

ENHANCED FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT CLASSIFICATION USING FEATURE EXTRACTION, FEATURE SELECTION AND MACHINE LEARNING CLASSIFIERS

CHANDANI KUMARI¹, MAHENDRA KUMAR GOURISARIA^{1*}, JUNALI JASMINE JENA¹

¹School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar-751027, Odisha, India

E-mail: ¹mkgourisaria2010@gmail.com

ABSTRACT

Classification of software requirement is very important in the software development field. Categorization of non-functional and functional requirements at early stage help to improve the design phase, quality of the software and reduce the probability of risks. Manual classification of software requirement is cumbersome due to huge volume of data and customer requirements that lead to error & delay in the development of the software. To resolve this issue, feature selection techniques such as Chi2 and PCA combined with feature extraction techniques such as TF-IDF, BOW, Fast-Text, GLOVE and WORD-2-VEC have been simulated with machine learning models such as NB, SVM, LR, D-Tree, Stacking, K-NN, Voting, GB, RF and AdaBoost to automatically classify the software requirements. In multi-class classification, Fast-Text and Chi2 simulated with Stacking and Voting classifiers achieves an accuracy of approximately 90%. In binary classification, TF-IDF simulated with SVM, Stacking and Voting classifiers reached an accuracy of 98%. Hence, supremacy of ensembles models like Stacking and Voting classifiers has been observed in automated software requirements classification task.

Keywords: *Functional Requirements, Non-Functional Requirements, Feature Extraction, Feature Selection, Software Requirements Classification*

1. INTRODUCTION

Software requirement classification has a crucial role in software development. This research is focused on categorization of user requirements into functional and non-functional given in Software Requirements and Specification [SRS]. Functional requirements represent “what a system software should do” whereas non-functional requirements represent “Quality aspects of software system”. Software requirement act as interface between developer and customer [1]. It helps to optimize the gap between developer and customer and also provide support to the developer for clearly understanding the requirements. Furthermore, it helps in creating the correct test cases, for software verification and validation accurate classification of requirements helps in tracking the progress of the project and helps in ensuring that final software meets the customer requirements.

Functional requirements represent “System behaviors”. It deals with the system particular features, tools and tasks [2]. It describes the input to the software such as login module, output of the

software such as about website, behaviors of the software such as login credentials are correct or not and functions of the software such as printing the bill. It provides the developer a clear idea about what needed to be developed and helps the customer to validate whether software meets the requirements or not. A proper SRS document with accurate classification of functional and non-functional requirements leads to reduction of error in the later phases of software development.

Non-functional Requirements represents the “System quality and performance”. It checks various qualitative aspects of the software such as speed, flexibility, scalability, maintainability etc. [3]. Basically, non-functional requirements are classified into 12 categories of the software system such as Security, Reliability, Portability, Performance, Efficiency, Usability, Supportability, Scalability, Availability, Maintainability, Operability, and Compatibility.

In this paper, the authors investigate the impact of both binary classification and multi-class

classification of Non-Functional Requirements [NFRs] and Functional requirements [FR] in three datasets. To resolve the challenges of manual classification, the automated classification has been performed by using text vectorization tools such as Fast-text, Bag of Words [BOW], WORD-2-VEC, TF-IDF and GLOVE with feature extraction tools such as Principal Component Analysis [PCA] and Chi-Squared [Chi2] simulated with machine learning models. Moreover, two new datasets have been used, on which no work has been done till now in the literature. This enhances the novelty of this work.

The main contribution of this research work as follows:

- ML classifier such as LR, NB, GB, D-Tree, K-NN, Stacking, RF, Voting, AdaBoost and SVM with feature extraction techniques such as Tf-Idf, Glove, Bow, Word-2-Vec and Fast Text combined with feature selection techniques such as PCA and Chi2 has been used for binary classification and multi-class classification of NFR and FR.
- Three datasets such as 'Software_Requirement.csv' that consists of 977 rows and 2 columns, 'Software_Risk_Prediction.csv' that consists of 299 rows and 14 columns, 'New_Software_Requirement.csv' that consists of 53 rows and 2 columns has been used where two datasets, 'software risk prediction.csv' and 'new_software_requirement.csv' have not been used in any research work till now, which highlights the nobility of this paper.
- Performance of the ensemble models such as SVM, Stacking and Voting classifiers was recorded to be higher than the other classifiers. Also, achievement of 99% accuracy in binary classification and 87% in multi-class classification, is also recorded as higher than the other similar works in the literature.

The rest of the paper is structured as follows: Section-2 demonstrates about the previous research works named as "Related Work". Section-3 demonstrates about the feature extraction methods, feature selection techniques and ML classifiers used in this paper, named as "Materials and Methods". Section-4 demonstrates about the derived results named as "Result and Discussion". Section-5 named as "Conclusion and Future work" demonstrates

about the future research work followed by references.

2. LITERATURE REVIEW

This section reviews various works in the field of categorization of Non-Functional and Functional Requirements. Binkhonain and Zhao [4] (2023) investigated HC4RC, a hierarchical ML model technique used to categorize software requirements while looking into two major issues such as high-dimensional low-sample size data and class imbalance which can be optimized by using hierarchical classification, semantic-role-based feature selection and dataset decomposition.

Muhammad bin Saleem et al. [5] (2025) demonstrated about a comparative investigation of three few-shot learning algorithms such as MAML, Matching Networks and Prototypical Networks to automate the software requirement classification by using promise dataset. By analyzing the result, it was found that MAML achieved an accuracy of 77% whereas Matching Network gave poor result with an accuracy of 49% only. Highest accuracy of 89% was given by Prototypical Network. Similarly, Alhoshan et al. [6] (2023) used a Zero-Shot Learning technique for binary and multi-class classification of software requirements and achieved an accuracy of 66% to 80%. It used the concept of complex label configuration and domain particular LLMs.

Vincenzo De Martino and Fabio Palomba [7] 2024 represented the "Graph-Mix" approach that was very effective semi-supervised method for categorizations in software requirements. It used two types of datasets i.e. 'security_requirement.csv' which achieved an accuracy of 88% whereas 'promise dataset' achieved an accuracy of 55%. Also, Rajat Izhar et al. [8] (2025) investigated a new technique in classification of software requirements i.e. machine learning model combined with hybrid-rule to reduce the ambiguity and to increase the accuracy. In this, 'promise dataset' was used and TF-IDF combined with Random Forest gave a good performance with an F1-score of 92%, recall of 89% and accuracy of 93%. Canedo and Mendes [9] (2020) used feature extraction and feature selection technique such as TF-IDF, CHI2 and BOW with machine learning models for software requirements classification. TF-IDF combined with RF gave best accuracy of approximately 90% in classification of FR and NFR. Similarly, Jindal et al. [10] (2021) also demonstrated an idea about mining of non-functional requirement from NLP SRS document.

Rahman and Nayem [11] (2023) used four vectorization methods such as TF-IDF, Chi2, Bow and Hashing vectorizer combined with fifteen ML models for non-functional requirement classification. The TF-IDF combined with Linear support vector classifier achieved best result of F1-score of approximately 82%. Similarly, K. kaur and P.Kaur [12] (2023) used the concept of deep learning methods such as CNN-BiLSTM for categorization of functional and non-function requirement using 'promise dataset'. Recurrent and convolutional architectures were merged to catch the clear local range and local patterns semantics easily. The algorithm reached an accuracy of 95%.

Vrutik Patel et. al [13] (2023) compared the vectorization techniques such as TF-IDF and BOW with four ML models such as NB, SVM, D-Tree and K-NN for classification of software requirements. TF-IDF combined with SVM which achieved an accuracy of approximately 90%. Also, Gaith Y Quba et. al [14] (2021) suggested an idea to classify software requirements by using feature extraction techniques such as Bow combined with ML models such as K-NN and SVM by using 'promise dataset'. This method achieved F1-scores of 72% in multi-class and 90% in binary classification. This paper also investigated that SVM combined with BOW provided the best result in the classification of NFR and FR. The Table-1 represents the literature survey of the work has been done earlier.

Table 1: Review of literature survey

Year	Method/Model	Dataset Used	Performance
2023 [4]	Hierarichical ML model such as high-dimensional low sample size and class imbalance	Promise_exp dataset used	Accuracy:63% F1-score:51%[10-fold] Accuracy:64% F1-score:50%[p-fold]
2025 [5]	Shot learning algorithm such as MAML, Matching Networks and Prototypical Networks	Promise_exp dataset used	Accuracy:77% [MAML] Accuracy:49% [Matching Network] Accuracy:89% [Prototypical Networks]
2023 [6]	ML/DL both classifier is used named Zero-Shot Learning Technique	Custom Non-Functional Requirements dataset (20 NFR groups)	Accuracy: 66%
2024 [7]	Graph-Mix approach	Security requireemnt.csv	Accuracy: 88%
2025 [8]	Machine Learning model combined with Hybrid-rule based model Feature Extraction Technique: TF-IDF ML Classifier: Random Forest, Clustering and Sentimental Analysis	Promise_exp dataset used	F1-Score: 92% Recall: 89% Accuracy: 93%
2020 [9]	K-NN, MNB, LR, SVM Feature Extraction: BOW, TF-IDF Feature Selection: CHI2	Promise_exp dataset used	F1-score:0.91(binary), 0.78(general),0.74(NF)
2021 [10]	Seven ML Classifier such as D-Tree, GNB, MNB, K-NN, BNB, SGDSVM, and SVM + four feature techniques TF-IDF, BOW, CHI2 & Hashing	Promise datasets (11 NFR categories, 625SRStatements)	Recall:61%, Precision:66%, Accuracy:76%, F1-score:61%
2023 [11]	Machine Learning such as SVM, LR, K-NN, D-Tree, NB with TF-IDF & BOW	Custom datasets of 625 software requirements (12 NFR subclasses +FR)	Accuracy:71.3%
2023 [12]	Custom CNN architectorCHI2+TF-IDF, RF, AdaBoost, XGBoost, LR +Word2Vec	1,400 manually labelled SRS requirements	F1-Score:74%, Recall:65%, Precision:79%
2021 [13]	Seven ML Classifier such as D-Tree, GNB, MNB, KNN, MNB, SGDSVM, and SVM + four feature techniques TF-IDF, BOW, CHI2 & Hashing	Promise datasets (11 NFR categories ,625SRStatements)	Accuracy:76%, F1-score:61% Recall:61%, Precision:66%,

2023 [14]	Machine Learning such as with K-NN, D-Tree, MNB, LR & SVM Vectorization Technique such as BOW & TF-IDF	Custom datasets of 625 software requirements (12 NFR subclasses +FR)	Accuracy: 77%
-----------	--	--	---------------

3. MATERIALS AND METHODS

This section gives a complete explanation of the technologies and plans used in this research work.

3.1 Dataset Description

In this paper three datasets have used 'Software_Requirement.csv' that consists of 977 rows and 2 columns [15], 'Software_Risk_Prediction.csv' that consists of 299 rows and 14 columns [16], 'New_Software_Requirement.csv' that consists of 53 rows and 2 columns [17]. All the datasets have been used for the categorization of non-functional requirements (NFR) and Functional Requirements (FR). In this work two types of classification have been done. First one is binary classification where the requirements have been classified as FR and NFR requirements. Second one is multi-class classification where the requirements have been classified as FR and multiple classes of NFR.

3.2 Feature Extraction Technique

Bag of Words (BOW): It is a word-presentation methods that transform phrases into numerical vectors based on text frequency [18]. It can be calculated by the formula given in Equation 1 where *doc* represents documents and *wd* represents word:

$$Fr(wd_j, doc_i) = \sum_{k=1}^{|doc_i|} ||(doc_i[k] = wd_j) \quad (1)$$

Term Frequency-Inverse Document Frequency (TF-IDF): It calculates how crucial a word present in SRS document is with respect to the whole documents. IDF optimize the weightage of similar words and improves the weightage of unique words [19]. It can be calculated by the formula given in Equation 2 where *fr* represents frequency, *wd* represents word, *WDoc* represents Whole documents, *doc* represents documents and *tm* represents term of Words.

$$TF-IDF(wd, Doc) = \left[\frac{fr(wd, doc)}{\sum_{t \in Doc} fr(tm, doc)} \right] * \left[\log \left(\frac{1 + WDoc}{1 + DF(wd)} \right) + 1 \right] \quad (2)$$

Word-2-Vectorization (Word-2-Vec): It transform text into continuous numerical vector and catches context and semantic ambiguity in a document [20]. It can be calculated by the given formula in Equations 3, 4 and 5 where *ED* represents embedding dimensions and *VS* represents Vocabulary size.

$$Win \in R^{(ED * VS)} \quad (3)$$

$$Wout \in R^{(ED * VS)} \quad (4)$$

$$\text{Word Vector} = \text{row of Win} \quad (5)$$

Glove: It is a text-extraction technique that absorbs term embeddings using global term occurrence frequency from the entire documents [21]. It can be calculated by the given in Equations 6 and 7 where *Y_{ij}* represents occurrence frequency, *B_i* and *B'* represents Bias term, *Wd_i* represents word vector and *Wd'* represents context vector.

$$\text{Prob}_{ij} = Y_{ij} / Y_i \quad (6)$$

$$\text{GloveFr} = Wd_i^T Wd_j' + B_i + B_j = \log(Y_{ij}) \quad (7)$$

Fast-Text: It is similar to Word-2-Vec and also increases text vectorization by using sub-word data breaking the terms into letter n grams [22]. It can be calculated by the given Equation 8 where *evg* represents embedding Vector, *Avgw* represents sum of all n gram vector and *ng* represents n-gram.

$$Avgw = \sum_{ng \in G(wd)} evg \quad (8)$$

3.3 Feature Selection Technique

Chi-Squared (Chi2): It evaluates whether the frequency of a class and terms are independent or not [23]. It can be calculated by the given Equation 9. *D* is doc where word *te* appears and belong to class *B*, *C* doc where *te* appears but class is not *B*, *B* is doc where *t* does not appear but class is *B*, *A* doc where *t* does not appear, class is not *B* and *doc* is entire requirements and *co* denotes the category.

$$\chi^2(te, co) = \frac{doc(DA - CB)^2}{(D+B)(C+A)(D+C)} \quad (9)$$

Principal Component Analysis [PCA]: It reduces a dimension of a features that convert high to low dimension of datapoints by using the concept of

variance [24]. It is calculated by given Equation 10 where pm is projection matrix, D is number of features, σ is standard deviation and μ is mean.

$$PCA_{op} = D - \mu / \sigma * pm \quad (10)$$

3.4 Machine Learning Algorithm

Support Vector Machine (SVM): It helps to identify an optimum decision boundary that increases the line between dissimilar groups and it also gives the best result with the high dimensionality data points [25].

Logistic Regression (LR): It helps to investigate the probability of a group in categorization of sub-groups [26].

Stacking: It combines the results of multiple models and predict the final result, that's why it is known as Ensemble models [27].

Decision Tree (D-tree): It breaks datapoints into a smaller part by taking the feature as input and predict the output [28]. It is also known as tree-based group and it calculate entropy and gini index for the best result.

K-Nearest Neighbour (K-NN): It investigate new feature points by using a most important features by calculating its K-value [29]. It can be calculated by using Euclidean distance metrics.

Voting: It combines the result of soft and hard computing to give the best result [30] and it is also a robust algorithm.

Naïve Baye's (NB): It gives the result on the basis of datapoints are independent or not [31]. It calculates the result by using the concept of probability.

Random Forest: It gives the result by merging more than one decision tree in which all trees are with random subgroups to resolve the issue of overfitting and increases the accuracy [32].

Gradient Boost (GB): It merge several weak learners sequentially to make a strong predictive classifier [33].

Adaptive Boost (AdaBoost): It enhances classification accuracy and performs well with HD data [34].

3.5 Flow-Chart

Figure-1 represents architecture diagram of classification of NFR and FR of software requirements. Performance measure is used to evaluate the performance of classification. Several ML classifiers, feature extraction technique and feature selection techniques can be used.

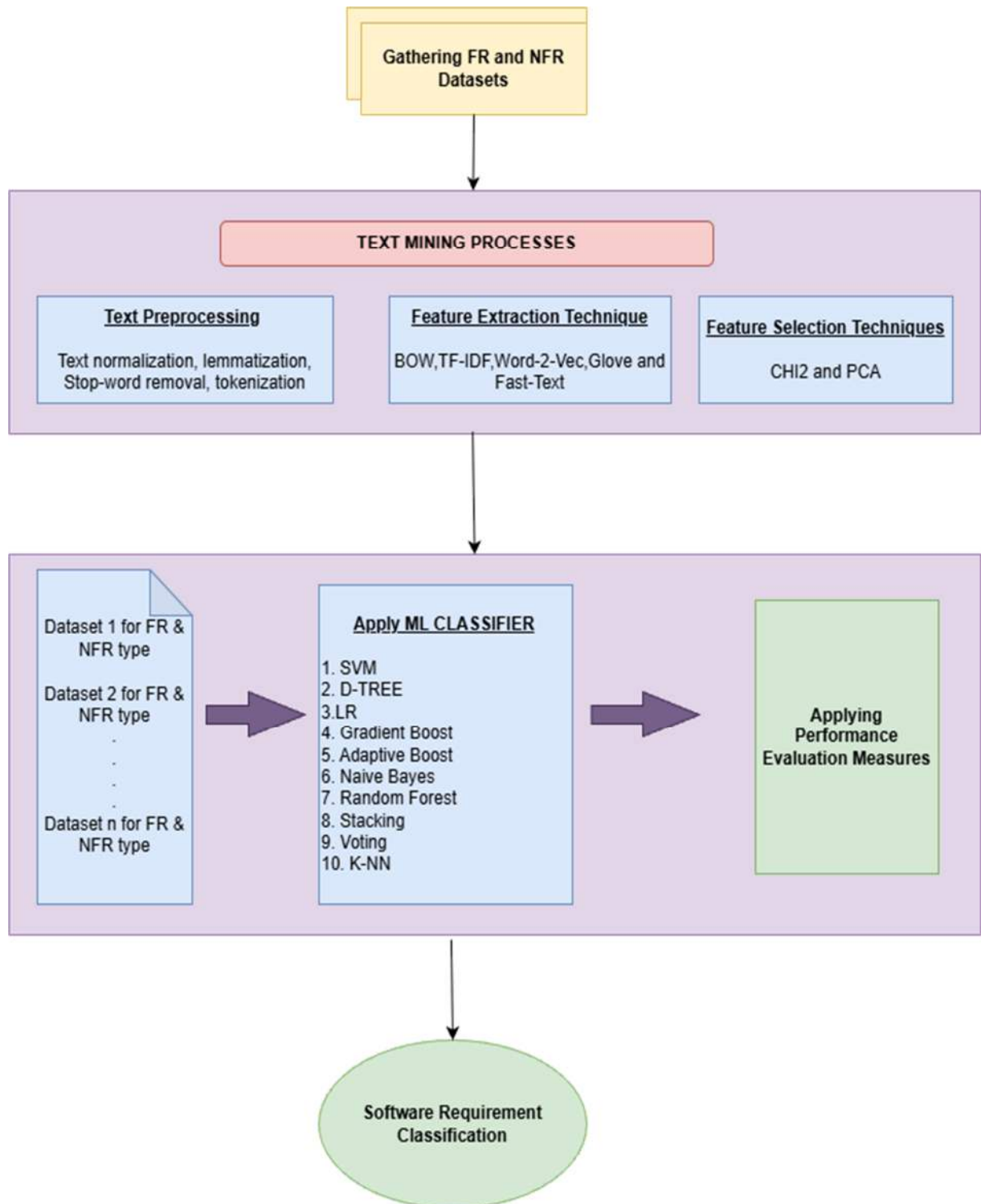


Figure 1: Architecture diagram of classification of non-Functional and Functional software Requirements

4. RESULTS AND DISCUSSION

The experiments were carried out on a workstation running Windows 11 Home OS and with an Intel 12th Gen Intel(R) Core (TM) i7-1255U 1.70 GHz CPU and 16 GB RAM. Scikit-Learn modules allowed for the usage of Python 3 with Anaconda Spyder. The results and the discussion are represented in this segment followed by short description. Several ML algorithms such as Logistic Regression (LR), Naïve Bayes (NB), Decision-Tree (D-Tree), Support Vector Machine (SVM), Gradient-Boost (GB), Adaptive Boost (AdaBoost), K-Nearest Neighbour (K-NN), Stacking, Random Forest (RF) and Voting were used. K-NN, Stacking, Random Forest (RF) and Voting were used.

Table-2 demonstrates the categorization of Multi-Class NFR and FR with dataset-1 'Software_Requirement.csv'. In this, Fast-Text combined with PCA & Chi2 provides best performance combined with SVM, Voting, RF, and Stacking with a recall of 89%, accuracy of 88%, F1-score of 87%, and precision of 89%. AdaBoost, NB, D-Tree were poor performers due to weak nature, dense vectors and overfittings. Word-2-Vec and Glove gave average performance with F1-score of 86%. Though some classifiers are having 90% accuracy but their precision, recall and F1-score is low, that's why SVM, Voting, RF and Stacking have been chosen as best classifiers.

Table-2: Results obtained after simulation with datasets-1 for multi-class classification of FR AND NFR.

Feature_Extraction	Feature_Selection	ML Classifier	F1-Score	Recall	Accuracy	Precision
BAG OF WORDS	BOW	Voting	0.77	0.78	0.90	0.76
		SVM	0.77	0.78	0.90	0.76
	BOW +CHI2	Voting	0.77	0.78	0.90	0.76
		GB	0.77	0.78	0.90	0.76
	BOW+PCA	LR	0.70	0.71	0.90	0.69
		Voting	0.70	0.71	0.90	0.69
		SVM	0.70	0.71	0.90	0.69
TERM-FEREQUENCY-INVERSE DOCUMENT FREQUENCY	TF-IDF	LR	0.71	0.70	0.90	0.69
		SVM	0.71	0.70	0.90	0.69
	TF-IDF +CHI2	Voting	0.77	0.78	0.90	0.76
		GB	0.77	0.78	0.90	0.76
	TF-IDF+PCA	LR	0.71	0.70	0.90	0.69
		Voting	0.71	0.70	0.90	0.69
		GB	0.71	0.70	0.90	0.69
WORD-2-VECTORIZATION	WORD-2-VEC	Stacking	0.84	0.83	0.85	0.84
	WORD-2-VEC +CHI2	Stacking	0.85	0.84	0.86	0.85
	WORD-2-VEC+PCA	Stacking	0.85	0.84	0.86	0.85
GLOVE	Glove	Stacking	0.85	0.84	0.86	0.85
	GLOVE +CHI2	Stacking	0.86	0.85	0.87	0.86
	GLOVE +PCA	Stacking	0.86	0.85	0.87	0.86
FASTTEXT	FASTTEXT	Stacking	0.87	0.86	0.88	0.87
	FASTTEXT +CHI2	Stacking	0.88	0.87	0.89	0.88
	FASTTEXT +PCA	Stacking	0.88	0.87	0.89	0.88

Graphical analysis has also been used to demonstrate this research. Figure 2.1 demonstrates about the BOW combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 2.2 demonstrates about the TF-IDF combined with two feature selection methods

such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 2.3 demonstrates about the Word-2-Vec combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software

requirements. Figure 2.4 demonstrate about the Glove separately combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 2.5 demonstrates about the Fast-text combined with two feature selection methods such as PCA & Chi2 simulated with ML

classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. It mainly shows the graphical analysis with all feature extraction techniques with feature election techniques simulated with ML classifiers for the clarification of software requirement.

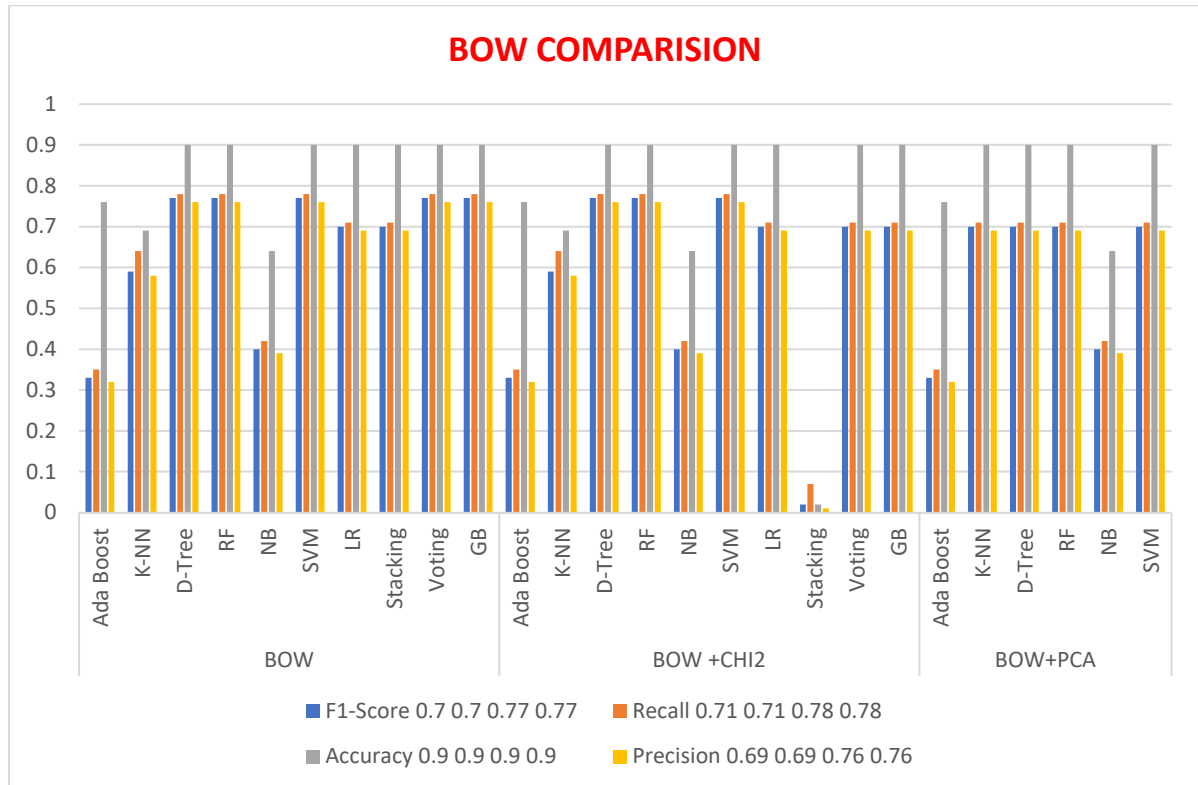


Figure 2.1 Result of BOW in Multi-Classification of FR & NFR with Dataset-1

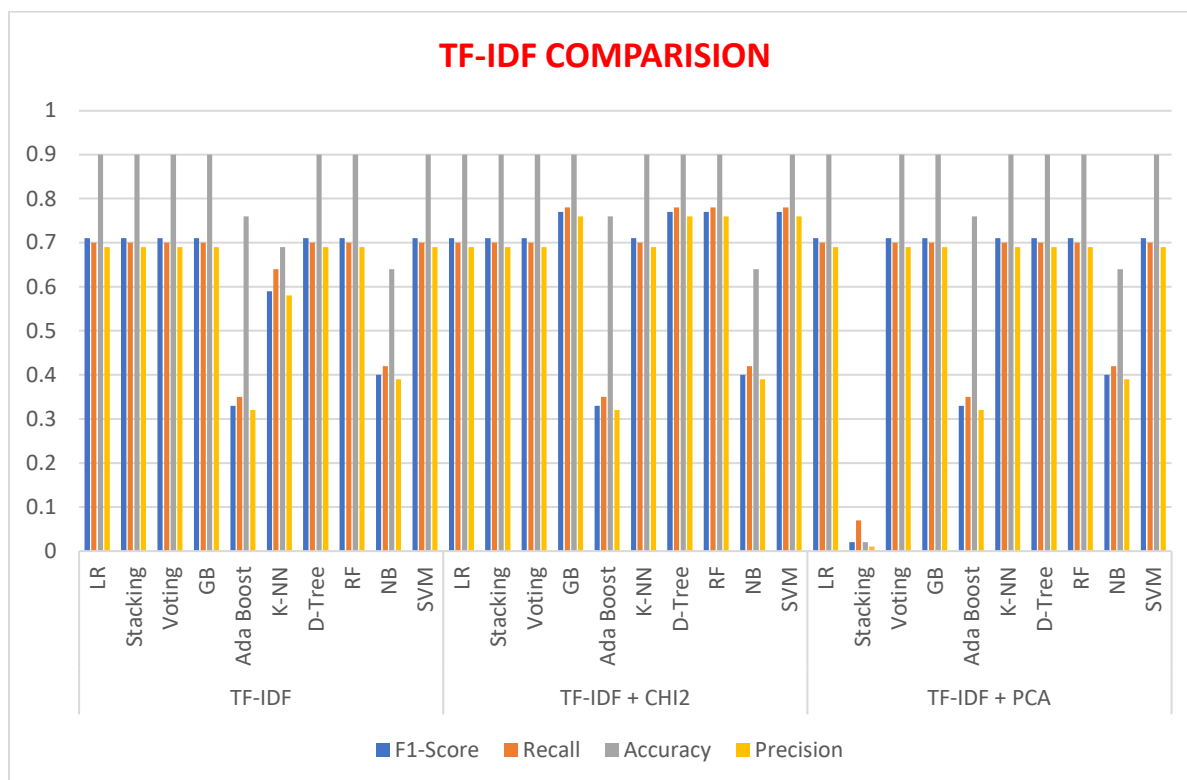


Figure 2.2 Result of TF-IDF in Multi-Classification of FR & NFR with Dataset-1

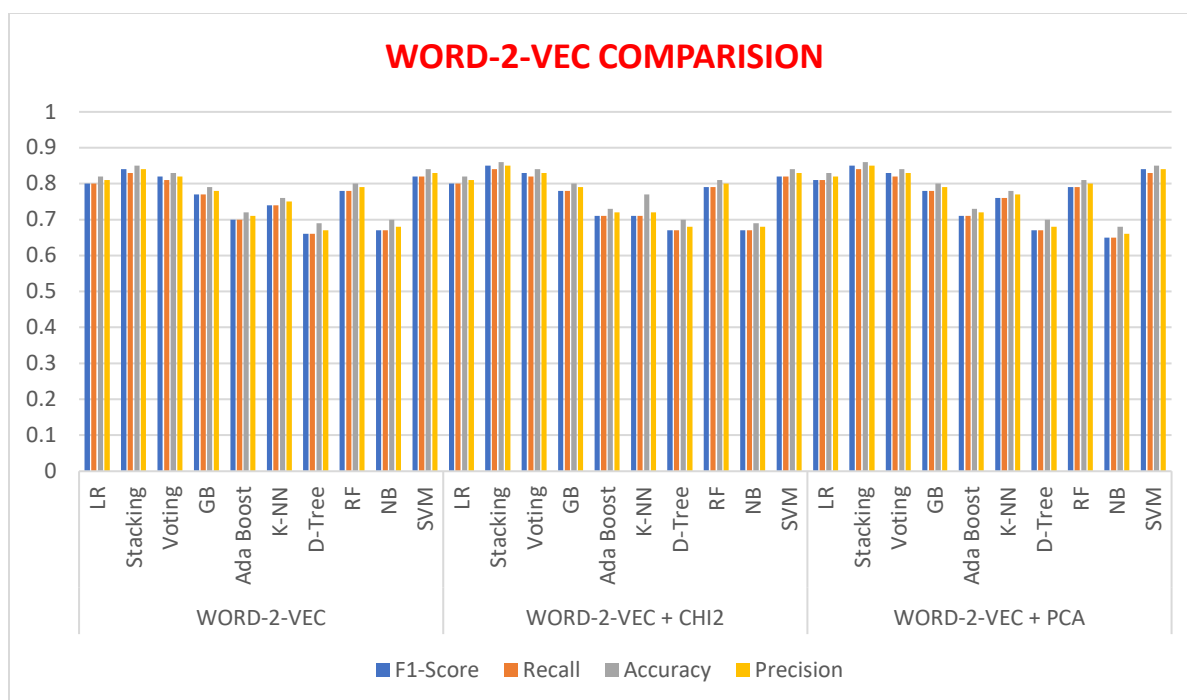


Figure 2.3 Result of WORD-VEC in Multi-Classification of FR & NFR with Dataset-1

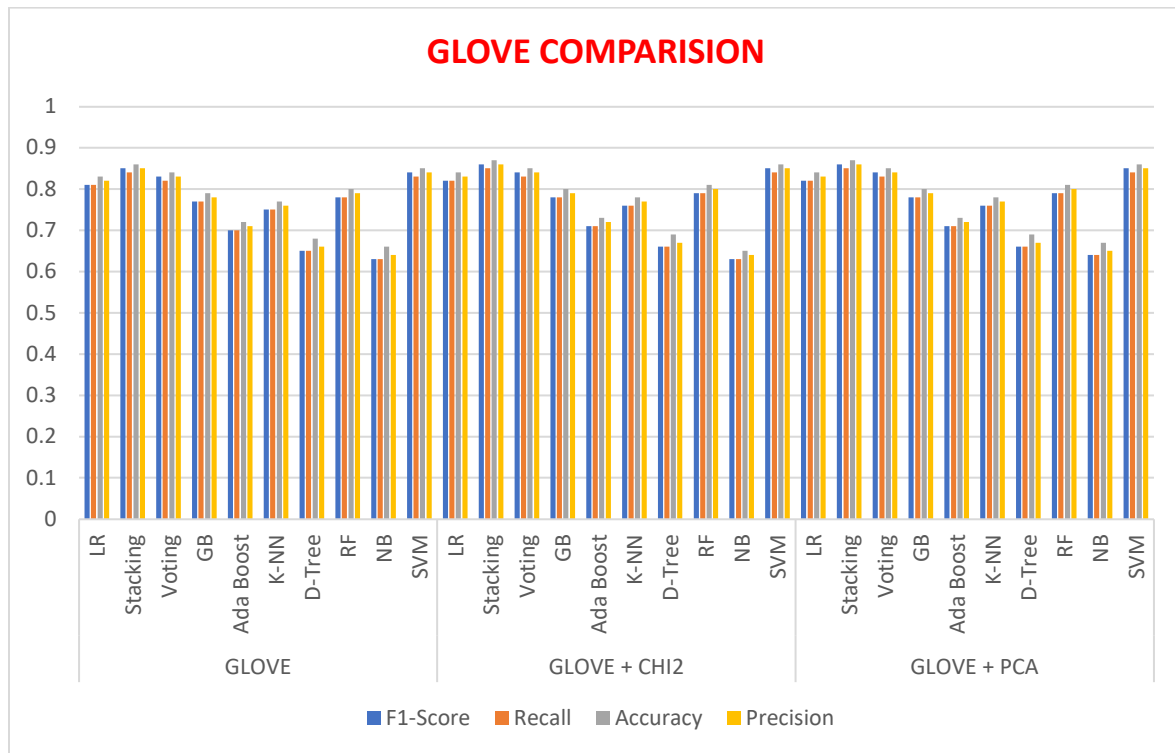


Figure 2.4 Result of GLOVE in Multi-Classification of FR & NFR with Dataset-1

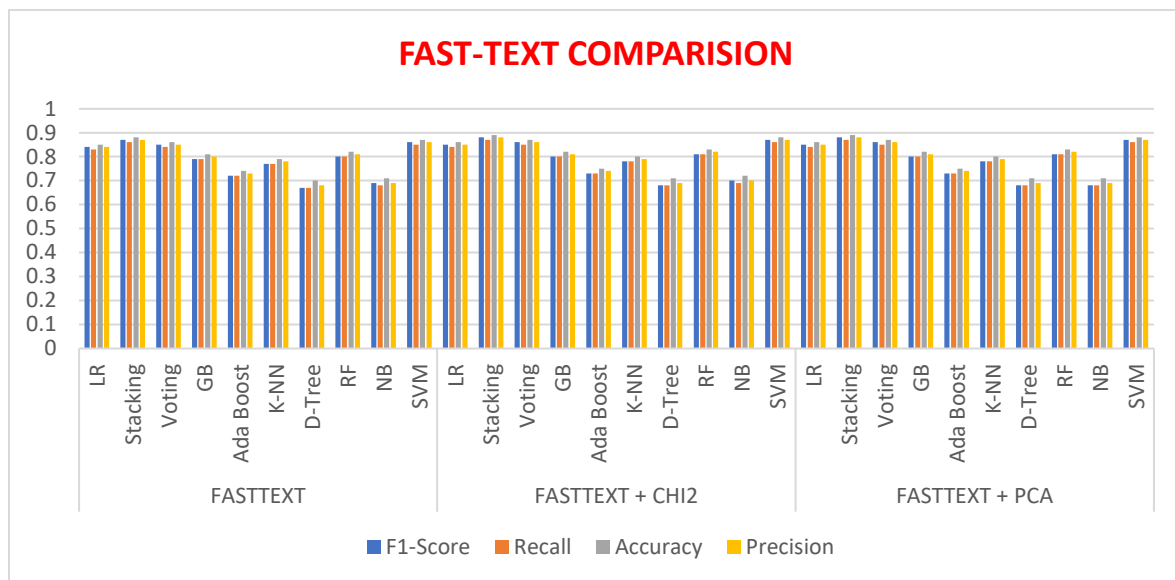


Figure 2.5 Result of FAST-TEXT in Multi-Classification of FR & NFR with Dataset-1

Table-3 demonstrates the categorization of Multi-Class NFR and FR with dataset-2 'Risk_Prediction.csv'. We have investigated that Word-2-vec combined with RF provides the best performance with F1-score and accuracy of 78%. Word-2-Vec combined with ensemble classifiers

also achieved an accuracy of approximately 77%. Average performing models like TF-IDF combined with PCA an SVM has an accuracy of 63% whereas Bow merged with SVM also reached an accuracy of 64%. Fast-Text combined with PCA & Chi2 gave poor performance.

Table-3 Results obtained after simulation with datasets-2 with multi-class classification of FR AND NFR.

Feature_Extraction	Feature_Selection	ML Classifier	F1-Score	Recall	Accuracy	Precision
BAG OF WORDS	BOW	SVM	0.64	0.68	0.68	0.66
	BOW +CHI2	D-Tree	0.51	0.63	0.63	0.56
	BOW+PCA	SVM	0.51	0.54	0.54	0.57
TERM FEREQUENCY-INVERSE DOCUMENT FREQUENCY	TF-IDF	SVM	0.57	0.60	0.60	0.58
	TF-IDF +CHI2	SVM	0.46	0.51	0.51	0.52
	TF-IDF+PCA	SVM	0.63	0.65	0.65	0.67
WORD-2-VECTORIZATION	WORD-2-VEC	Stacking	0.75	0.76	0.76	0.76
	WORD-2-VEC +CHI2	RF	0.77	0.78	0.78	0.78
	WORD-2-VEC+PCA	Voting	0.76	0.77	0.77	0.77
GLOVE	Glove	LR	0.56	0.62	0.62	0.56
	GLOVE +CHI2	LR	0.56	0.62	0.62	0.56
	GLOVE +PCA	GB	0.53	0.61	0.61	0.50
FASTTEXT	FASTTEXT	SVM	0.47	0.51	0.51	0.46
	FASTTEXT +CHI2	GB	0.08	0.20	0.20	0.05
	FASTTEXT +PCA	RF	0.55	0.58	0.58	0.62

Graphical analysis has also been used to demonstrate this research. Figure 3.1 demonstrates about the BOW combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 3.2 demonstrates about the TF-IDF combined with two feature selection methods such as PCA & Chi2 merged with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 3.3 demonstrates about the Word-2-Vec combined with two feature selection methods such as PCA & Chi2 simulated with ML

classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 3.4 demonstrates about the Glove with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 3.5 demonstrates about the Fast-text combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements.

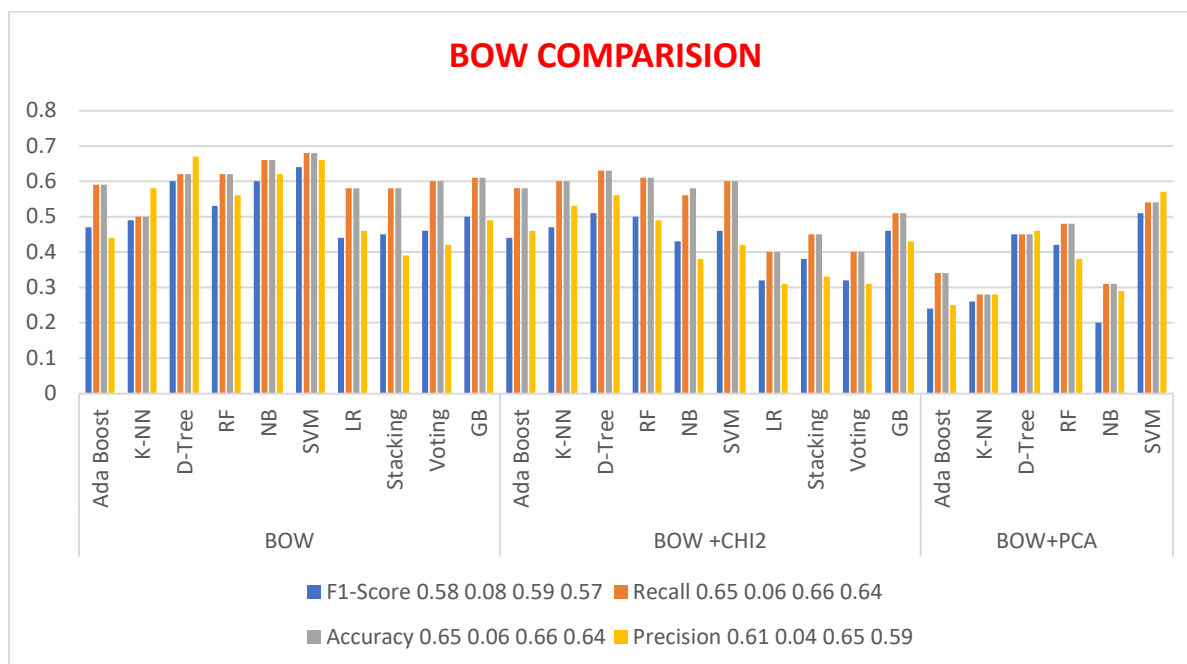


Figure 3.1. Result of BOW in Multi-Classification of FR & NFR with Dataset-2

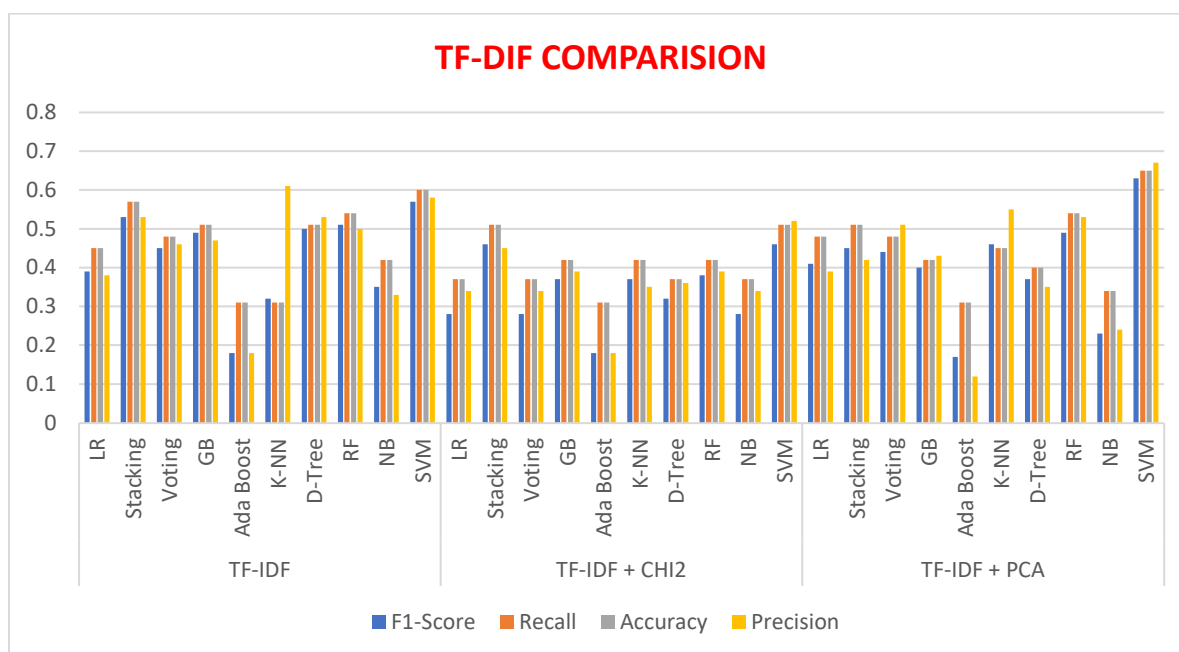


Figure 3.2. Result of TF-IDF in Multi-Classification of FR & NFR with Dataset-2

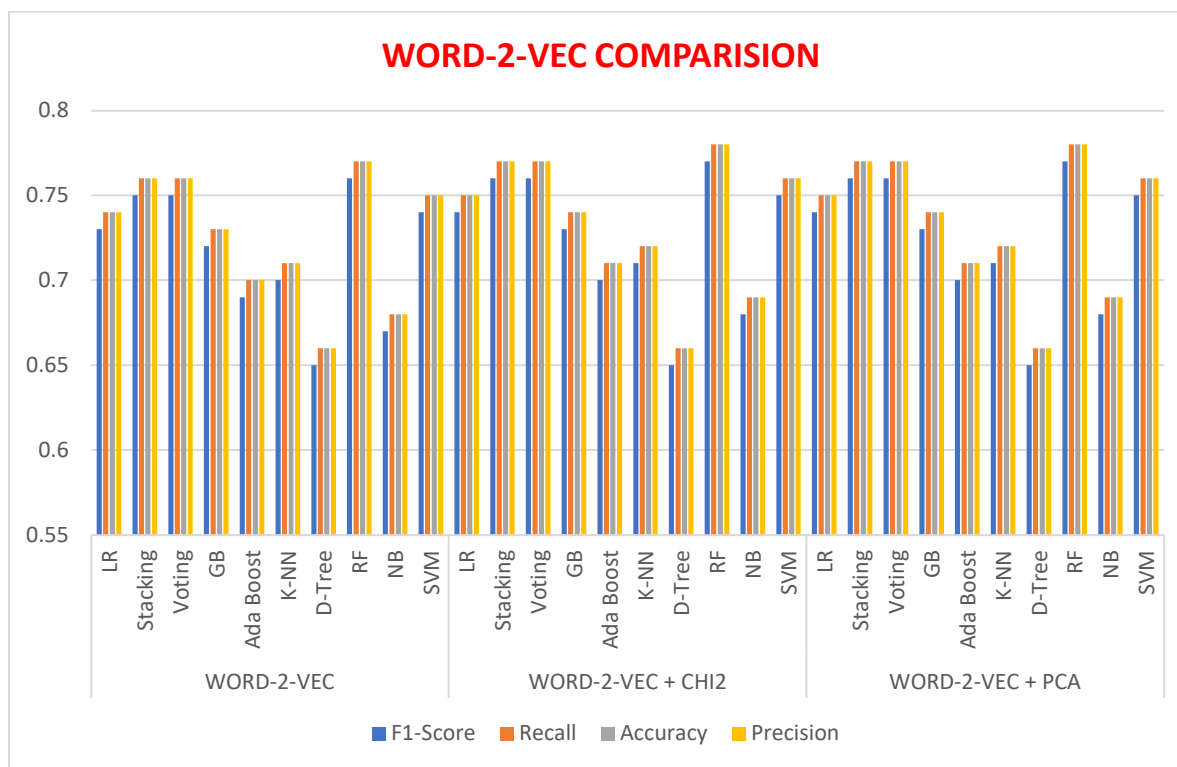


Figure 3.3. Result of WORD-2-VEC in Multi-Classification of FR & NFR with Dataset-2

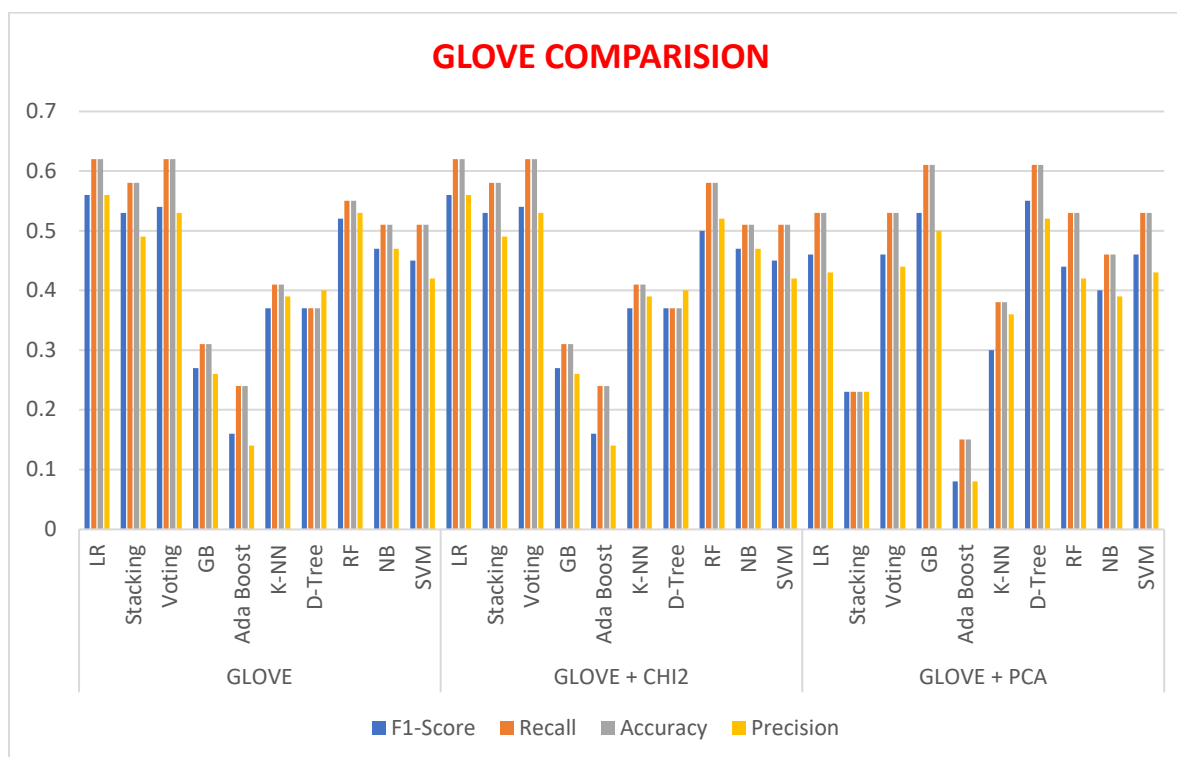


Figure 3.4. Result of GLOVE in Multi-Classification of FR & NFR with Dataset-2

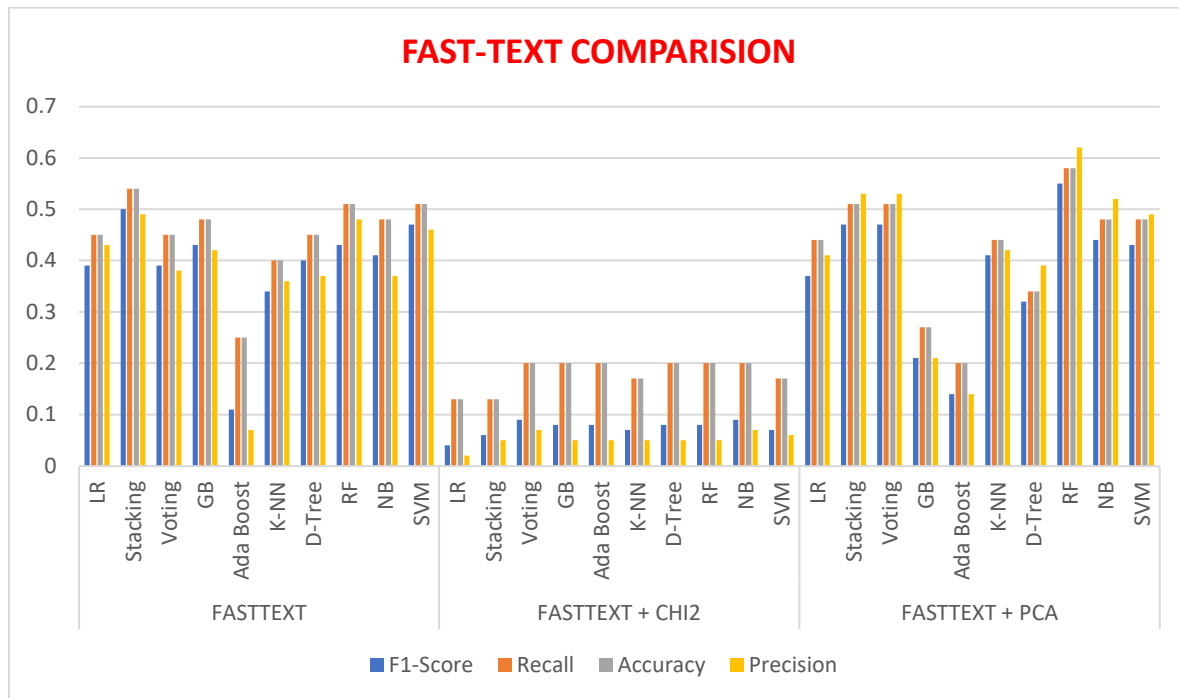


Figure 3.5. Result of FAST-TEXT in Multi-Classification of FR & NFR with Dataset-2

Table-4 demonstrates the categorization of Multi-Class NFR & FR with dataset-3 'New_Software_Requirement.csv'. It has been observed that Glove combined with chi-squared and merged with RF that achieved the best precision of 77%, F1-score of 74% and accuracy of 77%. Fast-

Text combined with PCA & Chi2, Bow with PCA have given poor performance with F1-score of 35% and 30%. Word-2-vec has also poorly performer classifier with F1-score of 37%. TF-IDF combined with Chi2 gives an average performance with an accuracy of 69% and F1-score of 64%.

Table- 4 Results obtained after simulation with datasets-3 with multi-class classification of FR AND NFR.

Feature_Extraction	Feature_Selection	ML Classifier	F1-Score	Recall	Accuracy	Precision
BAG OF WORDS	BOW	RF	0.61	0.69	0.69	0.57
	BOW +CHI2	GB	0.61	0.69	0.69	0.57
	BOW+PCA	SVM	0.46	0.53	0.53	0.42
TERM FEREQUENCY-INVERSE DOCUMENT FREQUENCY	TF-IDF	SVM	0.48	0.53	0.53	0.46
	TF-IDF +CHI2	SVM	0.64	0.69	0.69	0.61
	TF-IDF+PCA	SVM	0.64	0.69	0.69	0.61
WORD-2-VECTORIZATION	WORD-2-VEC	SVM	0.33	0.31	0.46	0.38
	WORD-2-VEC +CHI2	SVM	0.41	0.46	0.46	0.40
	WORD-2-VEC+PCA	RF	0.42	0.46	0.46	0.41
GLOVE	Glove	LR	0.46	0.53	0.53	0.43
	GLOVE +CHI2	RF	0.74	0.77	0.77	0.77
	GLOVE +PCA	D-Tree	0.55	0.61	0.61	0.52
FASTTEXT	FASTTEXT	SVM	0.57	0.61	0.61	0.60
	FASTTEXT +CHI2	LR	0.42	0.53	0.53	0.37
	FASTTEXT +PCA	SVM	0.57	0.61	0.61	0.60

Graphical analysis has also been used to demonstrate this research. Figure 4.1 demonstrates about the BOW combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 4.2 demonstrates about the TF-IDF combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 4.3 demonstrates about the Word-2-Vec combined with two feature selection methods such as PCA & Chi2 simulated with ML

classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 4.4 demonstrates about the Glove combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 4.5 demonstrates about the Fast-text combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements.

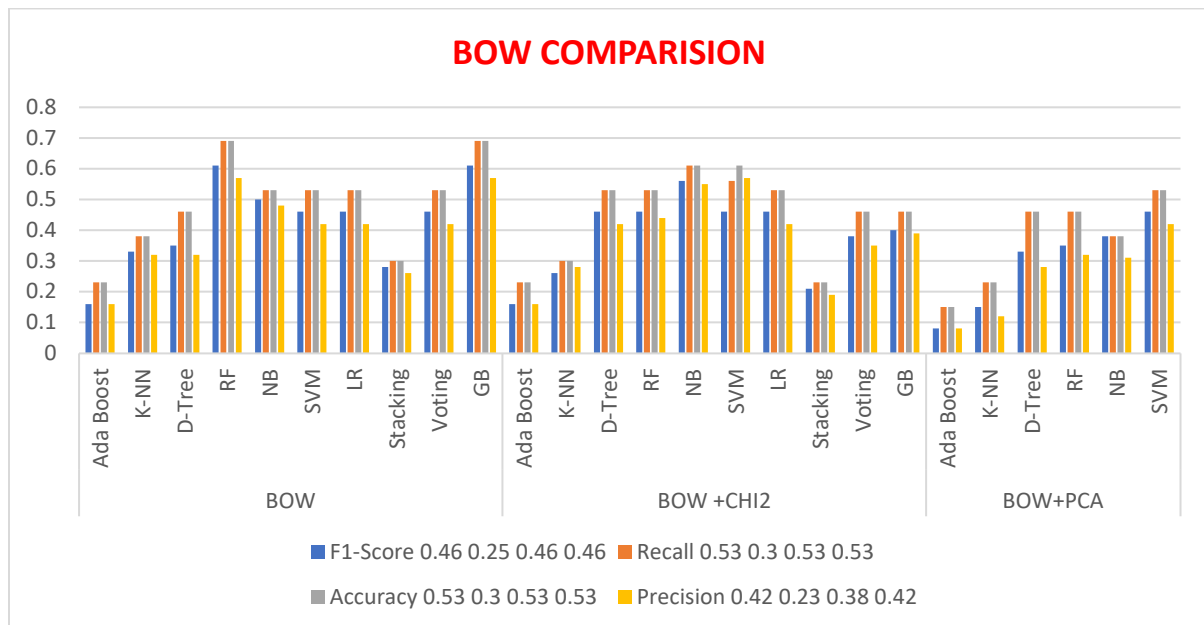


Figure 4.1. Result of BOW in Multi-Classification of FR & NFR with Dataset-3

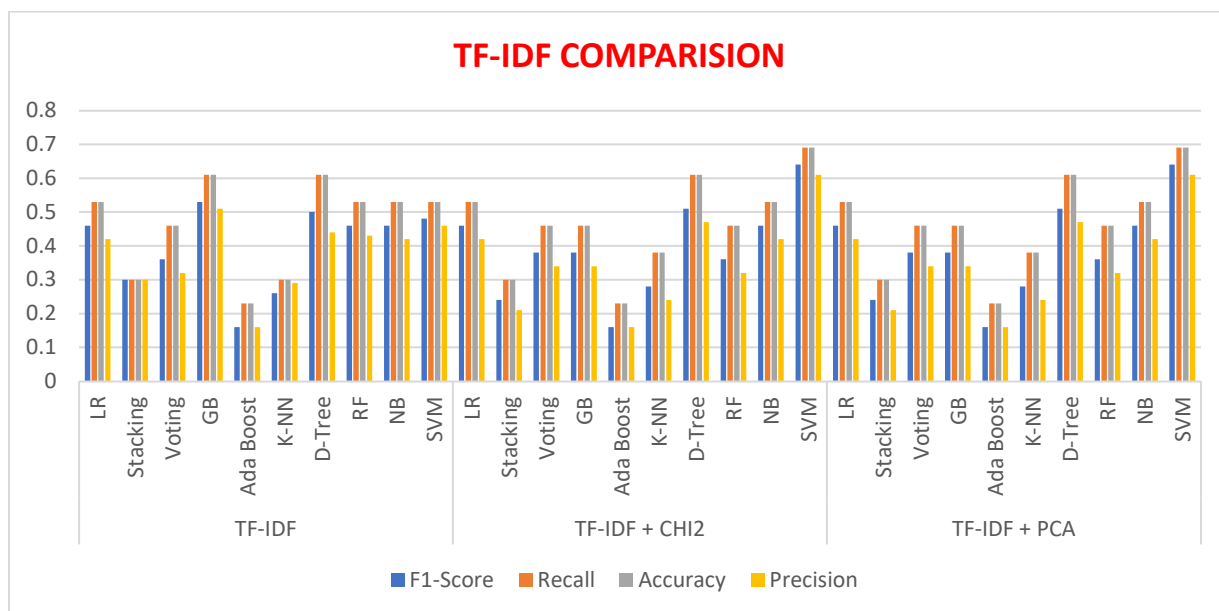


Figure 4.2. Result of TF-IDF in Multi-Classification of FR & NFR with Dataset-3

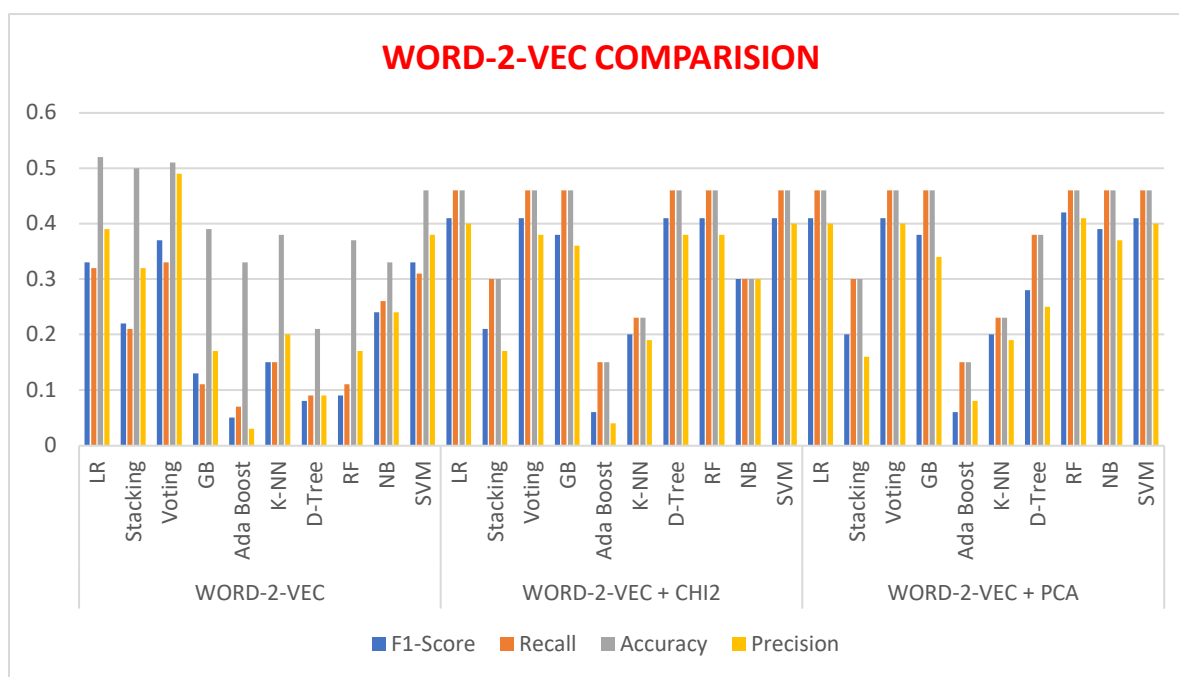


Figure 4.3. Result of WORD-2-VEC in Multi-Classification of FR & NFR with Dataset-3

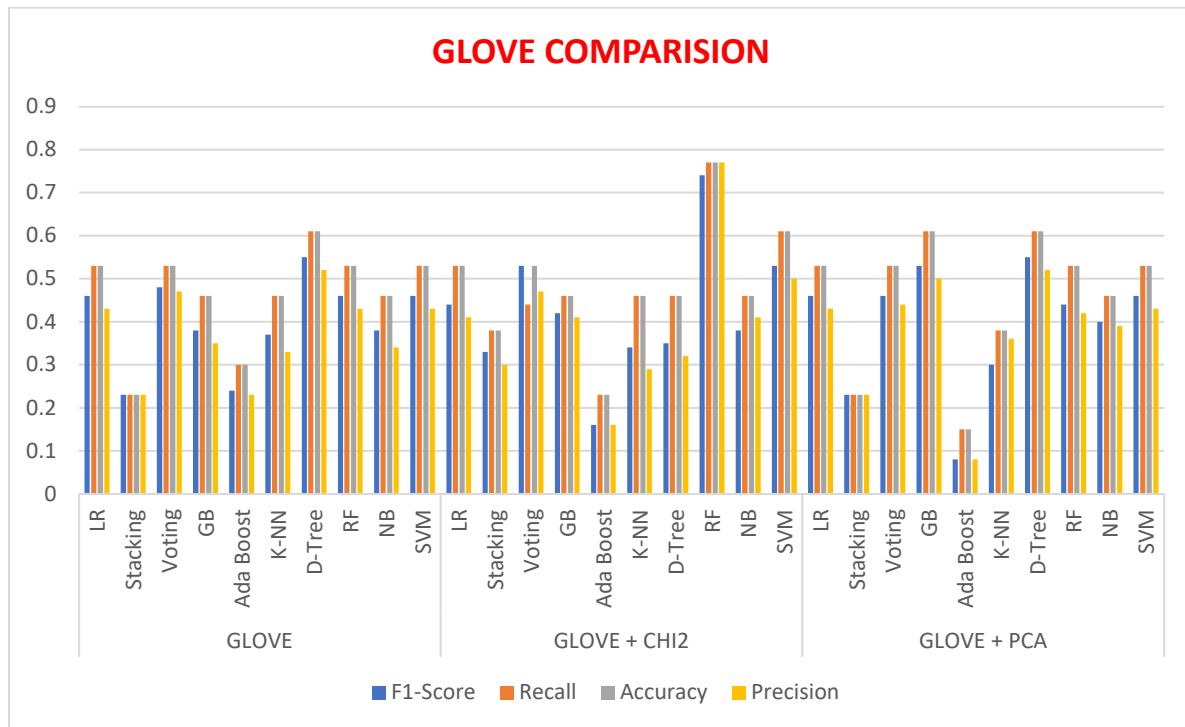


Figure 4.4. Result of GLOVE in Multi-Classification of FR & NFR with Dataset-3

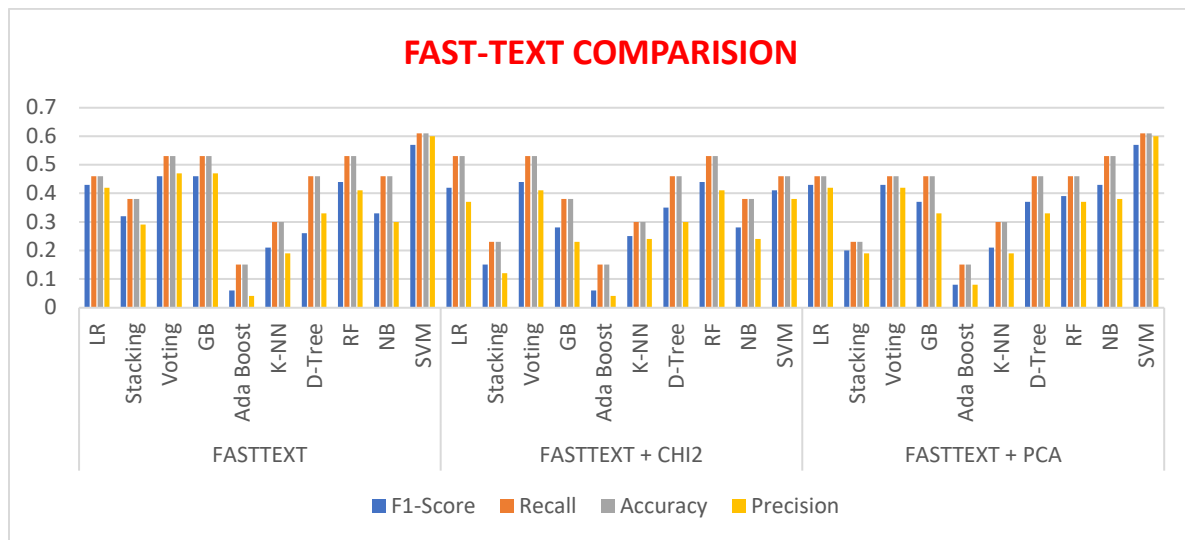


Figure 4.5. Result of FAST-TEXT in Multi-Classification of FR & NFR with Dataset-3

The table-5 demonstrates the categorization of Binary Classification of NFR & FR with dataset-1 'Software_requirement.csv'. It has been investigated that TF-IDF combined with Chi2 simulated with ML classifier such as SVM, Voting, LR, Stacking and NB have achieved an accuracy of 98% but SVM achieved the best F1-score, recall, accuracy and precision of 98%. Bow combined with Chi2 simulated with LR, stacking & NB reached an accuracy of 97%. Word-2-VEC, Glove and Fast-Text combined with SVM and Stacking has also given a significant F1-score of 88% to 89%.

Table-5 Results obtained after simulation with datasets-1 with Binary classification of FR AND NFR.

Feature_Extraction	Feature_Selection	ML Classifier	F1-Score	Recall	Accuracy	Precision
BAG OF WORDS	BOW	LR	0.96	0.98	0.96	0.95
	BOW +CHI2	Stacking	0.97	0.98	0.97	0.96
	BOW+PCA	LR	0.92	0.92	0.92	0.92
TERM FEREQUENCY-INVERSE DOCUMENT FREQUENCY	TF-IDF	SVM	0.98	0.98	0.98	0.98
	TF-IDF +CHI2	SVM	0.98	0.98	0.98	0.98
	TF-IDF+PCA	SVM	0.89	0.89	0.89	0.89
WORD-2-VECTORIZATION	WORD-2-VEC	SVM	0.87	0.88	0.88	0.87
	WORD-2-VEC +CHI2	SVM	0.88	0.88	0.88	0.88
	WORD-2-VEC+PCA	Stacking	0.89	0.89	0.89	0.89
GLOVE	Glove	Stacking	0.89	0.89	0.89	0.89
	GLOVE +CHI2	Stacking	0.89	0.89	0.89	0.89
	GLOVE +PCA	Stacking	0.89	0.89	0.89	0.89
FASTTEXT	FASTTEXT	Stacking	0.89	0.90	0.90	0.89
	FASTTEXT +CHI2	Stacking	0.90	0.90	0.90	0.90
	FASTTEXT +PCA	Stacking	0.89	0.89	0.89	0.89

Graphical analysis has also been used to demonstrate this research. Figure 5.1 demonstrates about the BOW separately and combined with two feature selection methods such as PCA & Chi2 merged with ML classifier to evaluate the performance measures in Binary-class categorization of NFR & FR in software requirements. Figure 5.2 demonstrates about the TF-IDF combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance measures in multi-class categorization of NFR & FR in software requirements. Figure 5.3 demonstrates about the Word-2-Vec combined with two feature selection

methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance measures in multi-class categorization of NFR & FR in software requirements. Figure 5.4 demonstrates about the Glove combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance measures in multi-class categorization of NFR & FR in software requirements. Figure 5.5 demonstrates about the Fast-text combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance measures in multi-class categorization of NFR & FR in software requirements.

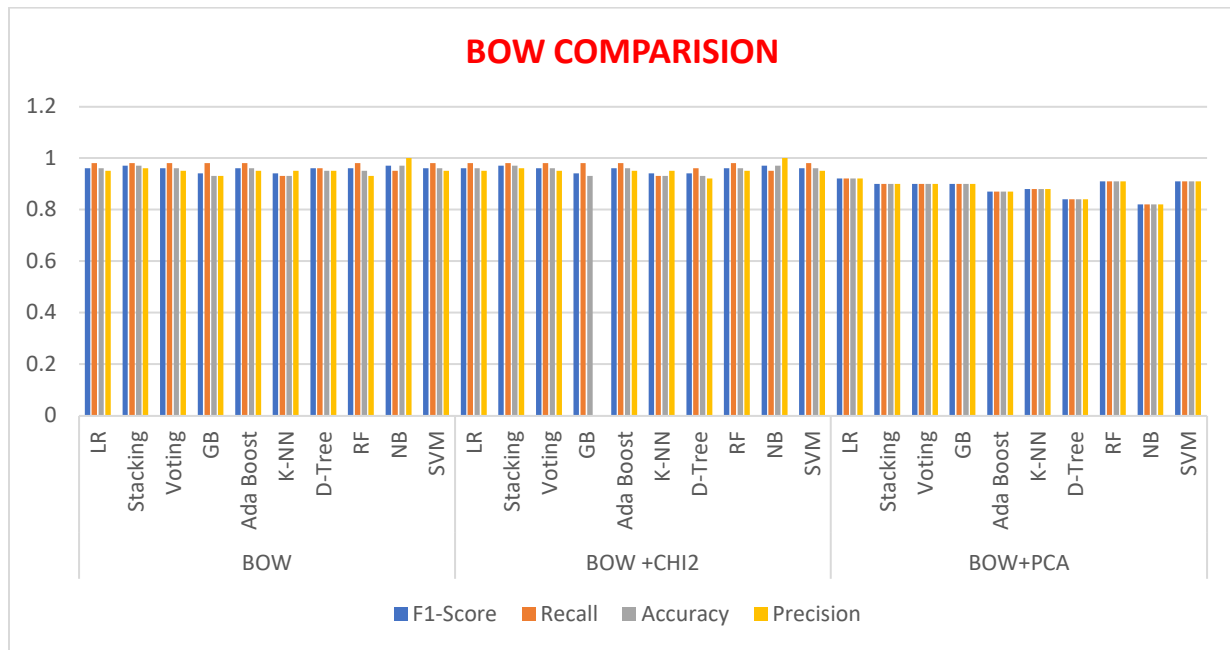


Figure 5.1. Result of BOW in Binary-Classification of FR & NFR with Dataset-1

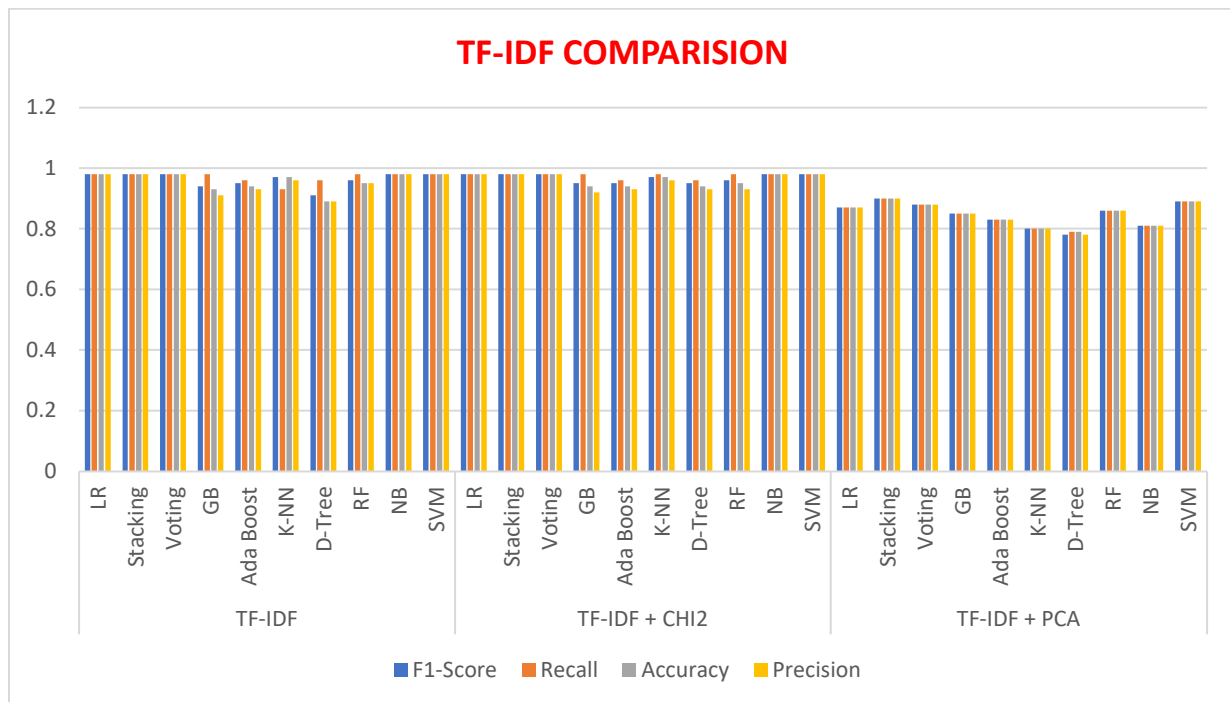


Figure 5.2. Result of TF-IDF in Binary-Classification of FR & NFR with Dataset-1

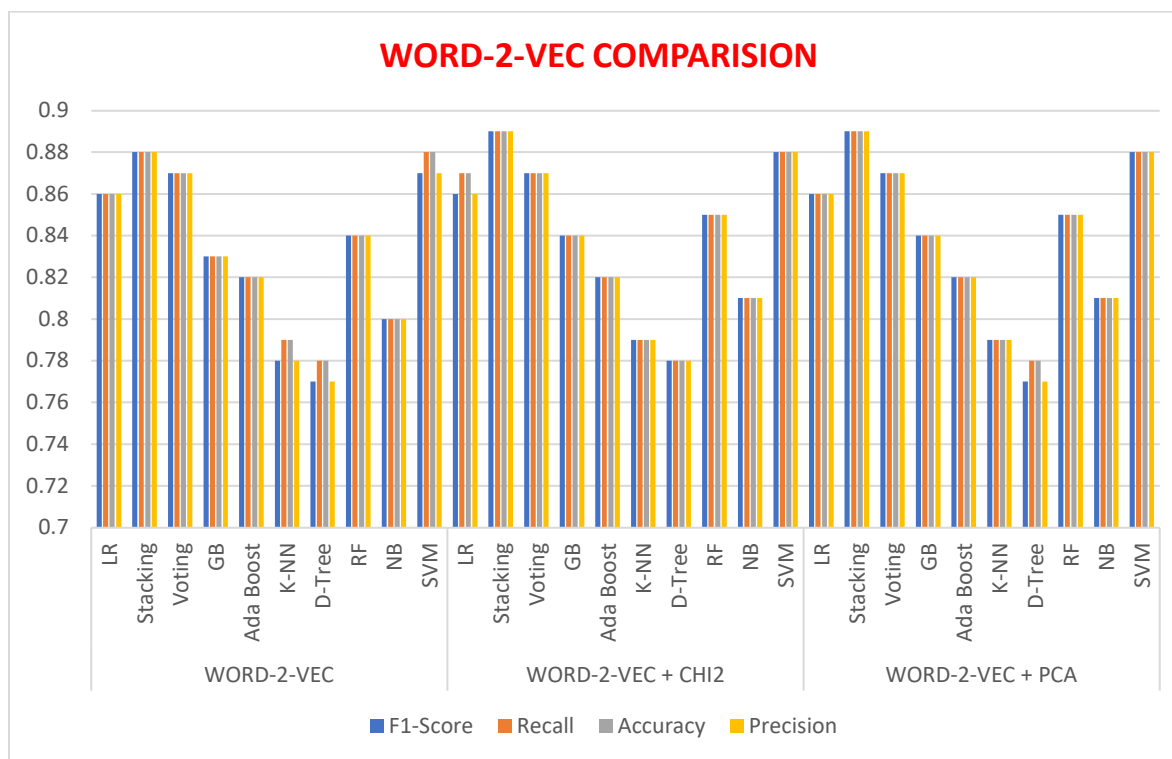


Figure 5.3. Result of WORD-2-VEC in Binary-Classification of FR & NFR with Dataset-1

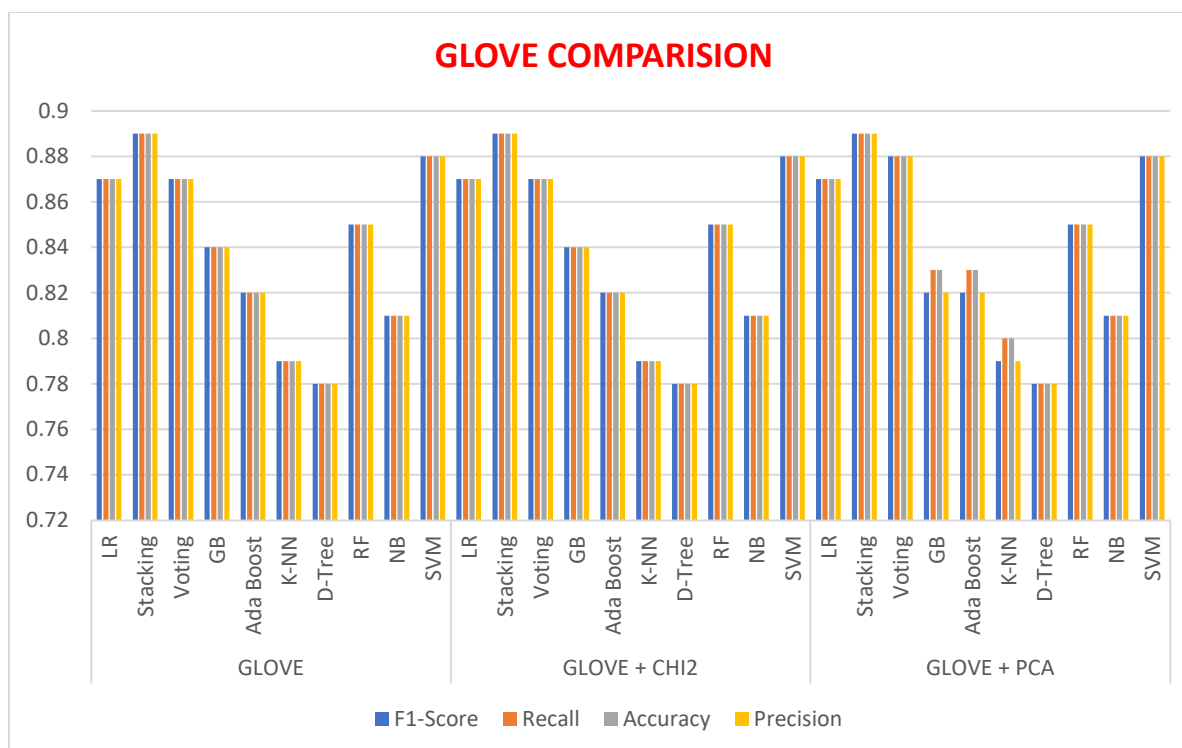


Figure 5.4. Result of GLOVE in Binary-Classification of FR & NFR with Dataset-1

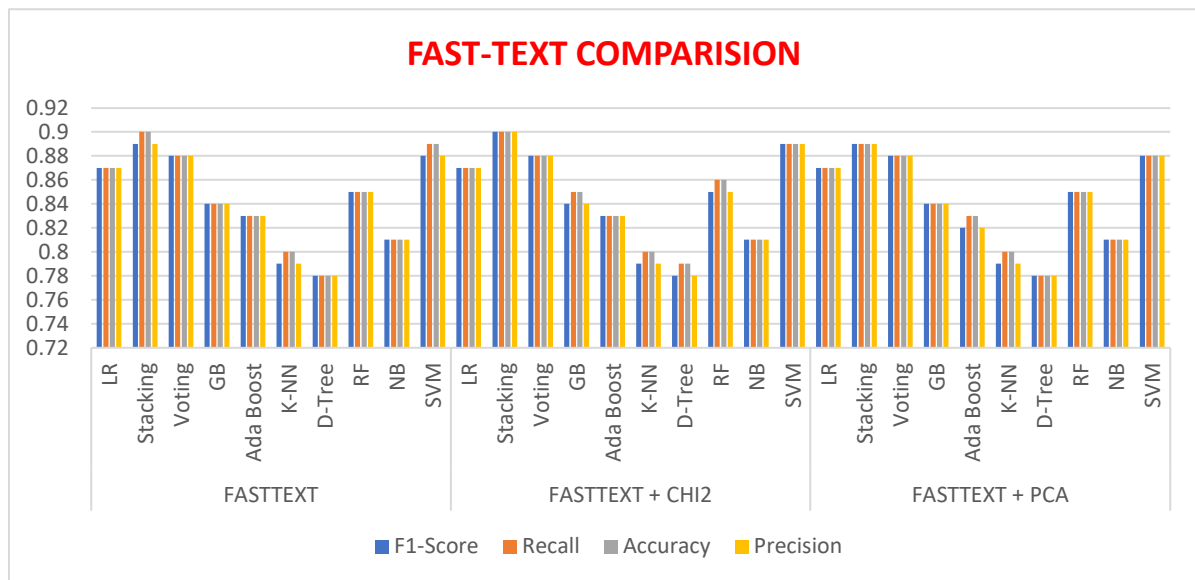


Figure 5.5. Result of FAST-TEXT in Binary-Classification of FR & NFR with Dataset-1

Table-6 demonstrates the categorization of binary classification of NFR & FR with dataset-2 'Risk_prediction.csv'. It has been investigated TF-IDF combined with Chi2 and simulated with LR has achieved the best accuracy of 92%. Fast-Text with PCA provides poor performance with an accuracy of 58% to 63% and D-Tree also gave worst

performance with all ML classifier in the range of 48% due to concept of overfitting. Glove provides an average performance with an accuracy of 86 % with AdaBoost and 83% with RF. Word-2-Vec is also similar to Glove that achieved an accuracy of 81% to 83%.

Table-6 Results obtained after simulation with datasets-2 with Binary-class classification of FR AND NFR.

Feature_Extraction	Feature_Selection	ML Classifier	F1-Score	Recall	Accuracy	Precision
BAG OF WORDS	BOW	Stacking	0.83	0.83	0.83	0.83
	BOW +CHI2	NB	0.92	0.92	0.92	0.92
	BOW+PCA	Stacking	0.84	0.84	0.84	0.84
TERM FEREQUENCY-INVERSE DOCUMENT FREQUENCY	TF-IDF	Voting	0.92	0.92	0.92	0.92
	TF-IDF +CHI2	LR	0.92	0.92	0.92	0.92
	TF-IDF+PCA	NB	0.89	0.89	0.89	0.89
WORD-2-VECTORIZATION	WORD-2-VEC	LR	0.83	0.83	0.83	0.83
	WORD-2-VEC +CHI2	LR	0.83	0.83	0.83	0.83
	WORD-2-VEC+PCA	RF	0.83	0.83	0.83	0.83
GLOVE	Glove	RF	0.83	0.83	0.83	0.83
	GLOVE +CHI2	LR	0.81	0.81	0.81	0.81
	GLOVE +PCA	Stacking	0.86	0.86	0.86	0.86
FASTTEXT	FASTTEXT	RF	0.73	0.73	0.73	0.73
	FASTTEXT +CHI2	Voting	0.76	0.76	0.76	0.76
	FASTTEXT +PCA	D-Tree	0.66	0.66	0.66	0.66

Graphical analysis has also been used to demonstrate this research. Figure 6.1 demonstrates about the BOW combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in binary-class categorization of NFR & FR in software requirements. Figure 6.2 demonstrates about the TF-IDF combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 6.3 demonstrates about the Word-2-Vec separately and combined with two feature selection methods such as PCA & Chi2

simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 6.4 demonstrates about the Glove with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 6.5 demonstrates about the Fast-text combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements.

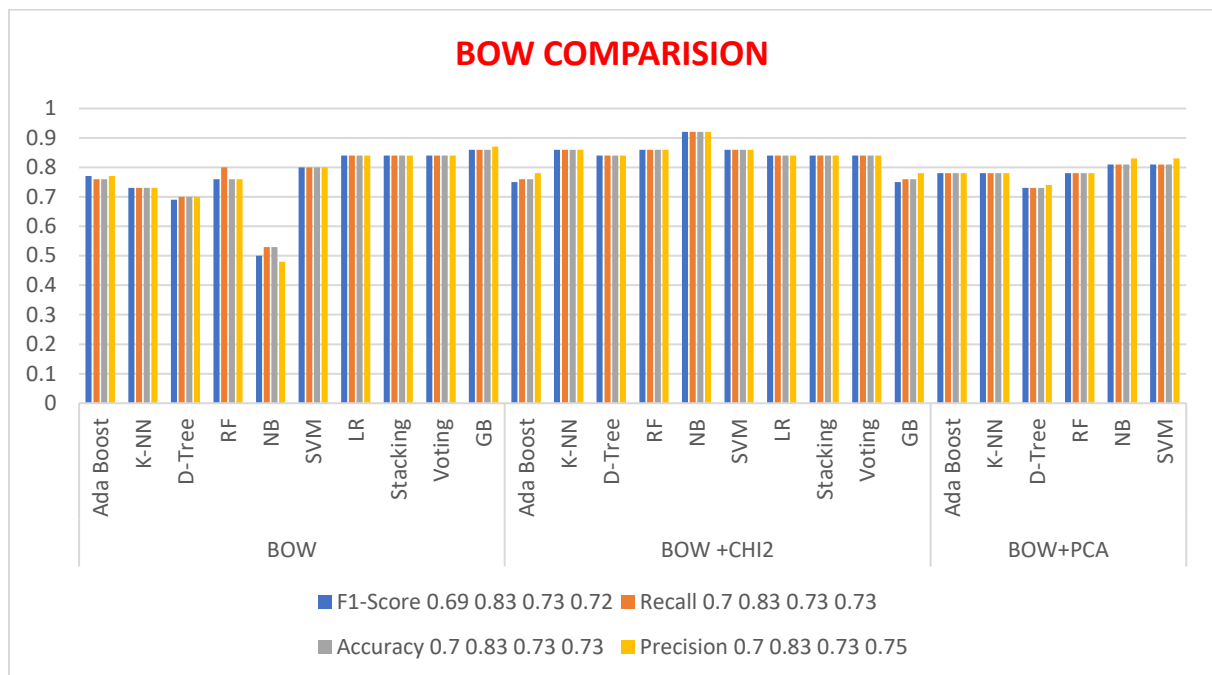


Figure 6.1. Result of BOW in Binary-Classification of FR & NFR with Dataset-2

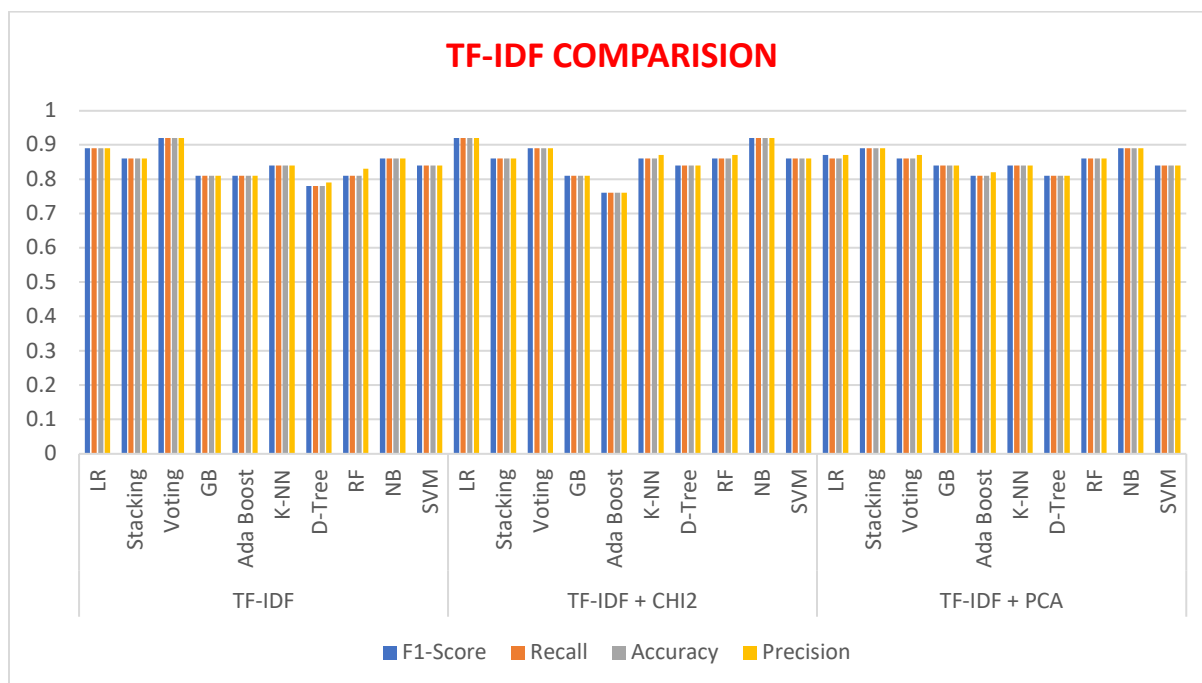


Figure 6.2. Result of TF-IDF in Binary-Classification of FR & NFR with Dataset-2

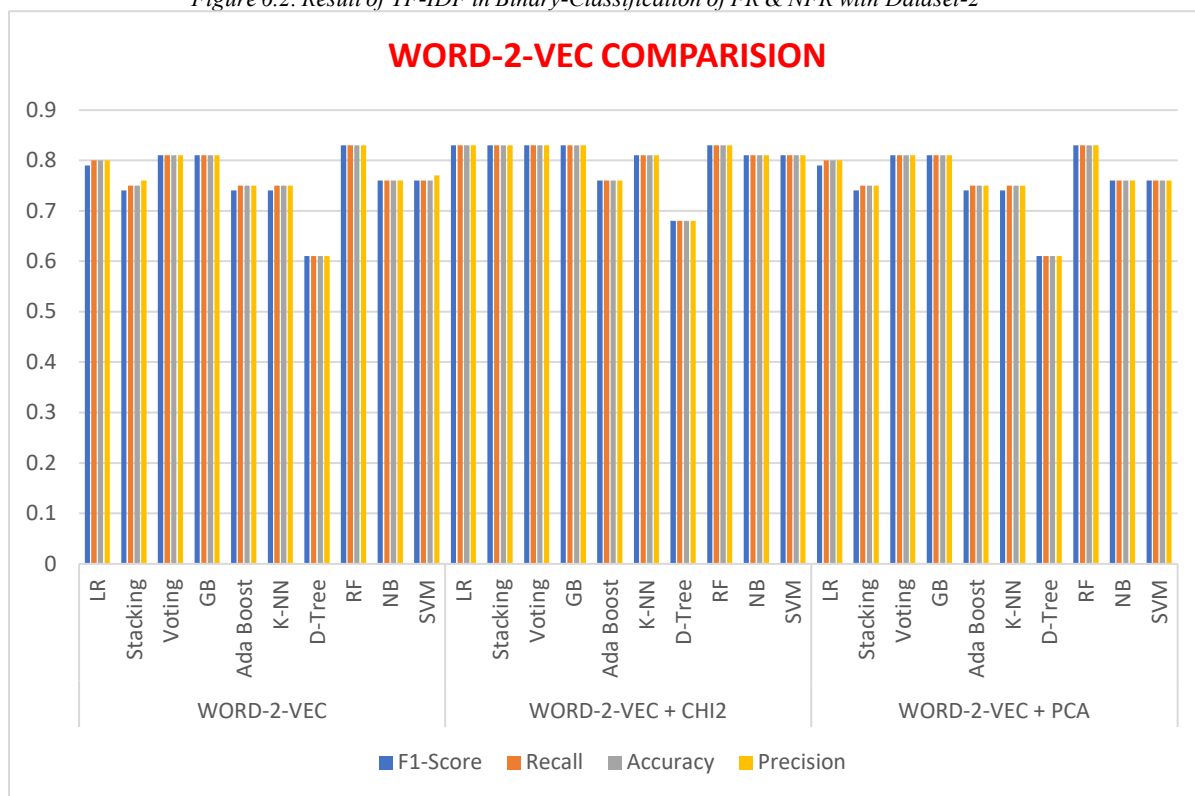


Figure 6.3. Result of WORD--VEC in Binary-Classification of FR & NFR with Dataset-2

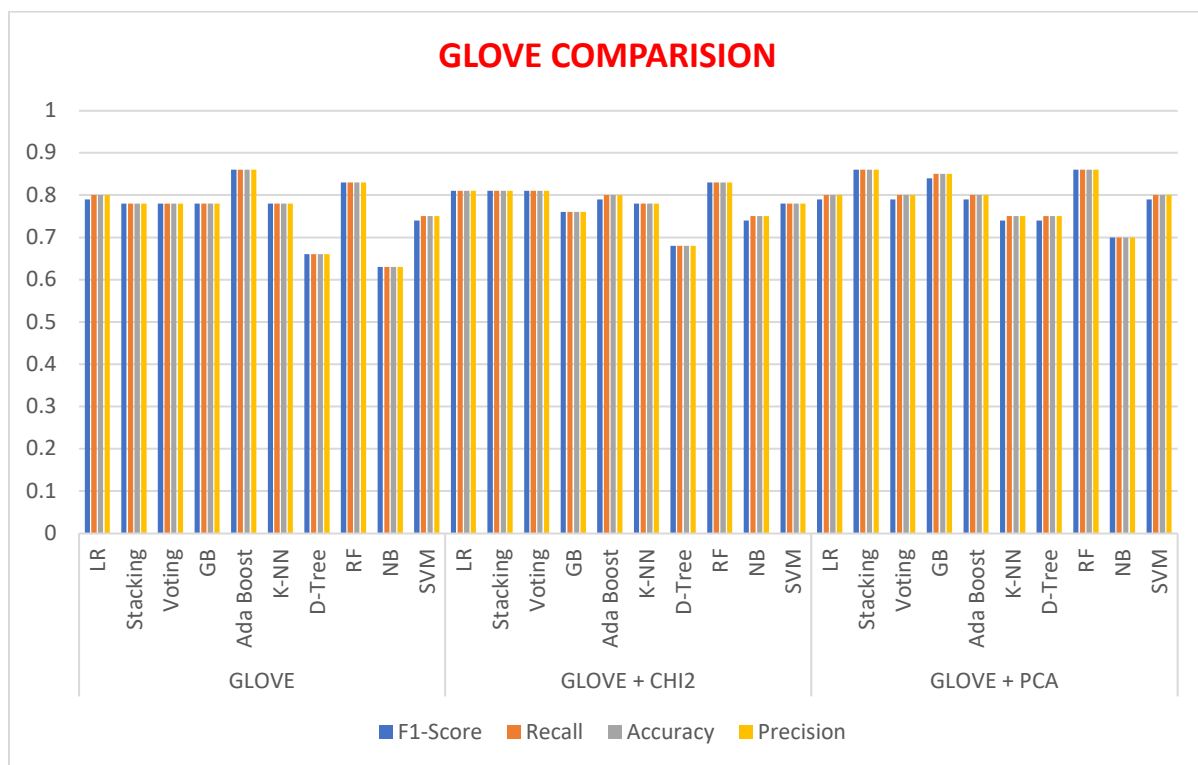


Figure 6.4 Result of GLOVE in Binary-Classification of FR & NFR with Dataset-2

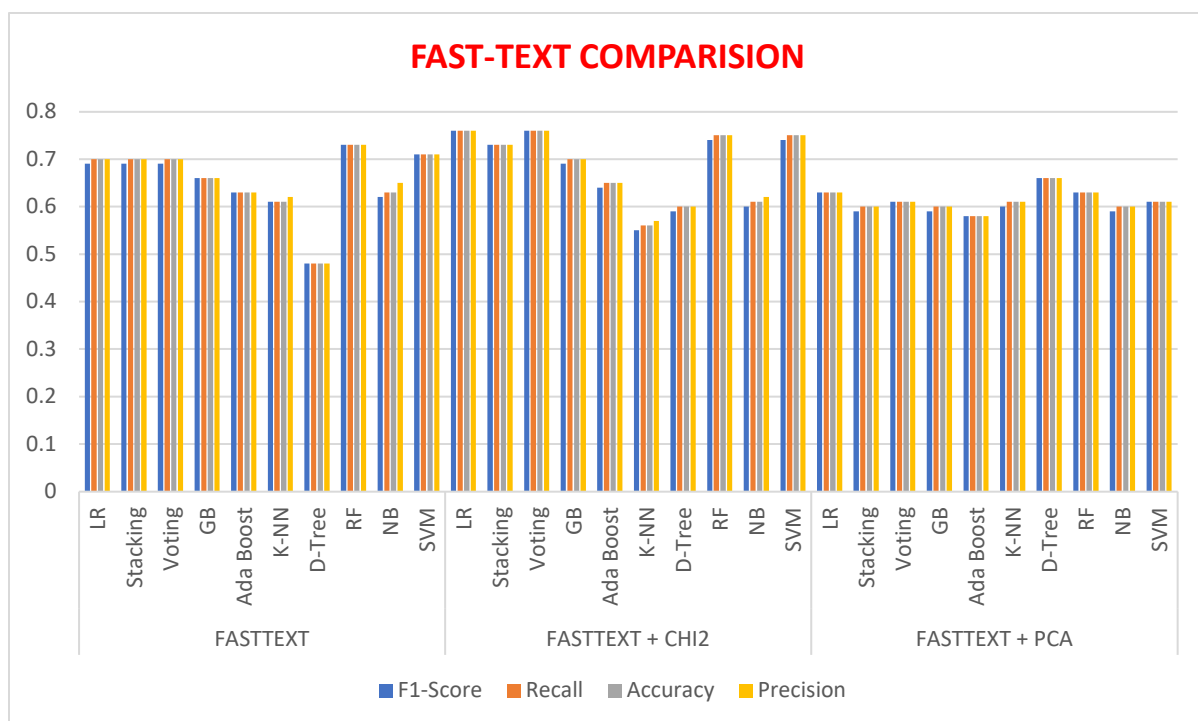


Figure 6.5 Result of FAST-TEXT in Binary-Classification of FR & NFR with Dataset-2

Table-7 demonstrates the categorization of Binary Classification of NFR & FR with dataset-3 'New_Software_Requirements.csv'. It has been investigated that AdaBoost combined with PCA and achieved an accuracy of 92% and highest

precision of 94%. Glove combined with K-NN, D-Tree and RF gave average performance precision of 92%. Fast-Text combined with PCA and SVM also gave average performance in precision of 91%.

Table-7 Results obtained after simulation with datasets-3 with Binary-class classification of FR AND NFR.

Feature_Extraction	Feature_Selection	ML Classifier	F1-Score	Recall	Accuracy	Precision
BAG OF WORDS	BOW	NB	0.92	0.92	0.92	0.92
	BOW +CHI2	NB	0.92	0.92	0.92	0.92
	BOW+PCA	RF	0.91	0.92	0.92	0.92
TERM FEREQUENCY-INVERSE DOCUMENT FREQUENCY	TF-IDF	Voting	0.91	0.92	0.92	0.92
	TF-IDF +CHI2	Ada Boost	0.91	0.92	0.92	0.92
	TF-IDF+PCA	Ada Boost	0.92	0.92	0.92	0.94
WORD-2-VECTORIZATION	WORD-2-VEC	LR	0.92	0.92	0.92	0.92
	WORD-2-VEC +CHI2	Stacking	0.91	0.92	0.92	0.92
	WORD-2-VEC+PCA	Stacking	0.91	0.92	0.92	0.92
GLOVE	Glove	RF	0.91	0.92	0.92	0.92
	GLOVE +CHI2	RF	0.91	0.92	0.92	0.92
	GLOVE +PCA	Stacking	0.91	0.92	0.92	0.92
FASTTEXT	FASTTEXT	RF	0.92	0.92	0.92	0.91
	FASTTEXT +CHI2	Stacking	0.92	0.92	0.92	0.91
	FASTTEXT +PCA	GB	0.92	0.92	0.92	0.91

Graphical analysis has also been used to demonstrate this research. Figure 7.1 demonstrates about the BOW combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in Binary-class categorization of NFR & FR in software requirements. Figure 7.2 demonstrates about the TF-IDF combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 7.3 demonstrates about the Word-2-Vec combined with two feature selection methods such as PCA & Chi2 simulated with ML

classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 7.4 demonstrates about the Glove combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements. Figure 7.5 demonstrates about the Fast-text combined with two feature selection methods such as PCA & Chi2 simulated with ML classifier to evaluate the performance in multi-class categorization of NFR & FR in software requirements.

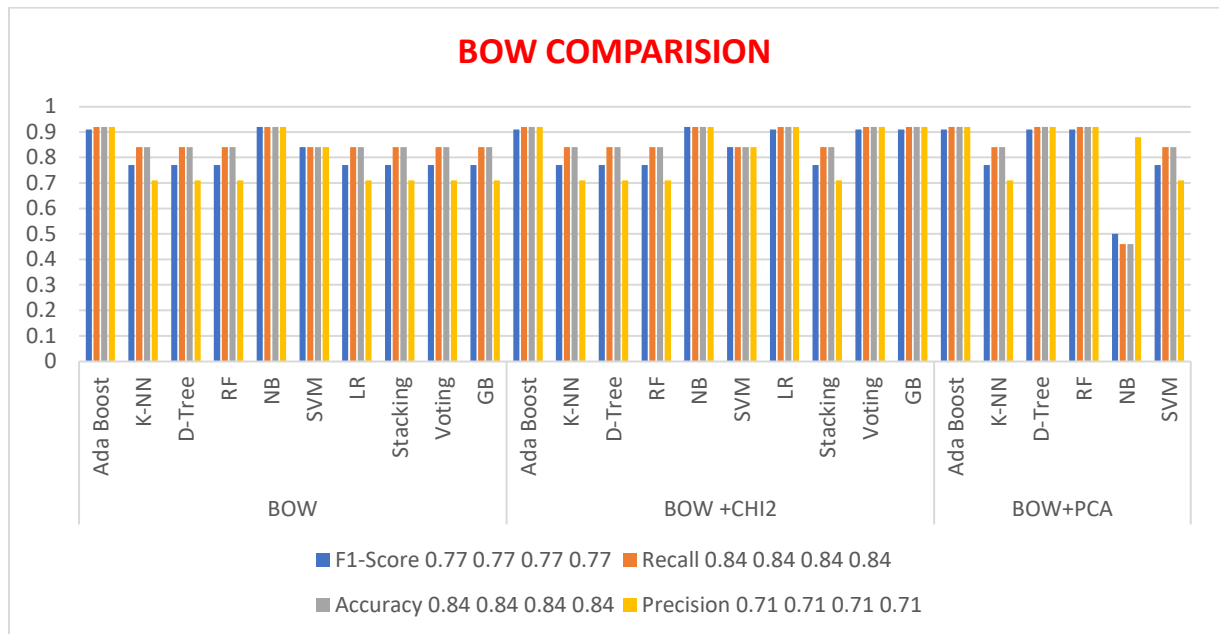


Figure 7.1 Result of BOW in Binary-Classification of FR & NFR with Dataset-3

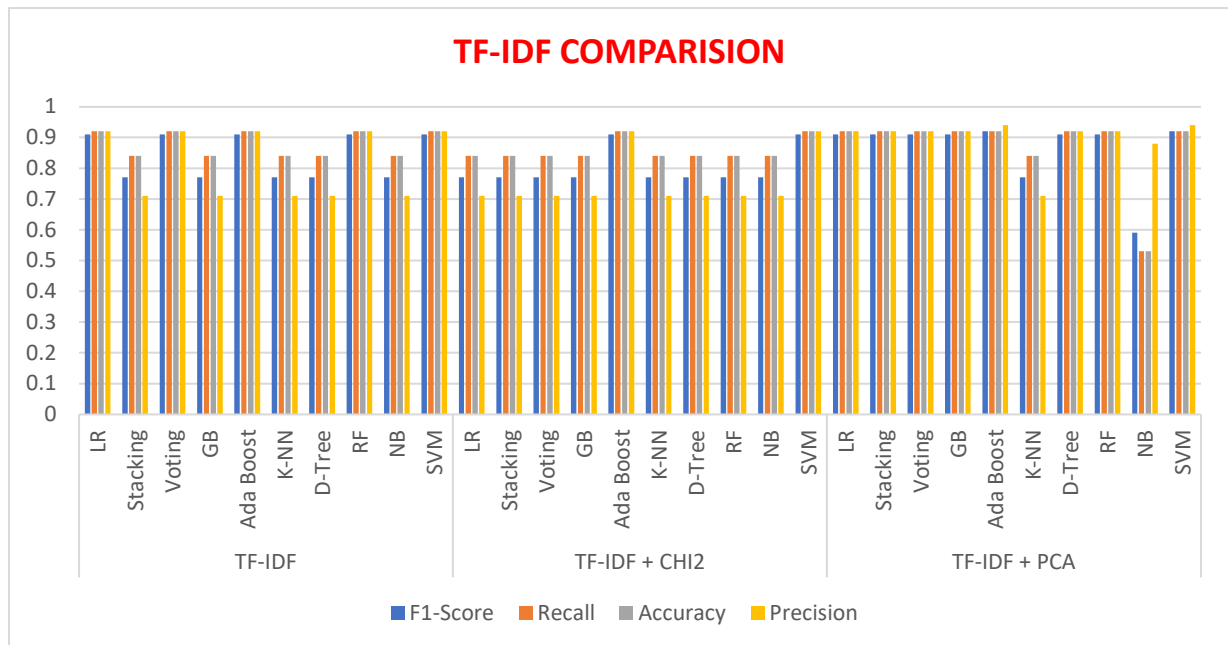


Figure 7.2 Result of TF-IDF in Binary-Classification of FR & NFR with Dataset-3

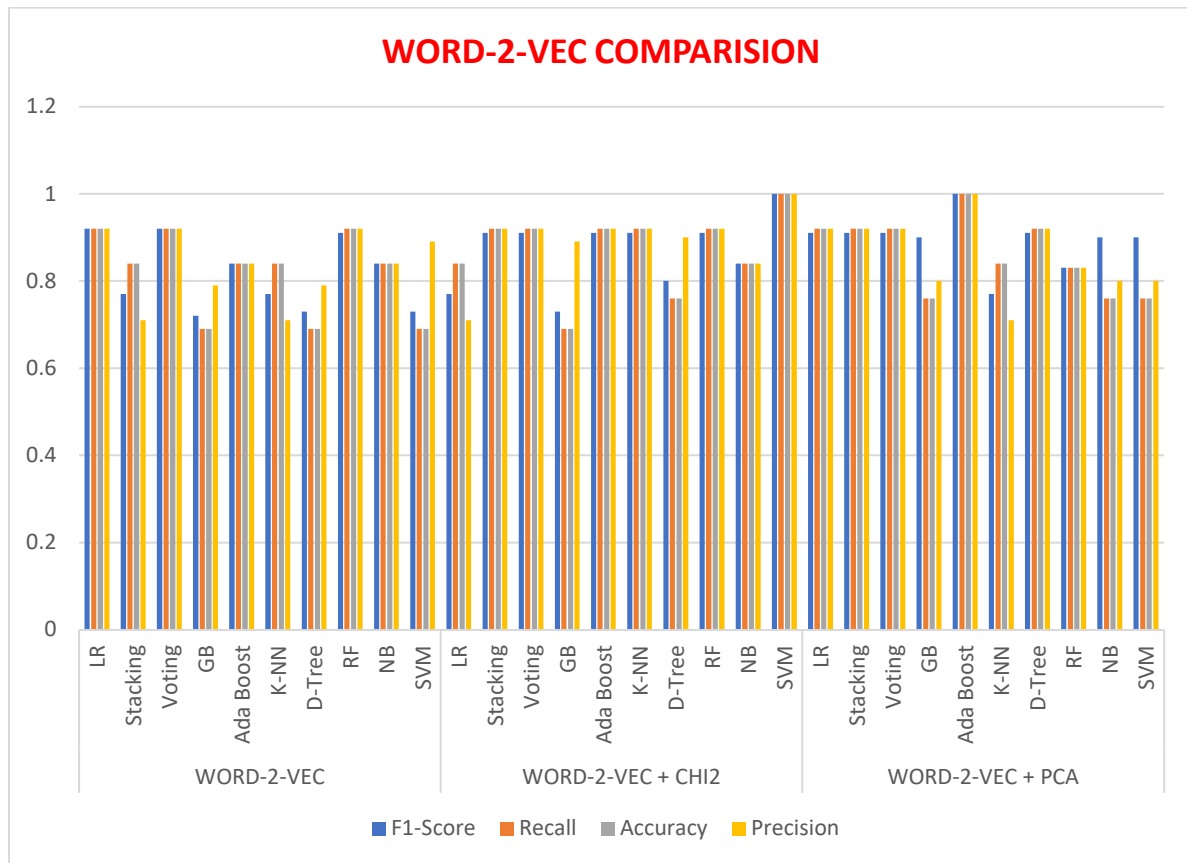


Fig.7.3 Result of WORD-2-VEC in Binary-Classification of FR & NFR with Dataset-3

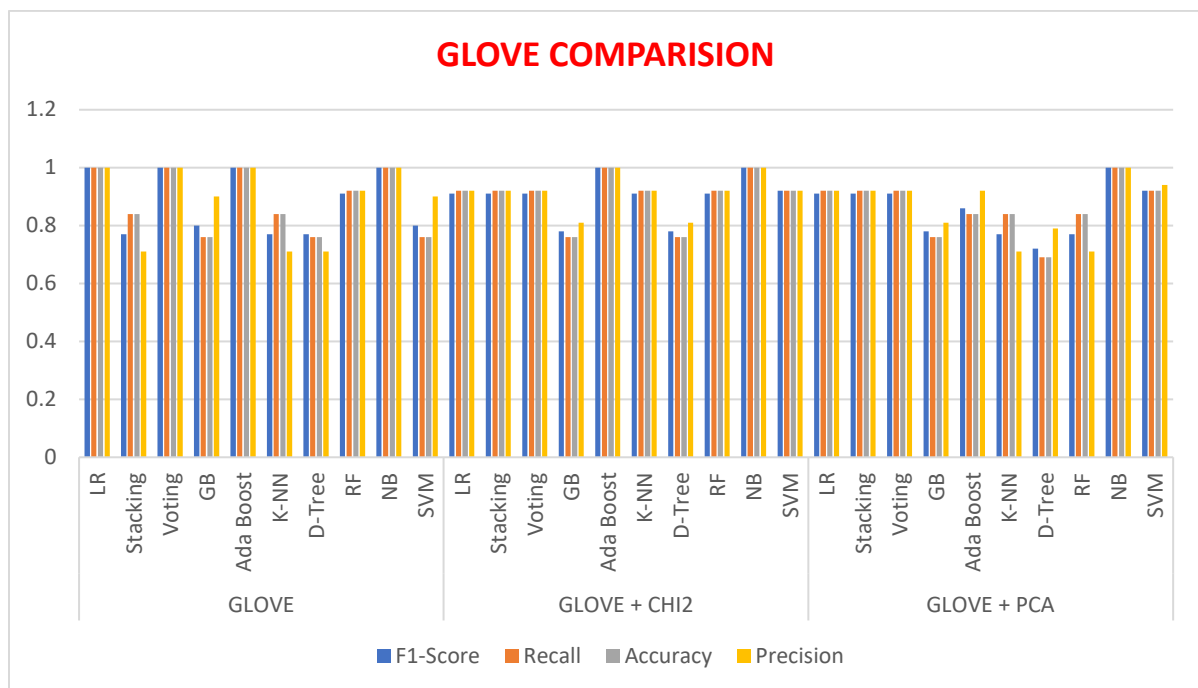


Fig.7.4 Result of GLOVE in Binary-Classification of FR & NFR with Dataset-3

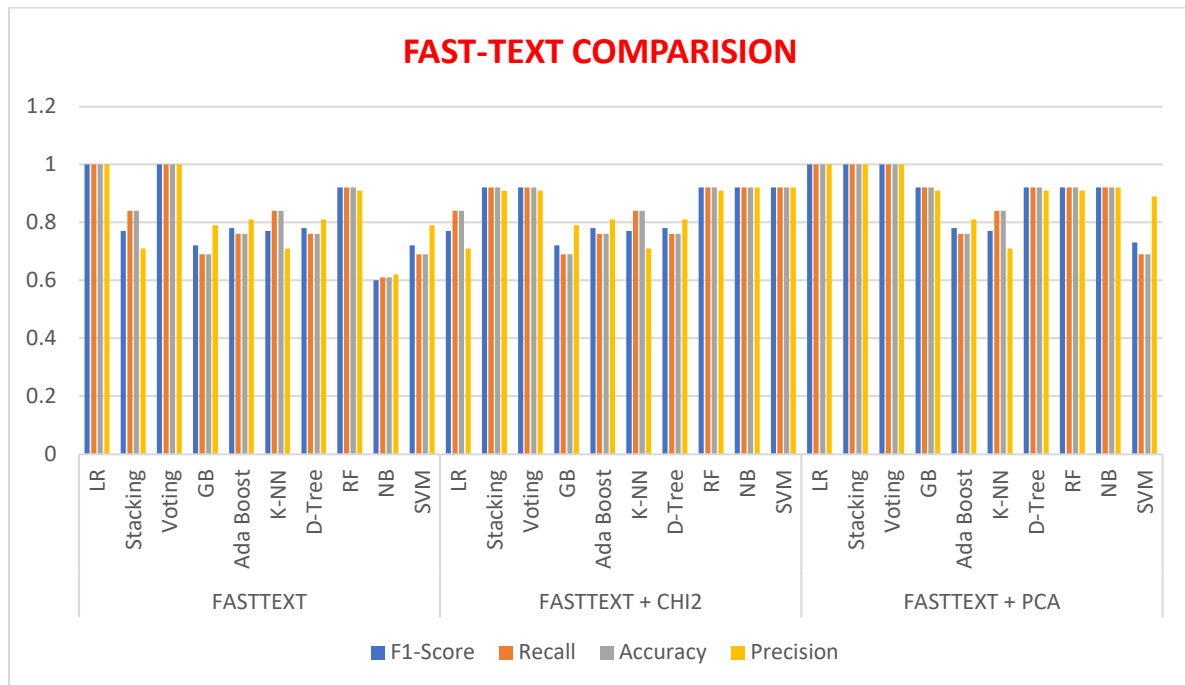


Figure 7.5 Result of FAST-TEXT in Binary-Classification of FR & NFR with Dataset-3

Some works on dataset-1 have been found in the literature. The Table-VIII represents the comparative performance study of binary and multi-class classification of NFR and FR with datasets-1 'software_requirement.csv' with other works. In this, performance is evaluated on applying ML classifier with different feature extraction and feature selection method such as Tf-Idf, BOW, Chi2 and also some research papers have used the concept of deep learning such as BiLSTM that achieves an accuracy of 90%, Zero-Shot learning has achieved an accuracy in the range of 72% to 80%. On dataset-2 and dataset-3 no other works could be found in the literature. This proves novelty of this research. In

this research work we have used three datasets such as

'software_requirement.csv', 'software_risk_prediction.csv' and 'new_software_requirement.csv' that is nobility of my work. In recent research papers all work is done on only dataset-1. It has been found that in multi-class categorization of NFR & FR, dataset-1 achieved an accuracy of 90%, dataset-2 achieved an accuracy of 78%, and dataset-3 achieved an accuracy of 77%. In binary-classification of NFR & FR, it has been investigated that dataset-1 achieved an accuracy of 98%, dataset-2 achieved an accuracy of 92% and dataset-3 achieved an accuracy of 92%.

Table-8 Comparative Analysis of Binary and Multi-Class Categorization of NFR and FR with Dataset-1

Work_Details	Models used	Results of Binary Classification	Results of Multi-Class Classification
Binkhonain and Zhao [1] (2023)	HierarchicalML models	Accuracy :63%, F1-score:51% [10-cross fold] Accuracy: 64%, F1-score: 50% [P-fold]	NIL
Muhammad bin Saleem et al. [2] (2025)	MAML, Prototypical, Networks	MAML :77% Proto-Net: 82%	NIL
Alhoshan et al. [3] (2023)	GPT based Zero-Shot	Accuracy:66%	Accuracy: 72 to 80%
Vincenzo De Martin and Fabio Palomba [4] (2024)	Hybrid-rule based method using RF, SVM, K-NN, NLP, LR, D-Tree, NB	Accuracy :93%	NIL

Canedo and Mendes [5] (2020)	SVM, NB, LR, K-NN	Accuracy: 91%	Accuracy: 78%
Jindal [6] (2021)	RF, Logit-Boost, MLP, Ada-boost, NB, J48, Bagging	NIL	Accuracy: 77%
Rahman and Nayem [7] (2023)	Ridge, LR, SGD, PA, BNB, MNB, SVC, CNB, D-Tree, K-NN, Bagging, RF, Ada-Boost & GB	NIL	Accuracy: 82%
K. Kaur and P.Kaur [8] (2023)	Deep-learning such as CNN-BiLSTM	Accuracy: 95%	NIL
Vrutik Patel et. al [9] (2023)	NB, SVM, D-Tree and K-NN	Accuracy: 90%	NIL
Gaith Y Quba et. al [10] (2021)	K-NN & SVM	Accuracy: 90%	Accuracy: 72%
My work [work-11]	SVM, LR, NB, GB, Ada-Boost, Stacking, RF, K-NN, D-Tree, Voting	Accuracy: 99%	Accuracy: 89%

5. CONCLUSION AND FUTURE WORK

This research demonstrated about the several feature selection, feature extraction and ML models for software requirement categorization in NFR & FR from the SRS Documents. Through investigation the overall results represented that Fast-Text merged with stacking and Chi2 gives the best results across all feature extraction and selection techniques which reached an accuracy of 89% and F1-Score of 88% in the multi-class classification of FR & NFR in software requirements classification by using the 'promise dataset'. In binary classification of FR & NFR across all experiments, TF-IDF is the best feature extraction method that gave an accuracy of 99% and F1-score is 98 %. It is also best option when TF-IDF combined with CHI2 and simulated with SVM & Stacking which also achieved an accuracy of 97%. PCA is found not much useful.

In the future, deep learning concept like GPT based techniques, Distil-Bert, Roberta, S-Bert and transformer classifier could also be used for automation of software requirements. The concept of other feature extraction and selection techniques like mutual-information, genetic models and particular domain datasets could be used for increasing the performance of the model. The research could also move towards hybrid feature methods like POS tags that automate the FR and NFR classification in more powerful manner.

ACKNOWLEDGEMENTS

The first author extends deep gratitude to Dr. Mahendra Kumar Gourisaria and Dr. Junali Jasmine Jena for their constant and unwavering support, without which this study would not have been possible.

REFERENCES

- [1] A. Hemmat, M. Sharbaf, S. Kolahdouz-Rahimi, K. Lano, & Y. S. Tehrani. Research directions for using LLM in software requirement engineering: A systematic N. Rahimi, F. Eassa, & L. Elrefaei. An ensemble machine learning technique for functional requirement classification. *symmetry*, 12(10), 1601, (2020).
- [2] S. Tiun, S. U. Mokhtar, H. S. Bakar, & S. Saad. Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text. In *journal of Physics: conference series* (Vol. 1529, No. 4, p. 042077). IOP Publishing, (2020).
- [3] M. Binkhonain, & L. Zhao. A machine learning approach for hierarchical classification of software requirements. *Machine Learning with Applications*, 12, 100457, (2023).
- [4] B. M. Saleem, N. Qamar, R. Danial, K. Sardar, M. Noor & U. Omer. Analyzing the Impact of Machine Learning Algorithms on Software Requirements Classification. *VFAST Transactions on Software Engineering*, 13(1), 88-98, (2025).
- [5] W. Alhoshan, A. Ferrari, & L. Zhao. Zero-shot learning for requirements classification: An exploratory study. *Information and Software Technology*, 159, 107202, 2023.

- [6] V. De Martino, & F. Palomba. Classification and challenges of non-functional requirements in ML-enabled systems: A systematic literature review. *Information and Software Technology*, 107678, 2025.
- [7] R. Izhar, S. N. Bhatti., & S.A. Alharthi (2025). Bridging Precision and Complexity: A Novel Machine Learning Approach for Ambiguity Detection in Software Requirements. *IEEE Access*.
- [8] E. Dias Canedo, & B. Cordeiro Mendes. Software requirements classification using machine learning algorithms. *Entropy*, 22(9), 1057, (2020) review. *Frontiers in Computer Science*, 7, 1519437, 2025.
- [9] E. Dias Canedo, & B. Cordeiro Mendes. Software requirements classification using machine learning algorithms. *Entropy*, 22(9), 1057, (2020)
- [10] R. Jindal, R. Malhotra, Jain, & A. Bansal. Mining non-functional requirements using machine learning techniques. *e-Informatica Software Engineering Journal*, 15(1), (pp. 85-114), 2021.
- [11] A. Rahman, A. Nayem, & S. Siddik. Non-Functional Requirements Classification Using Machine Learning Algorithms. *International Journal of Intelligent Systems and Applications*, 15(3), pp. 56-69, 2023.
- [12] K. Kaur, & P. Kaur. MNoR-BERT: multi-label classification of non-functional requirements using BERT. *Neural Computing and Applications*, 35(30), 22487-22509, 2023.
- [13] V. Patel., P. Mehta., & K. Lavingia (2023, April). Software Requirement Classification Using Machine Learning Algorithms. In *2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1)* (pp. 1-6). IEEE.
- [14] G.Y.Quba, H. Al Qaisi, A. Althunibat, & S.AlZu'bi (2021, July). Software requirements classification using machine learning algorithm's. In *2021 international conference on information technology (ICIT)* (pp. 685-690). IEEE.
- [15] V. Singh, "Software Requirements Dataset," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/iamvaibhav100/software-requirements-dataset>. [Accessed: Jun. 17, 2025], Jun. 2023.
- [16] R. Naseem, Z. Shaukat, M. Irfan, M. A. Shah, A. Ahmad, F. Muhammad, A. Glowacz, L. Dunai, J. Antonino-Daviu, and A. Sulaiman, "A Dataset for Software Requirements Risk Prediction," in *Proc. (conference name)*, 2019.
- [17] H. N. Ahmed, "software-requirements-diverse," Kaggle Dataset. Available:online, 2024.
- [18] K. Juluru, H. H. Shih, N. K. Keshava Murthy, & P. Elnajjar. Bag-of-words technique in natural language processing: a primer for radiologists. *RadioGraphics*, 41(5), 1420-1426,(2021).
- [19] M. Das, & A. J. P.Alphonse. A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset. *arXiv preprint arXiv:2308.04037*, (2023).
- [20] A. Chabukswar, P. Deepa Shenoy, & R. K. Venugopal. Identification of misinformation using word embedding technique word2Vec, machine learning, and deep learning models. In *International Conference on Data Management, Analytics & Innovation* (pp. 45-56). Singapore: Springer Nature Singapore, (2024, January).
- [21] J. Pennington, R. Socher, & D. C. Manning, Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543), (2014, October).
- [22] S. Tiun, A. U. Mokhtar, H. S. Bakar, & S. Saad. Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text. In *journal of Physics: conference series* (Vol. 1529, No. 4, p. 042077). IOP Publishing, (2020, April).
- [23] C. Kumari, M. K. Gourisaria, & J. J. Jena (2025, August). Machine Learning-Based Functional and Non-Functional Classification in Software Requirement. In *2025 9th International Conference on Inventive Systems and Control (ICISC)* (pp. 1365-1370). IEEE, (2025, August).
- [24] A. Maćkiewicz, & W. Ratajczak. Principal components analysis (PCA). *Computers & Geosciences*, 19(3), 303-342, (1993)
- [25] A. D. Pisner, & M. D. Schnyer, (2020). Support vector machine. In *Machine learning* (pp. 101-121). Academic Press.
- [26] A. Das, (2021). Logistic regression. In *Encyclopedia of quality of life and well-being research* (pp. 1-2). Cham: Springer International Publishing.
- [27] N. A. S. Alexandropoulos, K. C. Aridas, B. S. Kotsiantis, & N. M. Vrahatis, (2019). Stacking strong ensembles of classifiers. In *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI*

- 2019, Hersonissos, Crete, Greece, May 24–26, 2019, *Proceedings 15* (pp. 545-556). Springer International Publishing.
- [28] Y. Y. Song, & Y. Lu. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, (2015).
- [29] P. K. Syriopoulos, G. N. Kalampalikis, B. S. Kotsiantis, & N. M. Vrahatis, (2023). K NN classification: a review. *Annals of Mathematics and Artificial Intelligence*, 1-33.
- [30] A. P. I. Pratama, J. S. E. Atmadji, A. D. Purnamasari, & E. Faizal, (2024). Evaluating the performance of voting classifier in multiclass classification of dry bean varieties. *Indonesian Journal of Data and Science*, 5(1), 23-29.
- [31] D. Berrara. Bayes' theorem and naive Bayes classifier, (2025).
- [32] Li J. Y. Li. J. Bai, X. Yang, Y. Jiang, & T. S. Xia, (2022). Multinomial random forest. *Pattern Recognition*, 122, 108331.
- [33] A. Z. Ali, H. Z. Abduljabbar, A. H. Tahir, B. A. Sallow, & M. S. Almufti, (2023). eXtreme gradient boosting algorithm with machine learning: A review. *Academic Journal of Nawroz University*, 12(2), 320-334.
- [34] C. Ying, M. Qi-Guang, L. Jia-Chen., & G. Lin. Advance and prospects of AdaBoost algorithm. *Acta Automatica Sinica*, 39(6), 745-758, (2013)