and feature selection based on rank aggregation. The methodology are presented in Section 3. The experimental setup described in Section 4. The Experimental Results and Discussion is presented in Section 5. The conclusion is presented in Section 6.

2. RELATED WORK:

The available literature on feature selection methods in machine learning-based sentiment/text classification and rank aggregation-based feature selection approaches is briefly reviewed in this section.

Many machine learning and data mining issues have used feature selection. The goal of feature selection is to choose a collection of features that minimises a classifier's prediction error. It has been discovered during the last several years that employing and mixing multiple learning models on the same task might yield superior outcomes. Numerous diverse techniques for feature selection, including feature creation, feature ranking, multivariate feature selection, effective search techniques, and methods for feature validity evaluation, have been compiled in earlier studies, such as those by Guyon and Elisseeff [10] or Hall and Holmes [9, 11].

On a Chinese sentiment corpus, Tan and Zhang [6] tested the effectiveness of four feature selection methods (document frequency, chi-square statistics, mutual information, and information gain) on five learning algorithms (centroid classifier, K-nearest neighbour algorithm, winnow classifier, Nave Bayes classifier, and Support Vector Machines). Chen et al. [7] provided two feature selection metrics for multi-class text classification using the Naïve Bayes classifier (multi-class odds ratio and class discriminating measure). For text categorization, Javed et al. [12] proposed a two-stage feature selection technique. A feature ranking strategy, such as information gain or bi-normal separation measure, is used initially in this scheme. After that, a feature subset selection approach is used, such as the Markov blanket filter. Rank aggregation may be used to integrate several feature ranks acquired using different feature selection methods to create a more robust feature set. Jong et al. [13] proposed an ensemble feature ranking technique that incorporates feature rankings from several runs of the ROGER evolutionary feature selection strategy (ROC-based genetic learner). Based on the area under the ROC curve, the ROGER method generates a linear combination of characteristics. Prati [14] looked analysed the results of four rank aggregation methods: Borda, Condorcet, Schulze, and Markov Chain. Information gain, gain ratio, symmetric uncertainty, chi-square, OneR, and ReliefF feature selection techniques were used to produce feature rankings. According to the findings of the experiments, the Schulze rank aggregation approach performs better. The performance of rank aggregation approaches for ensemble gene selection was investigated by Dittman et al. [15]. Twenty-five feature ranking techniques were used, including the area under the ROC curve, deviation, F-measure, geometric mean, and Gini index. Nine rank aggregation approaches were used to investigate the effects of different rank aggregation methods. Enhanced Borda, exponential weighting, highest rank, lowest rank, mean, median, robust rank, Round Robin, and stability selection rank aggregation are some of the approaches used. According to the results of the experiments, rank aggregation approaches produce the best results, while the differences between the various rank aggregation methods are not statistically significant. Bouaguel et al. [16] proposed an ensemble feature selection approach combining the Relief algorithm, correlation-based feature selection, and the information gain measure. A genetic algorithm was used to create a consensus ranked list from the ranked lists of separate feature selection methods. Wald et al. [17] investigated whether ensemble approaches may help individual feature selection methods perform better in gene selection. Area under the ROC curve, probability ratio, fold change ratio, signal-to-noise ratio, and information gain were chosen as individual feature selection approaches. A mean rank aggregation approach has been used to integrate these different methodologies. The incorporation of mean rank aggregation improved the performance of information gain and fold change ratio approaches, according to the experimental data. Bouaguel et al. [18] proposed a credit score feature selection technique based on rank aggregation. Individual feature selection approaches included relief, Pearson correlation coefficient, and mutual information procedures. Two separate aggregation approaches are used to combine these strategies (majority voting and mean aggregation). The findings of the experiments showed that aggregation strategies work better in the credit scoring sector. Sarkar et al. [19] proposed a feature selection technique that combines Borda rank aggregation with information gain, chi-square, and symmetrical uncertainty-based feature selection methods. The method for integrating many individual feature selection approaches is a key in ensemble feature selection, but determining which efficient components to include in the feature selection model is also crucial.

Olsson and Oard [20] utilised three frequently used filter-based feature ranking strategies for text classification issues, with the combining methods being lowest, highest, and average rank. Wang et al. provide two intriguing research in this domain, and they do a few exceptional efforts in this area. The first looks at the ensembles of six regularly used filter-based rankers [21], while the second looks at 17 various feature ranking approaches ensembles [22], including six commonly used rankers, the signal-to-noise filter technique (S2N) [23], and 11 threshold-based rankers. The ensembles in this second study are made up of various numbers of rankers, ranging from two to eighteen single feature selection techniques [24].

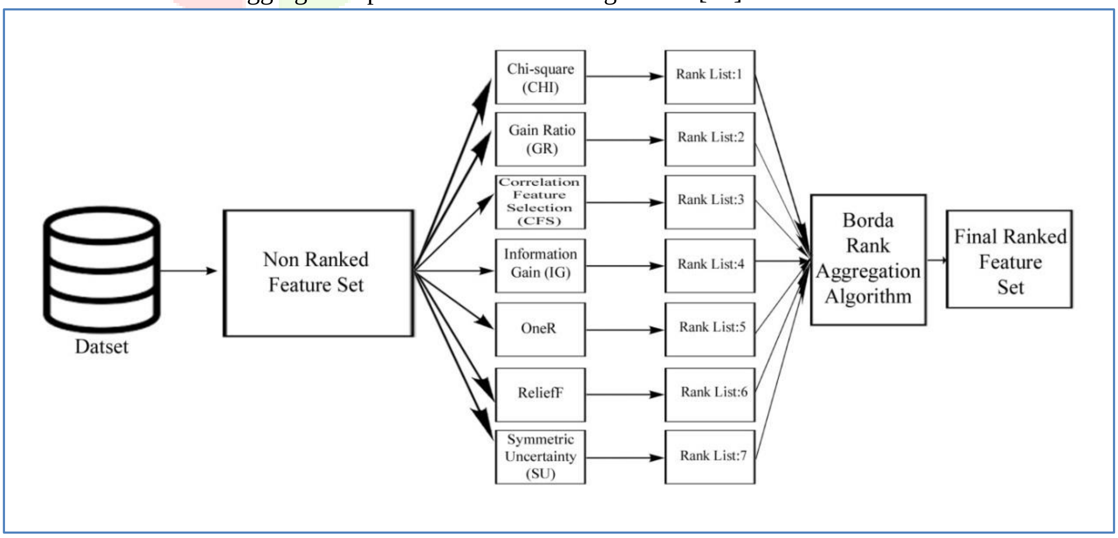
Yongjun Piao et al. [4] introduced an ensemble technique for high-dimensional data classification, in which each classifier is built from a separate set of features selected by partitioning redundant features. Their approach to feature redundancy was to partition the original feature space, then train each produced feature subset using a support vector machine, and then integrate the outputs of each classifier using the majority voting technique.

Alexandros Karatzoglou et al. [8], tutorial focuses on cutting-edge algorithmic developed in the area of recommender systems that provide a detailed picture of the progress of ranking models in the field, summarised the strengths and weaknesses of existing methods, and discussed open issues that could be promising for future community research. Krisztian Balog et al. [25] used supervised learning with a large feature set to compare these two methodologies in an experiment. The major conclusion is that ranking beats classification across all assessment settings and indicators, and research shows that a ranking-based method has more room for development in the future [26]. Jia [27] developed a hybrid FS approach for SDP that combines the strengths of three different filter methods (chi-squared, information gain, and correlation filter). The TopK features were chosen based on the average rating of each feature in the corresponding rank list. Models based on the hybrid FS approach outperformed models based on individual filter methods in their experiments. Nonetheless, the skewed rankings of each feature can impair the efficacy of averaging rank lists of features [28]. Furthermore, randomly picking TopK traits may not be the ideal strategy since valuable features may be overlooked during the selection process. To overcome the filter selection problem, Wang et al. [22] developed an ensemble of FS algorithms in SDP. Using 18 distinct filter FS approaches, 17 ensemble methods were implemented. The ensemble approach used averaging of feature rankings from individual rank lists. They concluded that ensemble techniques were preferable based on their findings. Averaging rank lists of features, like Jia [27], might be influenced by the skewed ranks of each characteristic. For feature selection in metric-based SDP, Xia et al. [29] combined ReliefF and correlation analysis. Furthermore, Malik et al. [30] an empirical comparison research on the application of an attribute rank technique was undertaken by. The use of principle component analysis (PCA) in conjunction with the ranker search method as a filter FS method was examined in detail. They came to the conclusion that using PCA in conjunction with the ranker search approach in the SDP process can increase the performance of classifiers. Despite the fact that their findings cannot be generalised owing to the narrow scope of their study, they are consistent with current SDP studies on the application of FS approaches in SDP. Regardless, choosing a good filter-based [31].

In this context, we proposed a feature selection technique FR-Borda that combines Borda rank aggregation with Chi-square (CHI), Gain Ratio (GR), Correlation Feature Selection (CFS), Information Gain (IG), One R, Relief F and Symmetric Uncertainty (SU) feature selection methods that maximises the Classification Accuracy and F-Measure. The method for integrating many individual feature selection approaches is a key in ensemble feature selection, but determining which efficient components to include in the feature selection model is also crucial [35]. The goal of our study is to construct a more robust ranking in order to reduce bias caused by sample variances, which is a big problem in twitter text sentiment analysis because there are so many variables [32]. In our study we rearranging the features in a paired analysis in accordance with how frequently one characteristic appears higher rated than another [33, 34, 14]. We are using four classifiers which are Naïve Bays Multinomial, LibLinear, 5-NN and Random Forest. This study makes a ten-fold contribution. This work provides a thorough empirical examination of individual feature selection strategies and our suggested feature selection method (FR-Borda) [3].

3. METHODOLOGY

The following are the phases in the rank aggregation based feature selection technique: 7 feature selection/evaluation strategies are used to examine a non-ranked feature collection. As a result, there are 7 sets of ranked feature sets with different rank orderings. The next step is to use the Borda Algorithm to perform rank aggregation on the feature sets, resulting in a final ranked feature set. The entire rank aggregation process is shown in Figure 2.1 [35].



Figur-2.1: Flow diagram of the rank aggregation based feature selection Algorithm.

3.1 Borda Rank Aggregation Method

We employ rank aggregation based on Borda ranking in this work. The score of a feature is calculated using a position-based scoring algorithm. Each place in a list created by each feature selection approach is assigned a predetermined score (this score is same for all the lists). The ultimate score for a separate feature is the total of all positional ratings from all lists, as shown in equation (1). The final score is used to calculate the final rank of a feature [35].

$$\boxed{score_{final} = \sum_{i=1}^{7} score_{pos(i,j)}} \quad (1)$$

The total number of feature selection strategies (or rankers) utilised is *7*. The *jth* position of a feature ranked by the ranker *i* is *pos(i, j)*. The score of a feature in list *i* generated by ranker *i* at the *jth* position is $score_{p(i,j)}$. The total of all the positional scores from all the lists is the $score_{final}$. We use candidates as features in this study. Algorithm 2.1 describes the rank aggregation phase [35].

Algorithm 2.1: Borda Count Algorithm

Obtain a *L* ranking list from a *L* sequence alignment; each list has *N* identities ordered in descending order, with the greater the order, the higher the rank.

for all ranking lists (*L*) **do**

  for all identities (*N*) **do**

    Calculate each identity's total Borda score as $B_c = \sum_{i=1}^{L} B_i$,where B_i is the score or rank of the i^{th} ranking list.

end for

end for

Sort B_c in ascending order and replace with the identity that corresponds [35].

3.2 Feature selection methods

The process of extracting an appropriate feature subset from a data collection so that classification algorithms may cope efficiently with high-dimensional feature spaces is known as feature selection. Feature selection strategies try to shorten the training time necessary to develop a classification model by eliminating irrelevant or duplicated features [36]. Wrapper-based approaches pick features based on the performance of a machine learning algorithm to enhance the prediction performance, whereas filter-based methods evaluate the merit/usefulness of features based on heuristics/evaluation metrics. Individual feature measures and group feature measures are the two types of filter-based feature selection techniques [38]. Individual feature measurements assess the value of characteristics using a specific assessment metric. A ranking of the characteristics is determined based on the value of this statistic. The value of feature subsets is assessed using group feature measurements. Individual feature measurements are more efficient in terms of running time than group-based measures. The individual filter-based measurements used in the framework are briefly described in this section [3].

3.2.1 Chi -square (CHI)

Chi-square is a popular class-sensitive feature selection approach that ranks features based on their Chi-square statistics alone, without taking into account feature interactions. This approach was originally introduced for categorical data alone, but it was later expanded to the continuous case [39]. The range of the numerical feature should be discretized into intervals before computing the Chi-square statistics of each feature [40]. The formula for Chi-square Test is:

$$X_c^2 = \frac{\sum (O_i - E_i)^2}{E_i}$$

Where, ***c*** = Degrees of freedom, *O* = Observed Value, *E* = Expected Value

3.2.2 Information gain (IG)

Information Gain (IG) [6] is a common metric for determining how effectively a word may be utilised for categorization based on the information it can supply to distinguish across classes. It's a metric indicating how much information a phrase contains [41]. It's an entropy-based approach for calculating impurity for feature values, and the formula is provided below [42].

$I(Y; X) = H(X) + H(Y) – H(X, Y)$

X and *Y's* combined entropy is *H(X, Y)*, where,

$$H(X,Y) = -\sum_{i=1}^{k}\sum_{j=1}^{l} P(X = x_j, Y = y_i)\log_2 P(X = x_j, Y = y_i)$$

When the predictive variable *X* is continuous rather than discrete, the information gain of the corresponding class attribute *Y* is calculated by taking into account all potential binary characteristics, $X\Theta$ that originate from *X* when a threshold Θ is set on *X* . Θ takes values from all of *X*'s values. The information gained is then simply: [43]

$I(Y; X) = \text{argmax}\ X\Theta\ I(Y; X\Theta)$

3.2.3 Gain Ratio (GR)

The Gain Ratio (GR) is a bias-reducing variation of the information gain. When picking an attribute, the gain ratio considers the number and size of branches. By taking into consideration the inherent information of a split, it corrects the information gain. The entropy of the distribution of instances into branches represents intrinsic information. As intrinsic information is greater, the value of an attribute falls [44, 45]:

$$\text{Gain (Attribute)} = \frac{\text{Gain ratio Attribute}}{\text{Intrinsic \_ info Attribute}}$$

3.2.4 Relief F

The original Relief method was developed by Kira and Rendell [46], and was influenced by instance-based learning [47]. The Relief algorithm [48] is a filter-based approach for ranking features that applies feature relevance criteria. The Relief method, unlike the statistical metrics used to rate the quality of qualities, takes contextual information into consideration. As a result, when there is a high reliance between the properties, it can manage them appropriately. The Relief algorithm, on the other hand, can only handle two-class issues. As a result, the ReliefF algorithm [49] was developed. It is a multi-class issue solution that extends the Relief technique to deal with incomplete and noisy data. By periodically sampling an instance, the ReliefF algorithm evaluates the worth of an attribute. It has the ability to work with both discrete and continuous data. The algorithm chooses an instance Ri at random. Then it looks for k of its closest neighbours in the same class (nearest hits Hj) and k of its closest neighbours in other classes (misses Mj (C)). The quality assessment for each characteristic is based on the closest hits and misses.

3.2.5 Symmetrical uncertainty coefficient

Witten and Frank, 2005 [51], established this as a standardised version of Mutual Information. The following is how it's defined:

$$U(A, B) = \frac{MI(A, B)}{H(A) + H(B)}$$

The Entropy of a set random variable X is H(X). This U has been used in several researches to recall pictures in medical image therapy [52]. Normalized Mutual Information is used in a variety of investigations [50, 53, 54, 55].

3.2.6 OneR

In rule-based algorithms, decision trees are used to generate categorization rules. The use of a decision tree has the drawback of being complex and difficult to comprehend. r = (p, q), where p is a precondition that performs a set of tests that may be approximated as true or false, and q is a class that is acceptable for the cases covered by rule r [56]. A broad rule of a rule-based algorithm aims to cover all instances that fit into a class q in a specified amount of time. OneR, which creates a one-level decision tree, is the simplest way for determining a categorization rule. OneR creates and tests rules on a single attribute at a time, branching for each value. The best classification for each branch is the one that appears most frequently in the training data [56, 57].

3.2.7 Correlation-based Feature Selecton (CFS)

CFS (Correlation-based Feature Selection) is a simple multivariate filter algorithm that uses a correlation-based heuristic evaluation function to rank feature subsets [58]. The evaluation function is biased in favour of subsets with features that are highly correlated with the class but uncorrelated with one another. Irrelevant features should be ignored because their correlation with the class will be low. Because they will be highly correlated with one or more of the remaining features, redundant features should be screened out. The extent to which a feature predicts classes in areas of the instance space not already predicted by other features will determine its acceptance [59].

3.3 N gram features

An N-gram is a group of words that repeatedly appear in a text. For building features for supervised machine learning models like decision trees and Naïve Bayes, N-gram is utilised. Before eliminating the stop words, n-gram tokenization is typically done. Go, R. Bhayani, and L. Huang (2009) [60] used three separate feature vectors—extracted unigrams, bigrams, and combinations of unigrams and bigrams—applied to several classifiers. Bigram is composed of two words, the first of which may be a stop word and which typically carries additional information. Bigram and unigram are strongly suggested for sentiment analysis of tweets in the majority of research studies. In our work we used unigrams feature extraction for sentiment analysis [61].

3.4 Classification Algorithms

To evaluate the effectiveness of FS techniques during the classification process, four widely used classification algorithms were examined. Unless otherwise noted Naïve Bayes Multinomial (NBM), LibLinear, 5-NN (k-nearest Neighbors classifier with k = 5; termed 5-NN in this study) and Random Forest. All of the classifiers were developed using the WEKA toolkit. A large-scale linear classification library that is open source is named Lib Linear. Both logistic regression and linear SVMs are supported by Lib Linear. Instance-based learning classifiers are KNNs. NBM for Bayes' theorem and REPTree for Tree-based methods. The classifier heterogeneity project's objective is to investigate how different FS methods and Borda Rank Aggregation Algorithm perform on different classifiers with variable feature ranking sizes.

4. EXPERIMENTAL SETUP

4.1 DATA SET USED:

One twitter dataset, tweets.csv, was collected for our tests from the publicly available site https://www.kaggle.com. The data aligns with an increase of COVID-19 cases in India during that time period. The tweets.csv dataset comprises tweets with the hashtag "covidindia" from the beginning of the epidemic until April 28, 2021. There are 9655 tweets, and the sentiment label has two possible values: negative and positive. Unigram (one word inside a tweet's text) were extracted as features. For our experiment, after data pre-processing and extraction our dataset was made up of 5153 instances and 1621 features or attributes.

4.2 WEKA WORKBENCH

To create and assess our experiments, we utilised Weka, a machine learning workbench. WEKA stands for "Waikato Environment for Knowledge Analysis" and is a free service provided by the University of Waikato in New Zealand. This workbench has a user-friendly interface and a number of capabilities for creating and analysing machine learning models [12]. These models may be used for a variety of purposes, including assessing essays automatically. An HP 15-r006TU laptop was used for all of the studies. The laptop has a 1.70 GHz Intel(R) Core(TM) i3 – 4010U CPU and 4 GB of RAM, but WEKA workbench is set to use just 1 GB. The laptop's operating system is Windows 7 64 bit [62].

4.3. EVALUATION MEASURE

We assess our algorithm using the following evaluation criteria:

4.3.1 CLASSIFICATION ACCURACY

The percentage of instances that a given classifier accurately classified is known as classification accuracy or it is defined as the number of correctly classified reviews to the total number of reviews. It is measured in percentage. For the purpose of assessment, the classification accuracy of this feature subset was examined and recorded using four distinct classifiers. Classification Accuracy defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

In where TP stands for "True Positive," TN for "True Negative," FP for "False Positive," and FN for "False Negative".

4.3.2 F-MEASURE

A harmonic mean of recall (R) and precision (P) is the definition of the F-measure (F). This F-measure represents the weighted average of the two quantities of precision and recall. F-measure has a maximum value of 1, and a minimum value of 0. Defining the F measure as:

$$F = \frac{2PR}{P + R}$$

The same four classifiers were used in our work to generate the F-measure using the feature subset [19].

4.3.3 K-FOLD CROSS VALIDATION

In this method, a data flow made up of n cases is first assembled; each having values for the covariates and response variables. Typically, the case file is then divided into k equal portions and randomly assigned. The first k segment, which consists of n/k cases, is removed from consideration. The remaining (n - n/k) cases are then used to parameterize a model, which is then tested against the first segment to determine the rate of classification error by comparing the model's results to each case's known outcomes (the values of the response variables). The model is then parameterized with the remaining examples, tested against the second segment, and so on for all k segments. Next, the second k segment from the complete case collection is set aside. We employed 10-Fold Cross Validation in our work [63].

5. EXPERIMENTAL RESULTS AND DISCUSSION

We evaluated the performance of our feature selection algorithm in terms of classification accuracy and F-measure by comparing with seven feature selection techniques namely including our suggested method FR-Borda and WFS (without feature selection). We refer our rank aggregation based feature selection algorithm as FR-Borda in the figures-2.1 shown in this paper. The results of classification accuracy over Dataset1 are given in Table-5.1. Again the results of F-Measure over Dataset1 are given in Figure 5.2. We utilised seven feature subset sizes for all of the feature rankers: 20, 70, 120, 170, 220, 270 and 320. These measurements were chosen to reflect a variety of feature subset sizes. In our dataset, the Classification Accuracy and F-Measure with our suggested method FR-Borda is higher as compared to those with the seven FS methods and "WFS" which stands for "without feature selection". In Table-5.1 and Table-5.2 each column, the best model for each feature subset size is boldfaced. WFS is included as an extra ranker in each table's bottom row. This allows the effectiveness of feature selection to be compared to the effectiveness of not using it. WFS, on the other hand, does not use the feature subset sizes stated in the tables since it uses all 1621 features available in our dataset and repeats them for each subset size.

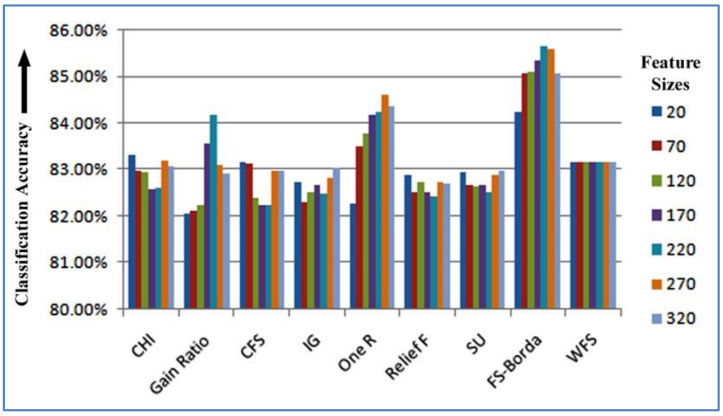
Table-5.1: Classification Accuracy Results of Dataset1

	Classifier	Classification Accuracy						
		20	70	120	170	220	270	320
NBM	CHI	83.33%	82.98%	82.96%	82.59%	82.61%	83.19%	83.08%
	Gain Ratio	82.05 %	82.11%	82.24%	83.58%	84.18%	83.12%	82.92%
	CFS	83.17%	83.14%	82.40%	82.26%	82.24%	83.00%	83.00%
	IG	82.73%	82.30%	82.53%	82.69%	82.50%	82.83%	83.06%
	OneR	82.28%	83.50%	83.78%	84.18%	84.24%	84.63%	84.36%
	ReliefF	82.90%	82.52%	82.73%	82.52%	82.44%	82.75%	82.71%
	SU	82.94%	82.69%	82.65%	82.67%	82.52%	82.88%	83.00%
	FR-Borda	**84.24%**	**85.08%**	**85.10%**	**85.35%**	**85.66%**	**85.60%**	**85.08%**
	WFS	83.16%	83.16%	83.16%	83.16%	83.16%	83.16%	83.16%
Lib Linear	CHI	83.39%	83.45%	83.80%	83.62%	83.74%	84.42%	84.79%
	Gain Ratio	82.92%	82.75%	84.30%	84.11%	84.40%	84.51%	84.84%
	CFS	83.31%	83.49%	83.35%	83.31%	83.76%	84.38%	84.55%
	IG	82.94%	83.27%	83.99%	83.85%	83.83%	84.30%	84.71%
	OneR	83.56%	83.49%	83.76%	84.38%	84.16%	84.18%	84.20%
	ReliefF	82.30%	82.75%	83.52%	83.64%	83.64%	83.76%	83.70%
	SU	83.43%	83.35%	83.82%	83.91%	83.62%	84.22%	84.71%
	FR-Borda	**84.20%**	**85.21%**	**85.79%**	**86.44%**	**86.86%**	**86.77%**	**86.94%**
	WFS	82.53%	82.53%	82.53%	82.53%	82.53%	82.53%	82.53%
5-NN	CHI	81.35%	82.03%	81.97%	81.93%	82.22%	81.84%	81.87%
	Gain Ratio	82.63%	82.67%	**83.25%**	**82.92%**	**82.94%**	81.93%	82.26%
	CFS	82.17%	82.07%	82.05%	82.15%	82.22%	82.42%	**82.79%**
	IG	80.92%	81.49%	81.91%	81.74%	81.54%	81.76%	81.86%
	OneR	83.49%	**83.17%**	82.83%	82.59%	82.69%	**83.00%**	82.67%
	ReliefF	81.20%	80.19%	80.57%	80.25%	80.30%	81.02%	81.14%
	SU	82.24%	81.99%	82.44%	81.68%	81.70%	81.89%	82.13%
	FR-Borda	**84.15%**	82.34%	82.81%	82.42%	82.59%	82.57%	82.53%
	WFS	82.15%	82.15%	82.15%	82.15%	82.15%	82.15%	82.15%
Random Forest	CHI	80.34%	82.03%	82.71%	82.77%	83.33%	83.41%	83.37%
	Gain Ratio	82.83%	82.71%	**83.87%**	**84.75%**	82.52%	83.12%	**83.80%**
	CFS	81.67%	81.68%	82.67%	82.86%	82.84%	83.21%	83.37%
	IG	79.68%	82.15%	82.81%	83.29%	83.02%	83.23%	83.54%
	OneR	83.27%	**82.83%**	82.96%	83.27%	83.14%	83.33%	83.68%
	ReliefF	78.94%	80.44%	81.27%	82.30%	82.81%	83.04%	83.19%
	SU	81.06%	81.56%	82.86%	83.10%	82.98%	83.37%	83.43%
	FR-Borda	**84.99%**	82.53%	83.17%	83.23%	**83.66%**	**83.80%**	83.78%
	WFS	77.62%	77.62%	77.62%	77.62%	77.62%	77.62%	77.62%

5.1 Classification Accuracy

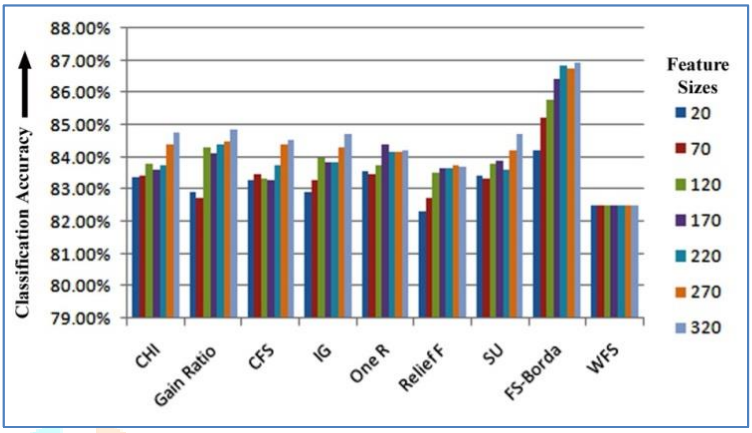
We'll start by examining the outcomes from the Naïve Bayes Multinomial. It can be demonstrated that CHI performs better than WFS when 20 and 220 features are used. But for other feature sizes, CHI never performs better than WFS. Gain Ration performs better than WFS for the top 170 and 220 features, but for other feature sizes, GR has never outperformed WFS. Only the top 20 features are where CFS beats WFS. In whatever size of features, IG has never beaten WFS. With the exception of the top 20 features, OneR consistently exceeds WFS in terms of feature size. ReliefF has never been surpassed WFS in any feature size. Additionally, in whatever size of top-ranked features, SU has never outperformed WFS. FR-Borda, the approach we recommend, consistently beats WFS in all sizes of top-ranked features. In terms of classification accuracy, FR-Borda ranks first in the top 20, 70, 120, 170, 220, 270, and 320 features. FR-Borda, which has an accuracy of 85.66 % with 220 of the top features, is the model that performs the best when trained with Naïve Bayes Multinomial. The Classification Accuracy results of seven Feature Ranking Methods, including FR-Borda and WFS based on Naïve Bayes Multinomial Classifier, is shown in Figure-5.1.

Figure-5.1: Classification Accuracy comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on Naïve Bayes Multinomial (NBM) Classifier



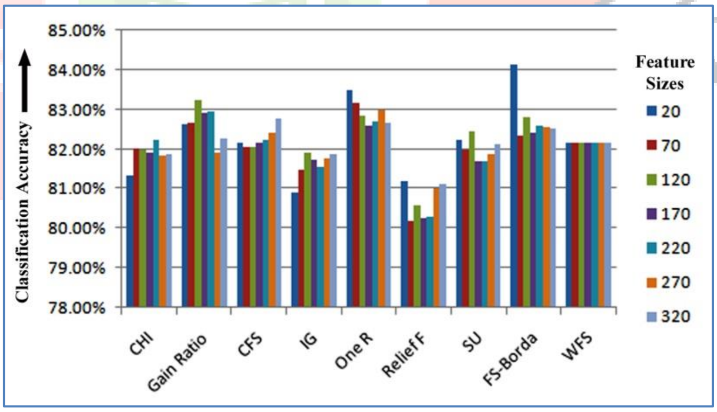
We can see from the LibLinear classifier's results that CHI, GR, CFS, IG, OneR, and SU beat WFS for all feature sizes. With the exception of the top 20 features, ReliefF consistently beats WFS in all other top-ranked feature sizes. FR-Borda, the approach we recommend, consistently beats WFS in all sizes of top-ranked features. In terms of classification accuracy, FR-Borda ranks first in the top 20, 70, 120, 170, 220, 270, and 320 features. The FR-Borda model, which has an efficiency of 86.86 % with 320 of the top features, is the best model trained with the LibLinear classifier. The Classification Accuracy results of seven Feature Ranking Methods, including FR-Borda and WFS based on LibLinear Classifier, is shown in Figure-5.2.

Figure-5.2: Classification Accuracy comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on LibLinear Classifier



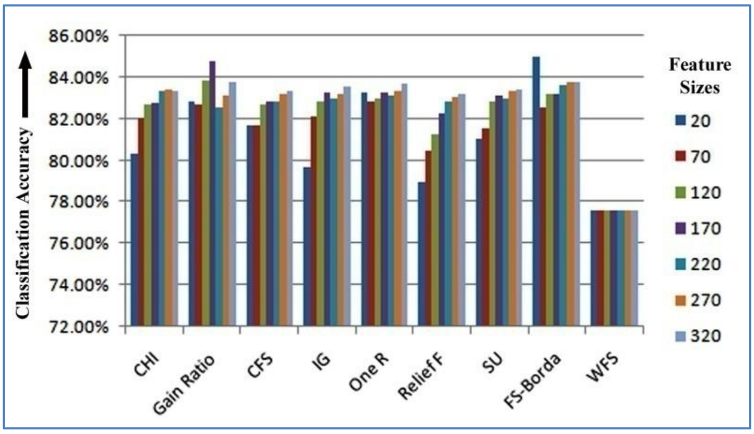
It is clear from the 5-NN results that CHI never outperforms WFS, with the exception of the top 220 features. Except for the top 270 features, GR consistently beats WFS. For top ranked features 120, 170, and 220, GR has the highest Classification Accuracy, at 83.25 %, 82.92 %, and 82.94 %, respectively. In the top 20, 170, 220, 270, and 320 features, CFS performs better than WFS. In the top 70 and 120 features, CFS has never beaten WFS. When we choose 320 features, CFS has the highest Classification accuracy rate at 82.79 %. In whatever size of top-ranked features, IG has never outperformed WFS. In whatever size of top-ranked features, OneR consistently beats WFS. OneR has the highest Classification Accuracy in the top 70 and 220 features, at 83.17% and 83.00%, respectively. Additionally, ReliefF has never outperformed top-ranked features trained with 5-NN in any size. SU beats WFS in the top 20 and 120 features, but in all other feature sizes, except 20 and 120, it never does so. FR-Borda, the approach we recommend, consistently beats WFS in all sizes of top-ranked features. FR-Borda has the highest Classification Accuracy in top 20 numbers of features. Classification Accuracy of 84.15% FR-Borda is the best performing model trained with 5-NN. The Classification Accuracy results of seven Feature Ranking Methods, including FR-Borda and WFS based on 5-NN Classifier, is shown in Figure-5.3.

Figure-5.3: Classification Accuracy comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on 5-NN Classifier



Finally, we demonstrate that all FS approaches beat WFS in any number of top-ranked feature sizes using the findings for the Random Forest classifier. When we chose the top 120, 170, and 320 ranked features, GR had the highest Classification Accuracy rate at 83.87%, 84.75%, and 83.80%, respectively. For the top 70 features, OneR has the highest Classification Accuracy, at 82.83%. FR-Borda, the approach we recommend, consistently beats WFS in all sizes of top-ranked features. With accuracy rates of 84.99%, 83.66%, and 83.80% for the top 20, 170, and 220 features, respectively, FR-Borda has the highest classification accuracy. When we choose the top 20 features, the model trained using Random Forest that performs the best is FR-Borda, with 84.99%. The Classification Accuracy results of seven Feature Ranking Methods, including FR-Borda and WFS based on Random Forest Classifier, is shown in Figure-5.4.

Figure-5.4: Classification Accuracy comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on Random Forest Classifier



5.2 F-MEASURE

We will first examine the Naïve Bayes Multinomial findings for F-Measure. It can be demonstrated that WFS consistently outperforms CHI, GR, CFS, IG, OneR, ReliefF, and SU for every top-ranked feature. When we selected 20 and 70 of the top-ranked features, the suggested approach FR-Borda never outperformed WFS. However, whether we selected 120, 170, 220, 270, and 320 features, FR-Borda consistently outperforms WFS. The highest F-Measure values for the 120, 170, 220, 270, and 320 features for the NBM classifier-trained FR-Borda are 0.841, 0.847, 0.852, 0.852, and 0.848, respectively. The best performing model trained with Naïve Bayes Multinomial is FR-Borda of 0.852 with 170 and 220 top ranked features. The F-Measure results of seven Feature Ranking Methods, including FR-Borda and WFS based on Naïve Bayes Multinomial Classifier, is shown in Figure-5.5.

Figure-5.5: F-Measure comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on Random Forest Classifier

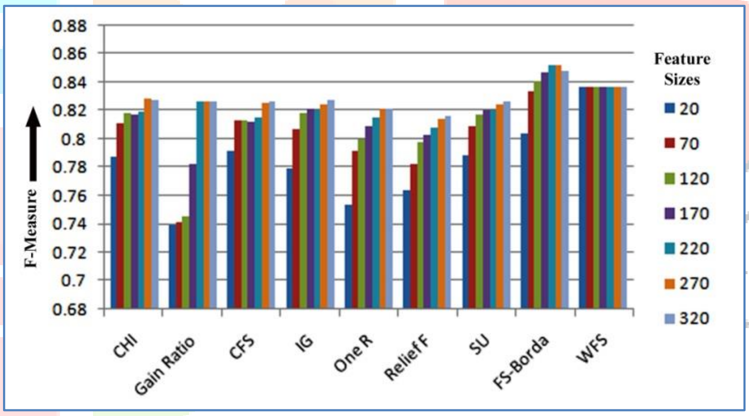
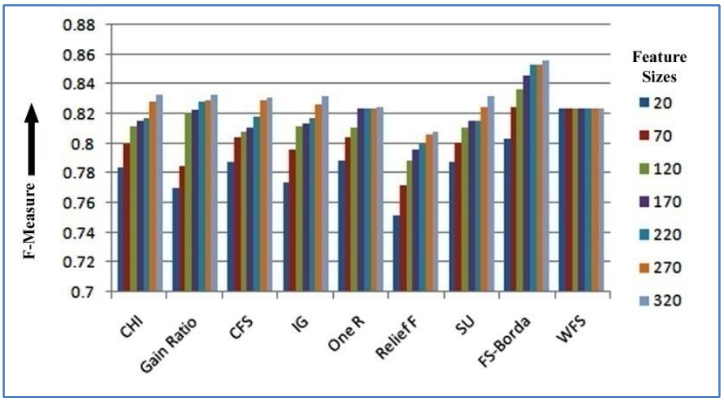


Table-5.2: F-Measure of Dataset1

	Classifier	F-Measure						
		20	70	120	170	220	270	320
NBM	CHI	0.788	0.811	0.818	0.817	0.819	0.829	0.828
	Gain Ratio	0.740	0.742	0.746	0.783	0.827	0.827	0.827
	CFS	0.792	0.813	0.813	0.812	0.815	0.826	0.827
	IG	0.779	0.807	0.818	0.822	0.820	0.825	0.828
	OneR	0.754	0.792	0.800	0.809	0.815	0.822	0.822
	ReliefF	0.764	0.783	0.798	0.803	0.808	0.814	0.816
	SU	0.789	0.809	0.817	0.821	0.820	0.825	0.827
	FR-Borda	0.804	0.834	**0.841**	**0.847**	**0.852**	**0.852**	**0.848**
	WFS	**0.837**	**0.837**	0.837	0.837	0.837	0.837	0.837
Lib Linear	CHI	0.784	0.800	0.812	0.815	0.817	0.828	0.833
	Gain Ratio	0.770	0.785	0.821	0.823	0.828	0.829	0.833
	CFS	0.788	0.804	0.808	0.811	0.818	0.829	0.831
	IG	0.774	0.796	0.812	0.814	0.817	0.826	0.832
	OneR	0.789	0.804	0.811	0.824	0.824	0.824	0.825
	ReliefF	0.752	0.772	0.789	0.796	0.800	0.806	0.808
	SU	0.788	0.801	0.811	0.815	0.815	0.825	0.832
	FR-Borda	0.803	**0.825**	**0.837**	**0.846**	**0.853**	**0.853**	**0.856**
	WFS	**0.824**	0.824	0.824	0.824	0.824	0.824	0.824
5-NN	CHI	0.769	0.773	0.764	0.761	0.763	0.762	0.763
	Gain Ratio	0.762	0.768	**0.788**	**0.791**	**0.790**	0.766	0.767
	CFS	0.780	0.774	0.768	0.762	0.760	0.761	0.762
	IG	0.764	0.770	0.773	0.765	0.762	0.761	0.761
	OneR	0.789	**0.793**	0.787	0.782	0.777	**0.775**	**0.770**
	ReliefF	0.757	0.755	0.766	0.763	0.763	0.769	0.767
	SU	0.782	0.777	0.777	0.764	0.761	0.762	0.765
	FR-Borda	**0.800**	0.780	0.777	0.767	0.769	0.767	0.763
	WFS	0.753	0.753	0.753	0.753	0.753	0.753	0.753
Random Forest	CHI	0.774	0.783	0.782	0.781	0.785	0.786	0.785
	Gain Ratio	0.767	0.784	**0.815**	**0.809**	**0.803**	0.791	0.790
	CFS	0.783	0.784	0.787	0.785	0.780	0.785	0.786
	IG	0.767	0.784	0.785	0.789	0.784	0.785	0.786
	OneR	0.786	**0.799**	0.804	0.808	0.805	**0.806**	**0.810**
	ReliefF	0.749	0.771	0.773	0.778	0.782	0.784	0.785
	SU	0.779	0.781	0.788	0.788	0.781	0.786	0.782
	FR-Borda	**0.822**	0.789	0.791	0.785	0.789	0.789	0.788
	WFS	0.769	0.769	0.769	0.769	0.769	0.769	0.769

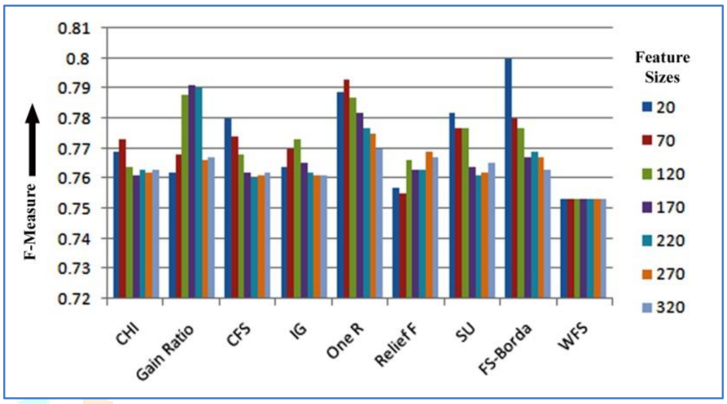
Next, we'll examine the LibLinear outcomes. It can be demonstrated that CHI never outperforms WFS in F-Measure for the top-ranked features of 20, 70, 120, 170, and 220. But WFS lost to CHI for the top-ranked features of 270 and 320. Additionally, whether we used 220, 270, and 320 features, GR outperforms WFS. When we selected 170, 220, 270, and 320 feature sizes, OneR performed admirably. OneR defeated WFS for 320 features. Only when we used the 270 and 320 highest ranked features, CFS, IG, and SU defeat WFS. In whatever size of features, ReliefF has never beaten WFS. With the exception of the top 20 features, our recommended approach, FR-Borda, consistently outperforms WFS and is superior in all top-ranked features. FR-Borda has the highest F-Measure values of 0.825, 0.837, 0.846, 0.853, 0.853 and 0.856 when we picked 70, 120, 170, 220, 270 and 320 top ranked features respectively. The best performing model trained with LibLinear is FR-Borda with 0.856 F-Measure value when we picked 320 top ranked features. The F-Measure results of seven Feature Ranking Methods, including FR-Borda and WFS based on Liblinear Classifier, is shown in Figure-5.6.

Figure-5.6: F-Measure comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on Random Forest Classifier



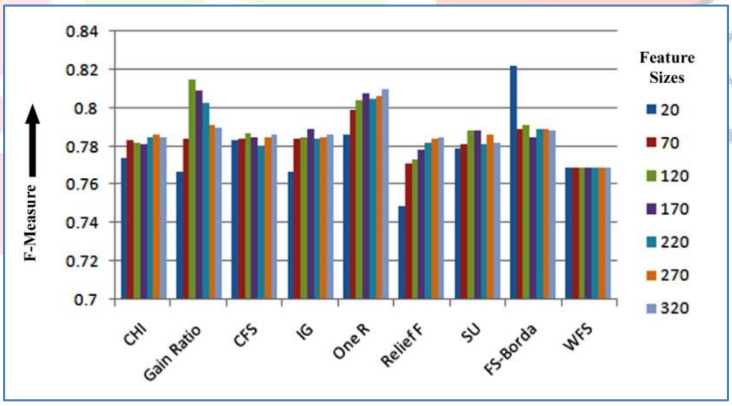
We'll examine the 5-NN findings next. It can be demonstrated that all FS approaches outperform WFS for all feature sizes. When we selected 120, 170, and 220 of the top-ranked features, GR had the highest F-Measure values of 0.788, 0.791, and 0.790, respectively. When we selected 70, 270, and 320 of the top-ranked features, OneR had the highest F-Measure values at 0.793, 0.775, and 0.770, respectively. Our recommended approach, FR-Borda, consistently outperforms WFS in any top-ranked feature. When we selected the top 20 features, FR-Borda had the highest F-Measure values of 0.800. The FR-Borda model, with a performance of 0.800 with 20 top-ranked features, is the best model trained with 5-NN. The F-Measure results of seven Feature Ranking Methods, including FR-Borda and WFS based on Liblinear Classifier, is shown in Figure-5.7.

Figure-5.7: F-Measure comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on Random Forest Classifier



We will now examine the outcomes of the Random Forest. It can be demonstrated that the CHI method outperforms the WFS for all feature sizes. When we examined the GR results, it was evident that GR performed better than WFS in all feature sizes except top 20 features. When we selected 120, 170, and 220 of the top-ranked features, GR had the highest F-Measure values at 0.815, 0.809, and 0.803, respectively. In all feature sizes, CFS and IG are superior to WFS. When we choose 70, 270, and 320 features for OneR, it performs well and also has the highest F-Measure values of 0.799, 0.806, and 0.810. Except for 20 features, ReliefF performs better than WFS. In any feature size, SU consistently outperforms WFS. Our recommended approach, FR-Borda, consistently outperforms WFS in any top-ranked feature sizes. FR-Borda has the highest F-Measure values of 0.822 when we picked 20 top ranked features. The best performing model trained with Random Forest is FR-Borda with 0.822 F-Measure value for 20 top ranked features. The F-Measure results of seven Feature Ranking Methods, including FR-Borda and WFS based on Liblinear Classifier, is shown in Figure-5.8.

Figure-5.8: F-Measure comparison of seven Feature Ranking Methods, including FR-Borda and WFS based on Random Forest Classifier



For all learners evaluated, our suggested strategy, FR-Borda, works exceptionally well and yields the best results. For 320 top-ranked features trained using the LibLinear classifier, the best Classification Accuracy was attained by FR-Borda, which was 86.94% accurate. Similar to this, for 320 top-ranked features learned with the LibLinear classifier, the best F-Measure value obtained by FR-Borda is 0.856. As a result our recommend feature selection method is FR-Borda with LibLinear classifier.

6. CONCLUSION

For tweet sentiment categorization, feature engineering approaches generally yield a huge number of features. When a large number of instances are combined, the resulting dataset might have a high dimensionality. Furthermore, it is computationally difficult to train classifiers on a huge dataset. Feature selection, which has gotten minimal attention in tweet sentiment classification research, picks an ideal collection of features, reducing the dataset's dimensionality, lowering computing costs, and maybe improving Classification Accuracy and F-Measure value. Utilizing four different learners, this study looked at seven filter-based feature selection strategies including our suggested method and compared them to using without feature selection. These strategies are used to pick seven distinct feature subsets from a dataset i.e. Dataset1, retrieved from https://www.kaggle.com.

Our work findings are encouraging, and future research should look at further feature selection approaches as well as the use of more than 320 features. This research should be broadened to include more datasets in order to see if the tendencies found in this study are also present in other datasets.

REFERENCES

[1] En.wikipedia.org. 2022. *Sentiment analysis - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Sentiment_analysis> [Accessed 26 June 2022].

[2] Liu, B., 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1), pp.1-167.

[3] Onan, A. and Korukoğlu, S., 2016. A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science*, 43(1), pp.25-38.

[4] Medhat, W., Hassan, A. and Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), pp.1093-1113.

[5] Aggarwal, C. and Zhai, C., 2012. A Survey of Text Classification Algorithms. *Mining Text Data*, pp.163-222.

[6] TAN, S. and ZHANG, J., 2008. An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications*, 34(4), pp.2622-2629.

[7] Chen, J., Huang, H., Tian, S. and Qu, Y., 2009. Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, 36(3), pp.5432-5435.

[8] Wu, O., Zuo, H., Zhu, M., Hu, W., Gao, J. and Wang, H., 2009. Rank Aggregation Based Text Feature Selection. *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*,.

[9] *Lecture Notes in Computer Science*, 2015. Advances in Computational Intelligence.

[10] Guyon, I. and Elisseeff, A., 2003. An Introduction of Variable and Feature Selection. *The Journal of Machine Learning Research*, 3, pp.1157–1182.

[11] Hall, M. and Holmes, G., 2003. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6), pp.1437-1447.

[12] Javed, K., Maruf, S. and Babri, H., 2015. A two-stage Markov blanket based feature selection algorithm for text classification. *Neurocomputing*, 157, pp.91-104.

[13] Jong, K., Mary, J., Cornuéjols, A., Marchiori, E. and Sebag, M., 2004. Ensemble Feature Ranking. *Lecture Notes in Computer Science*, pp.267-278.

[14] Prati, R., 2012. Combining feature ranking algorithms through rank aggregation. *The 2012 International Joint Conference on Neural Networks (IJCNN)*,.

[15] Dittman, D., Khoshgoftaar, T., Wald, R. and Napolitano, A., 2013. Classification Performance of Rank Aggregation Techniques for Ensemble Gene Selection. In: *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference*. pp.420-425.

[16] Bouaguel, W., Brahim, A. and Limam, M., 2013. FeatureSelectionbyRankAggregationandGeneticAlgorithms. *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing (KDIR-2013)*, pp.74-81.

[17] Wald, R., Khoshgoftaar, T. and Dittman, D., 2012. Ensemble Gene Selection Versus Single Gene Selection: Which Is Better?. Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, pp.350-355.

[18] Bouaguel, W., Bel Mufti, G. and Limam, M., 2013. Rank Aggregation for Filter Feature Selection in Credit Scoring. *Mining Intelligence and Knowledge Exploration*, pp.7-15.

[19] Sarkar, C., Cooley, S. and Srivastava, J., 2014. Robust Feature Selection Technique Using Rank Aggregation. *Applied Artificial Intelligence*, 28(3), pp.243-257.

[20] Olsson, J. and Oard, D., 2006. Combining feature selectors for text classification. *Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06*,.

[21] Wang, H., Khoshgoftaar, T. and Gao, K., 2010. Ensemble Feature Selection Technique for Software Quality Classification. SEKE 2010: Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering, pp.215-220.

[22] Wang, H., Khoshgoftaar, T. and Napolitano, A., 2010. A Comparative Study of Ensemble Feature Selection Techniques for Software Defect Prediction. *2010 Ninth International Conference on Machine Learning and Applications*,.

[23] Yang, C., Huang, C., Wu, K. and Chang, H., 2008. A Novel GA-Taguchi-Based Feature Selection Method. *Lecture Notes in Computer Science*, pp.112-119.

[24] Seijo-Pardo, B., Bolón-Canedo, V., Porto-Díaz, I. and Alonso-Betanzos, A., 2015. Ensemble Feature Selection for Rankings of Features. *Advances in Computational Intelligence*, pp.29-42.

[25] Balog, K. and Ramampiaro, H., 2013. Cumulative citation recommendation. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*,.

[26] Koturwar, P., Girase, S. and Mukhopadhyay, D., 2015. Performance analysis of classification and ranking techniques. *2015 Annual IEEE India Conference (INDICON)*,.

[27] Jia, L., 2018. A Hybrid Feature Selection Method for Software Defect Prediction. *IOP Conference Series: Materials Science and Engineering*, 394, p.032035.

[28] Jacquier, E., Kane, A. and Marcus, A., 2003. Geometric or Arithmetic Mean: A Reconsideration. *Financial Analysts Journal*, 59(6), pp.46-53.

[29] Xia, Y., Yan, G., Jiang, X. and Yang, Y., 2014. A new metrics selection method for software defect prediction. *2014 IEEE International Conference on Progress in Informatics and Computing*,.

[30] Malik, M., Yining, L. and Shaikh, S., 2020. The Role of Attribute Ranker using classification for Software Defect-Prone Data-sets Model: An Empirical Comparative Study. *2020 IEEE International Systems Conference (SysCon)*,.

[31] Balogun, A., Basri, S., Mahamad, S., Abdulkadir, S., Capretz, L., Imam, A., Almomani, M., Adeyemo, V. and Kumar, G., 2021. Empirical Analysis of Rank Aggregation-Based Multi-Filter Feature Selection Methods in Software Defect Prediction. *Electronics*, 10(2), p.179.

[32] Venkatesh, B. and Anuradha, J., 2019. A Review of Feature Selection and Its Methods. *Cybernetics and Information Technologies*, 19(1), pp.3-26.

[33] Sebag, M., Aze, J. and Lucas, N., n.d. Impact studies and sensitivity analysis in medical data mining with ROC-based genetic learning. *Third IEEE International Conference on Data Mining*,.

[34] Tuv, E., Borisov, A., Runger, G. and Torkkola, K., 2009. Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination. *The Journal of Machine Learning Research*, 10, pp.1341-1366.
[35] Sarkar, C., Cooley, S. and Srivastava, J., 2014. Robust Feature Selection Technique Using Rank Aggregation. *Applied Artificial Intelligence*, 28(3), pp.243-257.
[36] DASH, M. and LIU, H., 1997. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4), pp.131-156.
[37] Chandrashekar, G. and Sahin, F., 2014. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), pp.16-28.
[38] Ren Diao and Qiang Shen, 2012. Feature Selection With Harmony Search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(6), pp.1509-1523.
[39] Liu, H & Setiono, R 1995, Chi2: feature selection and discretization of numeric attributes. in Anon (ed.), *Proceedings of the International Conference on Tools with Artificial Intelligence.* IEEE, pp. 388-391, Proceedings of the 1995 IEEE 7th International Conference on Tools with Artificial Intelligence, Herndon, VA, USA, 11/5/95.
[40] Najdi, S., Gharbali, A. and Fonseca, J., 2017. Feature ranking and rank aggregation for automatic sleep stage classification: a comparative study. *BioMedical Engineering OnLine*, 16(S1).
[41] Lei, S., 2012. A Feature Selection Method Based on Information Gain and Genetic Algorithm. *2012 International Conference on Computer Science and Electronics Engineering*,.
[42] Hung, L., Alfred, R. and Ahmad Hijazi, M., 2015. A Review on Feature Selection Methods for Sentiment Analysis. *Advanced Science Letters*, 21(10), pp.2952-2956.
[43] Vege, Sri Harsha, "Ensemble of Feature Selection Techniques for High Dimensional Data" (2012).*Masters Theses & Specialist Projects.*Paper 1164.http://digitalcommons.wku.edu/theses/1164
[44] Altidor, W., Khoshgoftaar, T. and Napolitano, A., 2012. Measuring stability of feature ranking techniques: a noise-based approach. *International Journal of Business Intelligence and Data Mining*, 7(1/2), p.80.
[45] R., P., M.L., V. and S., S., 2011. GAIN RATIO BASED FEATURE SELECTION METHOD FOR PRIVACY PRESERVATION. *ICTACT Journal on Soft Computing*, 01(04), pp.201-205.
[46] Kira, K. and Rendell, L., 1992. A Practical Approach to Feature Selection. *Machine Learning Proceedings 1992*, pp.249-256.
[47] Urbanowicz, R., Meeker, M., La Cava, W., Olson, R. and Moore, J., 2018. Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, 85, pp.189-203.
[48] Dessì, N., Pascariello, E. and Pes, B., 2013. A Comparative Analysis of Biomarker Selection Techniques. *BioMed Research International*, 2013, pp.1-10.
[49] Draper, B., Kaito, C. and Bins, J., 2003. Iterative Relief. *2003 Conference on Computer Vision and Pattern Recognition Workshop*,.
[50] Sarhrouni, E., Hammouch, A. and Aboutajdine, D., 2012. Application of Symmetric Uncertainty and Mutual Information to Dimensionality Reduction of and Classication Hyperspectral Images. *International Journal of Engineering and Technology (IJET)*, 4(5), pp.268-276.
[51] Witten, I. and Frank, E., 2005. *Data mining*. San Francisco, Calif.: Morgan Kaufmann.
[52] Kwak, N., Kim, C. and Kim, H., 2008. Dimensionality reduction based on ICA for regression problems. *Neurocomputing*, 71(13-15), pp.2596-2603.
[53] Denton, E., Holden, M., Christ, E., Jarosz, J., Russell-Jones, D., Goodey, J., Cox, T. and Hill, D., 2000. The Identification of Cerebral Volume Changes in Treated Growth Hormone-Deficient Adults Using Serial 3D MR Image Processing. *Journal of Computer Assisted Tomography*, 24(1), pp.139-145.
[54] Holden, M., Denton, E., Jarosz, J., Cox, T., Studholme, C., Hawkes, D. and Hill, D., 1999. <title>Detecting small anatomical change with 3D serial MR subtraction images</title>. *SPIE Proceedings*,.
[55] Otte, M., 2001. Elastic registration of fMRI data using Bezier-spline transformations. *IEEE Transactions on Medical Imaging*, 20(3), pp.193-206.
[56] Sonal Singh, Shyam S Choudhary and S. Bhavishya, 2018. Feature Selection Effects on Classification Algorithms. *International Journal of Engineering Research and*, V7(02).
[57] Sujatha, M. and Lavanya Devi, G., 2013. Feature Selection Techniques using for High Dimensional Data in Machine Learning. *IJERT*, 02(09), pp.2209-2216.
[58] Hall, M., 1999. *Correlation-based Feature Selection for Machine Learning*. Doctor of Philosophy.
[59] Bolón-Canedo, V., Sánchez-Maroño, N. and Alonso-Betanzos, A., 2012. A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, 34(3), pp.483-519.
[60] Go, A.; Bhayani, R. & Huang, L. (2009), 'Twitter Sentiment Classification using Distant Supervision', *Processing* , 1--6.
[61] Foroozan Yazdani, S., Tan, Z., Kakavand, M. and Mustapha, A., 2018. NgramPOS: a bigram-based linguistic and statistical feature process model for unstructured text classification. *Wireless Networks*, 28(3), pp.1251-1261.
[62] Prusa, J., Khoshgoftaar, T. and Dittman, D., 2015. Impact of Feature Selection Techniques for Tweet Sentiment Classification. FLAIRS Conference 2015, pp.1-6.
[63] Marcot, B. and Hanea, A., 2020. What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?. *Computational Statistics*, 36(3), pp.2009-2031.