INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Multi-text classification of Urdu/Roman using machine learning and natural language preprocessing techniques

M Ameen Chhajro[1]*, Mansoor Ahmed Khuhro[1], Kamlesh Kumar[1], Asif Ali Wagan[1], Aamir Iqbal Umrani[1], Asif Ali Laghari[1]

1 Department of Computer Science, Sindh Madressatul Islam University, Karachi, Pakistan

OPEN ACCESS

Received: 11-04-2020

Accepted: 17-05-2020

Published: 18-06-2020

Editor: Dr. Natarajan Gajendran

Citation: Chhajro MA, Khuhro MA, Kumar K, Wagan AA, Umrani AI, Laghari AA (2020) **Multi-text classification of Urdu/Roman using machine learning and natural language preprocessing techniques**. Indian Journal of Science and Technology 13(19): 1890-1900. https://doi.org/10.17485/IJST/v13i19.230

***Corresponding author**.
M Ameen Chhajro

Department of Computer Science, Sindh Madressatul Islam University, Karachi, Pakistan
ameen.chhajro@smiu.edu.pk

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

Abstract

Objectives: This research presents multi-text classification from the news text dataset. The main purpose of this work is to classify multi-text for Urdu and Roman language using Natural Language processing and Machine Learning classification models. **Methods/Statistical analysis:** In this research, online news data has been collected through beautiful soup web scraping tool. In order to analyze the model accuracy news data is divided into six categories which has been composed from various online newspaper platforms. The main news corpus data consists of 10500 news in Urdu and Roman Urdu language including, Accidental, Education, Entertainment, International, Sports and Weather news have been primarily focused in the proposed research study. Furthermore, preprocessing is performed on text corpus using Natural Language Processing technique; for example, data cleaning, data balancing, and stop word removal. For feature extraction count vector, TF-IDF and Chi2 are employed as word filtering. For multi-text classification the Machine Learning classification schemes have been implemented namely, Naive Bayes Classifier, Logistic Regression, Random Forest Classifier, Linear SVC, and K-Neighbors Classifier. After comparative analysis results showed that Linear Support Vector Classifier provided 96% accuracy among other tested methods. **Findings**: Multi-Text classification of Urdu Roman language having different writing styles, word structure, irregularities, grammar, and combined corpus is a challenging task. For this purpose, we implemented different Machine Learning algorithms with Natural Language preprocessing technique which provided optimal results in classification of multi-text news data.

Keywords: MultiText Classification; Machine Learning; NLP Preprocessing Techniques

1 Introduction

The Mutli-Text classification of Urdu language through combination of Urdu and Roman Urdu text is being considered one of the most challenging task in text classification. As Urdu is one of the famous and national languages of Pakistan which is spoken

and written in major part of the country regions. Other countries whose literature also relates with Urdu are India, Bangladesh, Afghanistan, and to a further extend few other countries in globe. The growing use of internet has made easier to access information and share in different communication platforms. Similarly, in business communication understanding different type of text data is important in order to rate, suggest, and review the consumer products. For example various news, articles, stories, blogs and reviews text content typically organized by topics and different products tagged by categories and users can be classified on the basis on how they talk about particular brand or product on online web based platforms However, the majority of text classification blogs and tutorials on the internet can be found in the form of binary text classification whose common example include email classification such as email spam filtering (spam vs. ham), sentiment analysis (positive vs. negative) respectively. The Research has also identified the problem with two Roman Urdu words have same spelling but lexically they are different from each other such as common and mango spelled aam in Roman Urdu, but for quality training of model words need to maintain consistency in its use throughout the process [1,2]. For the analysis of textual data category, the most common and useful approach which plays an important role in the field of NLP like opinion mining, sentiment analysis, tweets, reviews, spam detection, email filtering is the common example of text categorization [3].

Increasing demand of information and rapid growth of online social platforms has given more importance to text classification for the purpose of text data management and arrangement [4].

The frequent news text information circulating on different websites consist variety of text data categories such as sports, entertainment, education, politics and more. Due to the huge amount of news data there is need to categorize the news classes quickly and automatically instead of some manual operations [5]. Nowadays social media and internet is even contributing more like an explosion of text data with variety of types and categories like Urdu, English, Roman Urdu, Arabic, and Chinese etc. Similarly, that has become more challenging for researchers, business holders, other private and Government agencies to deal with such data for decision-making, authentication and information extraction for that textual data. For the purpose to make automation system for this textual data three major topics are mainly focused like feature engineering, selection of features and various machine learning algorithms. Bag-of-Word technique is considered as more widely used approach in feature engineering [6].

1.1 Literature review

Lately field of text mining is gaining more importance due to availability of different data types by multiple sources in the form of unstructured and semi structured information. The prime purpose of text mining is to enable user to extract information from different sources and then perform various operations like information retrieval, classification techniques which range from (supervised, unsupervised and semi supervised), Data mining, Natural Language processing and in combination with machine learning approaches for automatic classification [7]. It also explores patterns from different data types from the documents.

Previously a lot of work has been performed on different languages but less focus is given to Urdu language by research community. Acquiring data from various sources like online blogs then for the purpose of collected text data classification different well-known classification algorithms of machine learning such as Decision Trees, Support Vector Machine, K-Nearest Neighbor etc are used. Tests through comparison analysis brought a final conclusion that K-NN is performing well than Decision tree and Support Vector Machine with the perspective of accuracy-value, precision and recall. Same Research paper has cited another research work which identified that other five main well-known classification algorithms were applied on Urdu Language data, that corpus contained 21769 documents related to news of seven different categories Culture, Business, Health, Entertainment, Sports). After applying various NLP preprocessing on it 93400 features are extracted from the corpus to apply machine learning algorithms up to 94% precision and recall using majority voting [8].

In [9] text classification was performed through stop words removal, feature selection, Bag-of-word, and TF-IDF. The dataset was used to test results using different machine learning methods. However, experiments revealed that most researchers used TF-IDF as method for feature selection. Moreover, Logistic regression and Naïve Bayes Classifier were potential text classification models. In other paper Febby Apri Wenando in [10] suggested machine learning algorithms including support vector machine, multinomial naive Bayes, and decision tree in order to classify Indonesian news articles. Where it showed that support vector machine algorithm provides higher accuracy with 93% f1 score. Similarly, Gurvinder Singh et al. performed sentiment analysis for positive and negative news. Here, Multivariate Bernoulli Naïve Bayes Classification and Multinomial Naïve Bayes Classification schemes were implemented for text categorization [11].

2 Proposed Methodology

In this section we have described the flow of our proposed work. Firstly data corpuses of Urdu and Roman Urdu Text have been collected from different sources. After that NLP preprocessing techniques are applied on text data for feature extraction

and selection. Finally, multiple ML classification algorithms have been implemented. The Figure 1 depicts the flow of working mechanism.

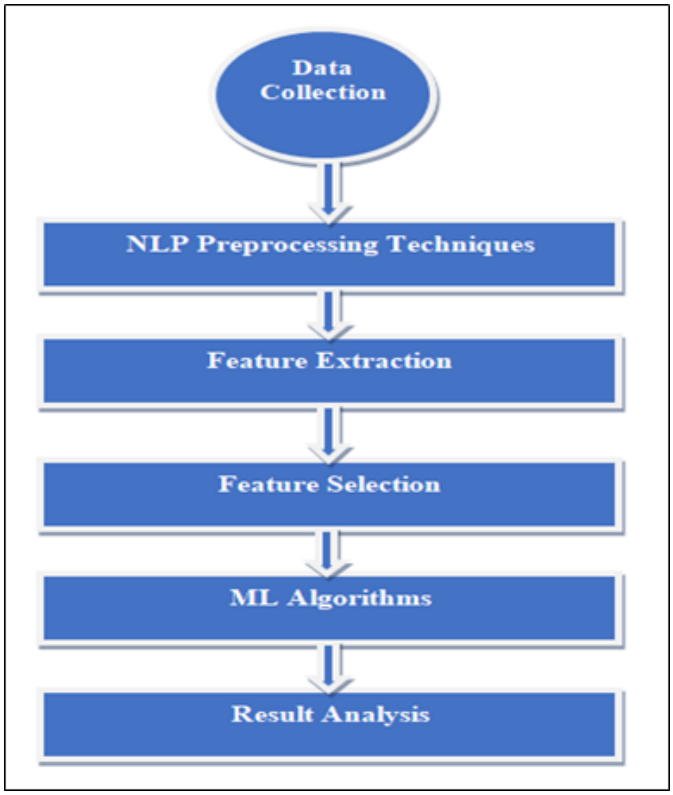

Fig 1. Process diagram

2.1 Data preparation

In first step we collected dataset of Urdu and Roman Urdu text data from various online platforms with the help of beautiful soup web scraping tool. The compiled data was in the form of raw data or an unstructured data so to process that text data through machine learning algorithm we need to have its structured format for any decision-making or its classification. The corpus data contains 10500 different text news of both Urdu and Roman and text data has been prepared for next steps in order to perform the text classification. The target categories of classification from the corpus are Accidental, Education, Entertainment, International, Sports and Weather news respectively.

2.1.1 ***Data imbalanced***

The data imbalanced distribution problem is frequently observed in the field of data science. The problem with the imbalanced data is that the data of one class significantly has higher frequency than the other class resulting in less data points. This problem directly affects the performance of majority of ML/DL classification algorithms because they are not efficient to handle data imbalance issue. As a result they are being inclined towards the classes with majority data points. Figure 2 depicts the imbalanced data distribution.

The literature review reveals that many normal classification algorithms like Logistic Regression and Decision Trees etc are not much efficient to handle imbalanced distribution of classes. This leads to a heavy bias towards the classes with larger data points, while classes with less data points are being considered as noise, and they are mostly ignored. Hence, the outcomes of classes with minority data points have a higher misclassification rate as compared to the majority classes. Consequently, when it comes to the performance evaluation the accuracy metric is not that much relevant if the model is trained on imbalanced data. However, there are some methods to handle the data imbalance issue and these are described below.

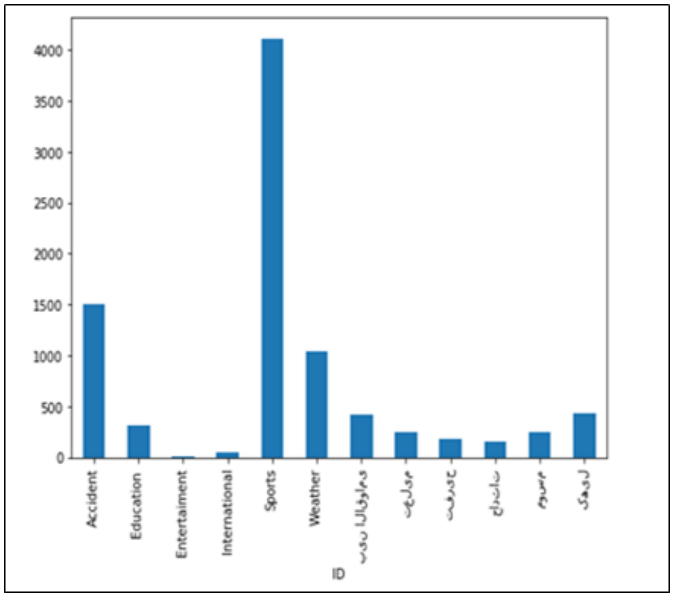

Fig 2. Imbalanced text data

2.1.2 ***Handling imbalanced data***

The collection of actual data while dealing with corpus is always better approach rather than generation of artificial data via sampling the existing data points.

2.1.3 ***Removing duplicate data***

One way to handle the higher frequency is to remove the duplication of data from the dataset meaning that there might be similar data points as repeating multiple times in your dataset. For example "Where is the order" and "Where is my order" has the same semantic meaning. Removing such repeated text content will help to reduce the duplicate message which will help to reduce the volume of majority class.

2.1.4 ***SMOTE (Synthetic Minority Over-sampling Technique)***

In order to balance the text data classes in our dataset, we have implemented SMOTE (Synthetic Minority Over-sampling Technique) to fix the problem of imbalanced data. Where data is being balanced through oversampling minority class data points and undersampling the majority class instances.

2.1.5 ***Undersampling***

It is the process to delete the data points from the majority class on random basis until both classes are balanced with equal data points. There is more probability to lose the information which ultimately cause to poor model training.

2.1.6 ***Oversampling***

It is the process to duplicate the data points of minority class randomly. The problem with this approach is that it might lead to over fitting problem with inaccurate predictions results on the test data.

More importantly, SMOTE approach effectively forces the decision region of the minority class to become more general. It Produces synthetic data points by taking each minority class sample and it introduces the synthetic examples along the line segments joining any/all of the k minority class nearest neighbors as it can seen in the Figure 3.

After performing experiments on our dataset using SMOTE technique, we obtained the result in the form of balanced data as shown in the Figure 4. It has been reported from various research work that this technique is not reliable choice for text data because the numerical form of vectors which are produced from the text data are often in high dimension. While in most cases SMOTE seems beneficial with low-dimensional data, it does not attenuate the bias towards the classification in the majority class for most classifiers when data are high-dimensional.

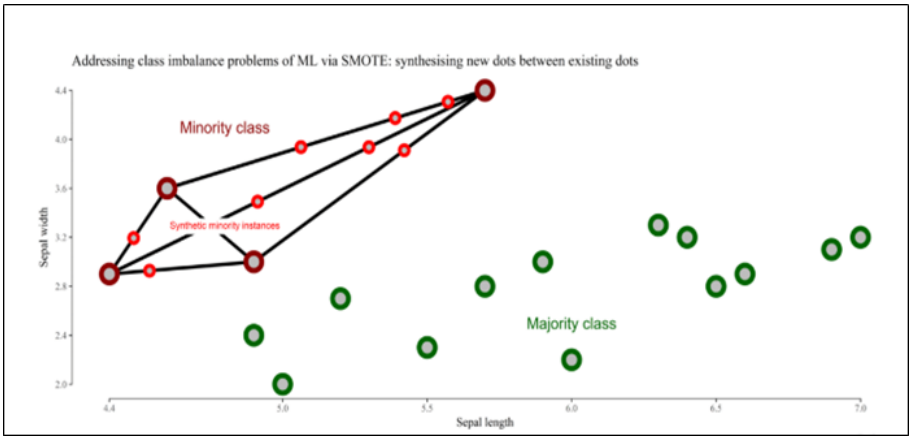

Fig 3. Addressing SMOTE technique

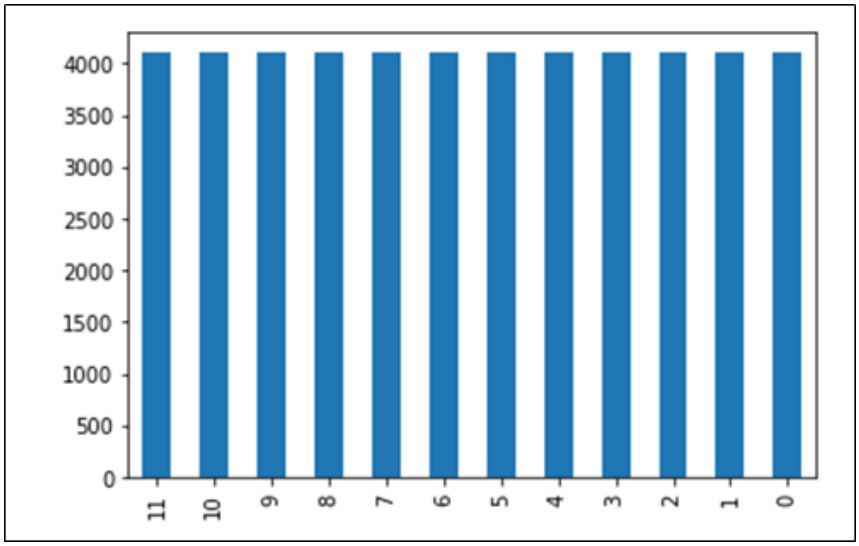

Fig 4. Data Balancing SMOTETechnique

2.1.7 ***Stop word removal***

The text data words which are not as much useful for information extraction and decision making like conjunctions, pronouns, articles and prepositions are considered Non-semantic words are usually described as stop words. In NLP preprocessing steps these words are removed from text data features because they have very minimum or no contribution in information sharing about the text sentence sentiments. Stop words in Urdu like □□□,□□□ □□,□□, etc and In Roman Urdu, stop words can be pronouns like 'hum', 'mein', 'tum', etc., these cause confusion in text classification process. After applying the stop words removal on text data we make sure there are no stop words in the feature vector. The testing result for stop word removal can be observed in Figure 5.

2.1.8 ***Feature extraction***

One of the important preprocessing steps in machine learning problem is feature extraction. In order to facilitate learning this approach builds the feature vector from textual data. A feature vector is simply an n-dimensional vector representation of the dataset with attributes it can be in different format like binary, categorical and numerical.

Term Frequency Inverse Document Frequency (TF-IDF) is a numeric statistical form whose main job is to highlight that how important a word is in a corpus or data collection. The decision is made on the basis of values, higher the TF-IDF words values the stronger relationship in the document. Futhermore, it has been analyzed that the combination of bag-of-words and

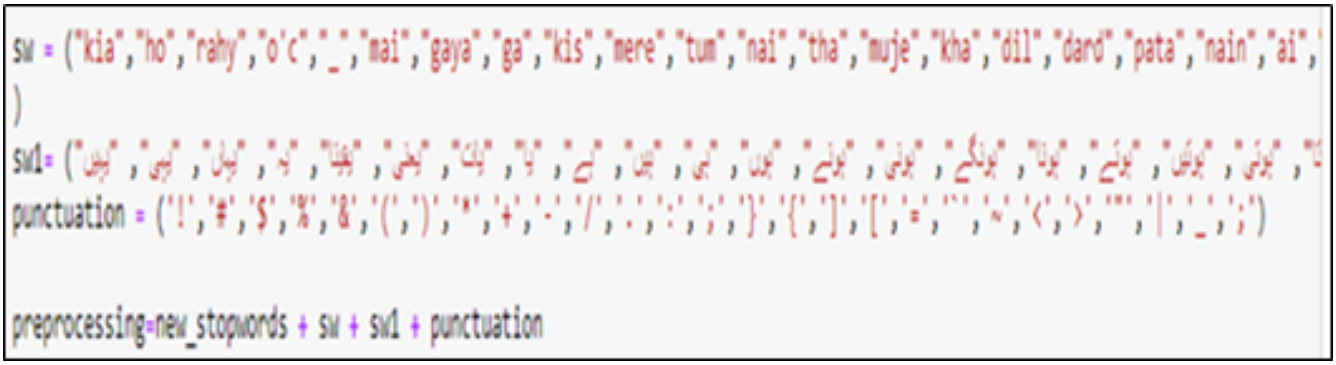

Fig 5. Stop words removal (test results)

TF-IDF can perform better either TF-IDF or bag-of-words[12]. We have considered TF-IDF in as feature extraction technique in order to get the advantages like it is Easy to compute, provides some basic metric to extract the most descriptive terms in the document and similarity can be calculated easily between two documents.

The major problem observed while working with language is that the classifiers and learning algorithms cannot work on raw data directly. To deal with this problem we need to have some feature extraction techniques which convert text data into matrix (or vector) of features. Therefore, during the preprocessing step, the texts are converted to a more manageable representation. The Bag-of-Words and TF-IDF are the two most common approaches used for extracting features from text data. We have used TF-IDF as feature extraction technique in this research study.Rellying only on preprocessing techniques on text is no good approach but feature extraction has significant contribution in the result improvements. The conducted research studies prove that they have extracted features from the corpus successfully. They preferred to choose different features in combination of different text features. Starting from simple features then moved to their eight features set. It was concluded that these set of eight features have been used in text sentiment analysis for English language[13].

Specifically, for each term in our dataset, we calculate a measure called Term Frequency, Inverse Document Frequency, abbreviated to tf-idf. We have used sklearn.feature_extraction.text.TfidfVectorizer library to calculate a tf-idf vector here is our lab experiment along with few other related terms as defined below for the process of feature extraction in this research.

sublinear_df : It is set to True to use a logarithmic form for frequency.

min_df : It is the minimum numbers of documents a word must be present in to be kept.

Nor m: It is set to l2; to ensure all our feature vectors have a euclidian norm of 1.

ngram_range: It is set to (1, 2) to indicate that we want to consider both unigrams and bigrams.

stop_words: It is set to "preprocessing variable (Which holds all the necessary stopwords for Urdu and Roman Urdu Language)" to remove all common pronouns ("a", "the", etc) to reduce the number of noisy features.

2.1.9 ***Feature selection***

The feature selection is an important step in data preprocessing in NLP and same when we are dealing with text data classification. One of research survey on opinion mining elaborated that a model that extracted data from Chinese opinion corpus namely (NTCIR6) used Chi- Square for the purpose of feature extraction from customer reviews[14]. The attributes and work tokens if they are significant in number affects the model in terms of time and efficiency in the case of text classification, so for said purpose feature size reduction is important. This approach contributes in two ways to decrease the vocabulary size and to reduce the noise features that leads to decline classification errors .For the subject purposes WEKA filters have better performances[15].There are three main reasons due to whicht we need to consider the feature selection approaches.1.Curse of dimensionality overfitting, 2.Occam's Razor and 3.Garbage in Garbage out respectively. Then most common approaches used in feature selection are defined below:

2.1.10 ***Filter based***

Filter method is used for the selection of a good subset of the original features set. In filter based approach we specify a metric based on that filter features. The common methods used in Filter approach are the Chi-square/Correlation.

2.1.11 ***Wrapper-based***

The Wrapper based method considers the selection of set of features as a search problem; Recursive Feature Elimination (RFE) is commonly used in Wrapper based approach.

2.1.12 ***Embedded***

There are different algorithms used by embedded method those algorithms have built-in feature selection methods. For example RF and Lasso have their own feature selection methods.

Figure 6 shows the test experimental results of this research in which Chi-square is implemented as feature selection method.

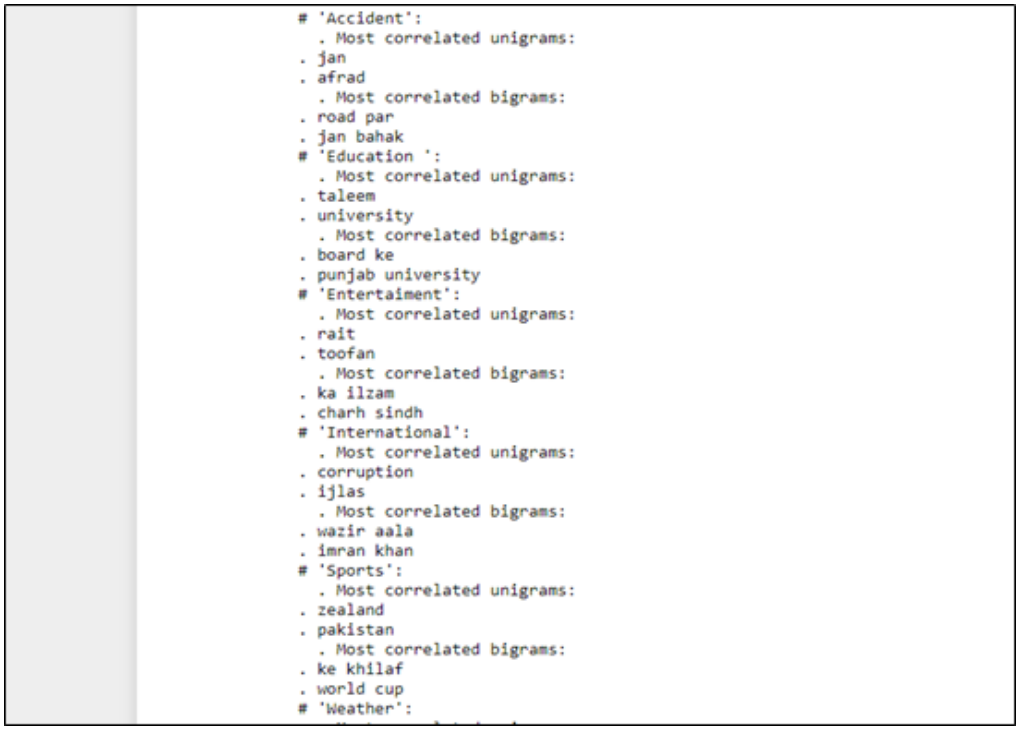

Fig 6. Feature selection (chi square)

3 ML Classification Algorithms

In this research different classification algorithms have been implemented for news text classification and these are explained as below:

3.1 K-NN Algorithm

The implemented K-NN algorithm in this research study provides 53% accuracy compared to our multi-text classification model. The K-NN model as regression works on the basis of average values of all k and that is counted as property value for object and when it works for classification purpose it gives a class membership as output.

K-NN is considered as kind of instance learning or lazy learning based method because it doesn't work until the whole classification is completed. Furthermore, in this method the neighbors who are nearest have more weight on average value than the far ones. Two approaches made for its neighbors selection one either from object property or from K-NN classification value[16].

3.2 Naive bayes classifier

It is a very common method used for text classification known as Naïve Bayes. It belongs to the category of supervised learning algorithms. NB working principle on data is that it should be in distributed form of multiple features .It has been also analyzed

that NB classifier makes assumptions independently. The concept of independent assumptions in NB is that it does not consider the features order which means every feature is independent they don't affect each other when it comes on classification task. One of the advantages of this method is that it performs well in classification even when we have less amount of training data. So in conclusion it works well in this case we have independent features but its performance decrease when the features dependency increased on each other. It also has been analyzed that it possesses good speed and accuracy when used on large databasesr. The accuracy we got from NB in this research study for multi-text classification is significantly high, that is 92%.

3.3 Random Forest

It works on the classification and regression of data. It builds a decision tree at training side. It places classes and mean prediction for regression and classification from each individual tree respectively. It is named random because the classification made by Random Forest is on the basis of random selection of data point/samples from the training data and features are selected randomly during the process of induction. The way it makes prediction for classification is on the basis of majority votes and for regression averaging result is considered. It is more powerful to noise and has good performance improvement in comparison with single tree classifier like C4.5 [17]. It is third text classification algorithm in this research study. It makes different decision trees via the selection of random data samples from the training data using random subspace and tree bagging techniques. In this method different trees are generated via random samples and expecting the classification decision from each tree of the random forest. Once all of the trees assembled in the forest, the labeled data get pass through the trees. Here come the proximities, the proximity of two events get increased by one if both events lie on the same leaf node. In the end, proximities get normalized with the total number of trees in the forest. The accuracy result of Random Forest Algorithm is poor that is 54% in this model of classification.

3.4 Linear SVC

It is the text classification approach we have used in this research study which is frequently used for effective categorization, news filtering, personalization, and information routing is Linear SVM classifier. In SVM the data samples or features scattered in the form of 2D space and observe the closest point that is called support vector. The features are treated as vectors in space, once closest point is found then draw a line connecting them. We have already made a line that separates these two points as far as possible, and the SVM says the best separated line is, that bisects the two points and is perpendicular to the line that connects them. We are making some connection between documents and classes by connecting them as well as separating them to the particular distance. Whenever a document appears, we map it to a point and check the point on the other end of the separating line, to predict its class. According to research a State-of-the-art classification accuracy can be achieved by applying a linear Support Vector Machine (SVM) to a 'bag-of-words' representation of the text, where each unique word in the training corpus becomes a separate feature [18]. Applying this model we got 96% accuracy which is the highest amongst all the classification algorithms.

3.5 Logistic regression

Logistic Regression is used for classification problems and also called linear Regression. It works on the basis of probability for its predictive analysis. The most common function used by Logistic Regression is sigmoid function and considered as one of the complex algorithms. According to this research study Linear Regression performed well on text classification with accuracy of 93% which ranks second from all other classification algorithms.

4 Results and Discussion

In this section various machine learning classifiers have been discussed which are selected training for classification of text data. Most common supervised learning classifiers which have implemented in this research are: Naive Bayes Classifier, Logistic Regression, Random Forest Classifier, Linear SVC, and K-Neighbors. Table 1 shows the accuracy results of each classifier which are obtained through text classification from corpus. It is worthwhile to mention that SVM classifier attained highest accuracy which is 96% amongst rest of tested algorithms. The K-NN has the lowest accuracy from all classifiers. Similarly, the Logistic Regression has proven better in results as comparison with other classification algorithms. When subject ML models were trained on both categories of Urdu and Roman Urdu text data then each algorithm was tested with perspective of its results via lab experiments on the text data of multi-categories Roman and Urdu is shown in the Figure 7 which describes the text data prediction results of various Machine Learning algorithms. Most of the models predicted well during the test on multi-text data in order to analysis efficiency and accuracy of each classification models we have plotted their accuracy results. The

tested accuracy benchmark results of classification algorithms have been shown in Figure 8. However, Figure 9 shows generated confusion matrix which provides actual and predicted outcomes in order to judge the accuracy of model.

Table 1. Text classification accuracy results

Sl.no	ML Classification Algorithms	Accuracy
1	K-Nearest Neighbor (K-NN)	53.27%
2	Linear SVC	96.1%
3	Logistic Regression	93.43 %
4	Naive Bayes Classifier	92.42 %
5	Random Forest Classifier	54.48 %

```
print("ROMAN URDU MultinomialNB")
print(1, clf.predict(count_vect.transform(["Darja Hararat 44 Degree Tak Janay Ka Imkaan"])))
print(2, clf.predict(count_vect.transform(["Aj afridi nai 50 run score kiye"])))
print(3, clf.predict(count_vect.transform(["Karachi mai zor dar dhamaka"])))
print(4, clf.predict(count_vect.transform(["Karachi mai aj Taiz Hawaon chalin"])))
print(5, clf.predict(count_vect.transform(["Karachi mai hadsa 3 log mar gaye"])))
print("URDU MultinomialNB")
print(6, clf.predict(count_vect.transform(["طالبان، امریکہ ایک بار پھر مذاکرات کی میز پر"])))
print(7, clf.predict(count_vect.transform(["چیف الیکشن کمشنر: عمران خان نے تین نام تجویز کر دیے"])))
print(8, clf.predict(count_vect.transform(["انٹیا: خاتون کو عدالت کے راستے میں آگ لگا دی گئی"])))
print("\n")
print("Roman URDU LogisticRegression")
print(1, clf1.predict(count_vect.transform(["Darja Hararat 44 Degree Tak Janay Ka Imkaan"])))
print(2, clf1.predict(count_vect.transform(["Aj afridi nai 50 run score kiye"])))
print(3, clf1.predict(count_vect.transform(["Karachi mai zor dar dhamaka"])))
print(4, clf1.predict(count_vect.transform(["Karachi mai aj Taiz Hawaon chalin"])))
print(5, clf1.predict(count_vect.transform(["Karachi mai hadsa 3 log mar gaye"])))
print("URDU LogisticRegression")
print(6, clf1.predict(count_vect.transform(["طالبان، امریکہ ایک بار پھر مذاکرات کی میز پر"])))
print(7, clf1.predict(count_vect.transform(["چیف الیکشن کمشنر: عمران خان نے تین نام تجویز کر دیے"])))
print(8, clf1.predict(count_vect.transform(["انٹیا: خاتون کو عدالت کے راستے میں آگ لگا دی گئی"])))
print("\n")
print("Roman URDU RandomForestClassifier")
print(1, clf2.predict(count_vect.transform(["Darja Hararat 44 Degree Tak Janay Ka Imkaan"])))
print(2, clf2.predict(count_vect.transform(["Aj afridi nai 50 run score kiye"])))
print(3, clf2.predict(count_vect.transform(["Karachi mai zor dar dhamaka"])))
print(4, clf2.predict(count_vect.transform(["Karachi mai aj Taiz Hawaon chalin"])))
print(5, clf2.predict(count_vect.transform(["Karachi mai hadsa 3 log mar gaye"])))
print("URDU Random ForestClassifier")
print(6, clf2.predict(count_vect.transform(["طالبان، امریکہ ایک بار پھر مذاکرات کی میز پر"])))
print(7, clf2.predict(count_vect.transform(["چیف الیکشن کمشنر: عمران خان نے تین نام تجویز کر دیے"])))
print(8, clf2.predict(count_vect.transform(["انٹیا: خاتون کو عدالت کے راستے میں آگ لگا دی گئی"])))
print("\n")
print("Roman URDU LinearSVC")
print(1, clf3.predict(count_vect.transform(["Darja Hararat 44 Degree Tak Janay Ka Imkaan"])))
print(2, clf3.predict(count_vect.transform(["Aj afridi nai 50 run score kiye"])))
print(3, clf3.predict(count_vect.transform(["Karachi mai zor dar dhamaka"])))
```

Fig 7. ML models multi-textdata predictions

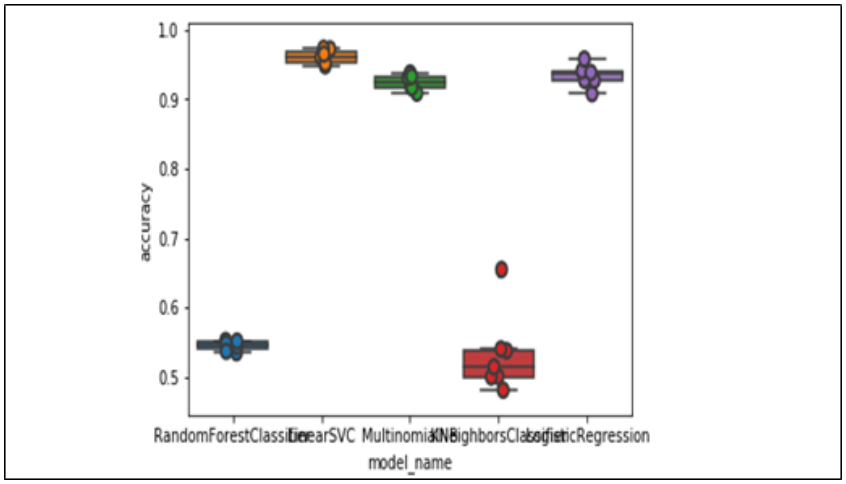

Fig 8. ML algorithms accuracy results

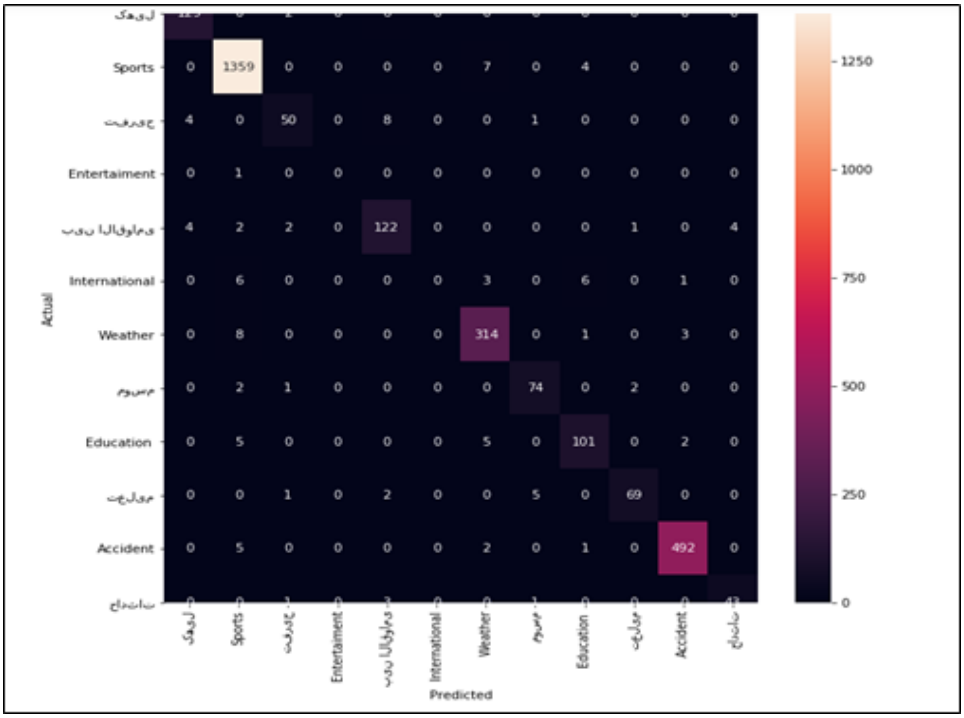

Fig 9. Confusion matrix

5 Conclusion

In this work different text classification algorithms were implemented for multi-text classification including Naive Bayes Classifier, Logistic Regression, Random Forest Classifier, Linear SVC, and K-Neighbors Classifier. Prior to Multi-Text classification various NLP preprocessing techniques were applied namely, data cleaning, feature extraction, and feature selection. However, obtained results showed that Linear SVC and Logistic Regression classifiers provided 96% and 93% accuracy compared to other tested classifiers on similar dataset for multi-text classification of Urdu language.

Future work

In future, we plan to extend our work by taking more categories of news text data of Urdu Roman Language as to verify existing algorithms accuracies.

References

1) Ghulam H, Zeng F, Li W, Xiao Y. Deep Learning-Based Sentiment Analysis for Roman Urdu Text. *Procedia Computer Science*. 2019;147:131–135. doi:10.1016/j.procs.2019.01.202.
2) Arif H, Munir K, Danyal AS, Salman A, Fraz MM. Sentiment analysis of roman urdu/hindi using supervised methods. *Proceedings of ICICC*. 2016;8:48–53. doi:10.22581/muet1982.1902.20.
3) Hassan S, Muhammad F, Ali S, Wasi S, Javeed I, Hussain SN, et al. Roman-Urdu News Headline Classification with IR Models using Machine Learning Algorithms. *Indian Journal of Science and Technology*. 2019;12(35):1–9. doi:10.17485/ijst/2019/v12i35/146571.
4) Dwivedi SK, Arya C. Automatic text classification in information retrieval: A survey. In: and others, editor. Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. 2016;p. 1–6. doi:10.1145/2905055.2905191.
5) Li Z, Shang W, Yan M. News text classification model based on topic model. *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. 2016;p. 1–5.
6) Zhao W. Deep Active Learning for Short-Text Classification. 2018. Available from: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1135693.
7) Korde V. Text Classification and Classifiers:A Survey. *International Journal of Artificial Intelligence & Applications*. 2012;3(2):85–99. doi:10.5121/ijaia.2012.3208.
8) Zaidi SAA, Hassan SM. Urdu/Hindi News Headline, Text Classification by Using Different Machine Learning Algorithms. . doi:10.13140/RG.2.2.12068.83846.
9) Zheng Y, IEEE. An Exploration on Text Classification with Classical Machine Learning Algorithm. In: and others, editor. 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). IEEE. 2019;p. 81–85. doi:10.1109/MLBDBI48998.2019.00023.

10) Londo, Yovellia GL, Kartawijaya DH, Ivariyani HT, WP YSP, P A, et al. A Study of Text Classification for Indonesian News Article. In: Rafi M, Ariyandi D, et al., editors. 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT). IEEE. 2019;p. 205–208. doi:10.1109/ICAIIT.2019.8834611.
11) Singh G, Kumar B, Gaur L, Tyagi A. Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification. In: and others, editor. 2019 International Conference on Automation, Computational and Technology Management (ICACTM). IEEE. 2019;p. 593–596. doi:10.1109/ICACTM.2019.8776800.
12) Bang A, Wu W, Han H, et al. Deep Active Learning for Text Classification. In: and others, editor. Proceedings of the 2nd International Conference on Vision, Image and Signal Processing. 2018;p. 1–6. Available from: http://dx.doi.org/10.1145/3271553.3271578. doi:10.1145/3271553.3271578.
13) Rafique A, Malik MK, Nawaz Z, Bukhari F, Jalbani AH, et al. Sentiment Analysis for Roman Urdu. *Mehran University Research Journal of Engineering and Technology*. 2019;38(2):463–470. doi:10.22581/muet1982.1902.20.
14) Rashid A, Anwer N, Iqbal M, Sher M, et al. A survey paper: areas, techniques and challenges of opinion mining. *International Journal of Computer Science Issues (IJCSI)*. 2013;10(6):18.
15) Bilal M, Israr H, Shahid M, Khan A, et al. Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques. *Journal of King Saud University - Computer and Information Sciences*. 2016;28(3):330–344. Available from: https://dx.doi.org/10.1016/j.jksuci.2015.11.003. doi:10.1016/j.jksuci.2015.11.003.
16) Usman M, Shafique Z, Ayub S, Malik K. Urdu Text Classification using Majority Voting. *International Journal of Advanced Computer Science and Applications*. 2016;7(8):265–273. doi:10.14569/ijacsa.2016.070836.
17) Ali J, Khan R, Ahmad N, Maqsood I. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*. 2012;9(5):272–272.
18) Ikonomakis M, Kotsiantis VS, Tampakas. Text classification using machine learning techniques. *WSEAS transactions on computers*. 2005;4(8):966–974.