DOI: 10.22067/cke.2021.69212.1004

Classification of Persian News Articles using Machine Learning Techniques*

Research Article

Sareh Mostafavi[1] Bahareh Pahlevanzadeh[2] Mohammad Reza Falahati Qadimi Fumani[3]

Abstract: Automatic text classification, which is defined as the process of automatically classifying texts into predefined categories, has many applications in our everyday life, and it has recently gained much attention due to the increased number of text documents available in electronic form. Classifying News articles is one of the applications of text classification. Automatic classification is a subset of machine learning techniques in which a classifier is built by learning from some pre-classified documents. Naïve Bayes and k-Nearest Neighbor are among the most common algorithms of machine learning for text classification. In this paper, we suggest a way to improve the performance of a text classifier using Mutual information and Chi-square feature selection algorithms. We have observed that MI feature selection method can improve the accuracy of Naïve Bayes classifier up to 10%. The empirical results show that the proposed model achieves an average accuracy of 80% and an average F1-measure of 80%.

Keywords: Automatic Persian text classification, K-Nearest Neighbor, Naïve Bayes, News text classification, Text categorization, Text mining

1. Introduction

With the rapid growth of electronic text documents generated every day on the Internet, text classification has gained more importance in recent years [1]. Text classification, also known as text categorization, is the process of assigning class labels to a text document according to its content [1]. Text classification has been successfully used in domains such as topic detection, spam e-mail filtering, news text classification, web page classification, author recognition, and sentiment analysis.

The news was not easily accessible until the beginning of the 21st century, but today the news is readily available on the Internet. Moreover, in the past only a small group of people needed international news, such as politicians, and the news required by most people was limited to local news. In other words, ordinary people did not need global news and therefore did not follow it; however, today people follow the worldwide news and show more interest in it. Therefore, news text classification is now a challenging field in text mining approaches. News text classification is defined as classifying news articles in one or more classes. Classification of news helps the users to easily access their desired news without wasting their time.

Considering the great number of texts available, manually classifying text documents is time-consuming, expensive, and even impossible; therefore, it is better to use automatic classification techniques for classifying news articles. In this regard, there are two main approaches to classify documents automatically: rule-based approach and machine learning approach. In the rule-based approach, a set of rules are written by human experts, and the classification process is done according to these rules. In machine learning approaches, a classifier is built by learning from some pre-classified documents.

One of the most challenging tasks in text classification is feature selection [2, 3]. Feature selection is the process in which a subset of the most relevant features is selected from the feature space [4]. This paper uses two feature selection methods, including Mutual Information (MI) and Chi-square (CHI), to enhance the performance of classifier model. In this paper, a comparison of these two methods of feature selection is presented and discussed as well.

A few research pieces have been conducted on building classifier models to classify texts in Persian. These studies are mostly done using Hamshahri corpus as their dataset, and there is a lack of research on any other Persian dataset. Automatic Persian text classification based on Persika corpus, a collection of Persian news articles collected from ISNA,[4] has been studied once in which MI and Chi-square are applied to improve the performance of the classifier algorithm. Therefore, the main aim of the present study is to build a classifier model for Persian news texts based on Persika dataset using KNN and Naïve Bayes as the classifier algorithms and the MI and chi-square as the feature selection algorithms to see how these feature selection methods improve the accuracy of the classifier.

In the second section of the paper, a brief literature on text classification is given to review the most common text classification techniques used by researchers. Besides, a comprehensive research literature on Persian text classification and, more specifically, on Persian news text classification is offered. In the third section of the paper, the method used in this study is discussed. In the fourth section, the evaluation metrics are introduced, and the model is evaluated using different evaluation metrics. And in the last section, the evaluation of the final proposed model for Persian news classification is discussed. Finally, the paper ends with a conclusion and some avenues for future studies are also suggested.

* Manuscript received March, 2, 2021; accepted May, 17, 2021.

[1] MSc Student, Department of Computational Linguistics, Regional Information Center for Science and Technology (RICeST), Shiraz, Fars, Iran.

[2] Corresponding Author. Assistant Professor, Department of Design and System Operations, Regional Information Center for Science and Technology (RICeST), Shiraz, Fars, Iran. Email: pahlevanzadeh@ricest.ac.ir.

[3] Associate Professor, Department of Computational Linguistics, Regional Information Center for Science and Technology (RICeST), Shiraz, Fars, Iran.

2. Related Work

2.1 Overview of the State-of-the-art Algorithms of Text Classification

According to Dalal and Zaveri , the history of text classification goes back to 1961. In the traditional approach, text classification was done using knowledge engineering techniques in the 1980s [5], which consisted of manually defined rules. Because the method is based on some logical rules, it is known as the rule-based approach. Since in the rule-based approach, the logical rules are written by human experts, building models in this approach is so expensive and time-consuming. Moreover, this approach is more computationally complicated [6]. Because of the problems of the rule-based approach, the machine learning approach has gained much popularity and has attracted many researchers' attention since the early 1990s [7, 8]. The machine learning approach is faster and more straightforward and does not need a vast number of human experts.

During the past decades, many machine learning techniques of text classification have been introduced and studied by researchers of different languages, most of which are for the English language. There are different machine learning algorithms for text classification, among which the most common ones are Naïve Bayes (NB) [9], k-Nearest Neighbor (KNN) [10], Decision Tree (DT) [11], Support Vector Machine (SVM) [12], and Neural Networks (NN) [13].

An exhaustive overview of the state-of-the-art algorithms of machine learning for text classification has been achieved by many authors such as [5-8]. Therefore, we only provide an overview of approaches used for text classification.

2.1.1 Naïve Bayes

Probabilistic classifiers have attracted much attention in recent years. Naïve Bayes classifiers are the most popular probabilistic approaches used in text classification in the literature [2]. Naïve Bayes classifiers are a group of classifiers using the Bayes rule with the assumption that the distribution of all terms in a document is independent of others. Naïve Bayes classifiers are called *Naïve* since the early 90s because the so-mentioned assumption is not true in the real world [2].

An experiment on the naïve Bayes text classifier was carried out by McCallum and Nigam (1998) [9]. In their work, the authors compared two standard event models of Naïve Bayes (i.e., multinomial event model and multivariate Bernoulli model). McCallum and Nigam believe that the Naïve Bayes algorithm is the simplest model among probabilistic models. Besides, they maintain that Naïve Bayes classifier works surprisingly well even though its primary assumption about the independence of attributes is not true in the real world. Other researchers believe that the performance of Naïve Bayes is very good in comparison with other text classification algorithms [14-18]. In more recent research, scientists have attempted to improve the performance of naïve Bayes using different methods [19-21]. Many researchers have attempted to improve Naïve Bayes by applying feature selection algorithms on the classifier to reduce the high dimensionality of feature space [22-24].

2.1.2 K-Nearest Neighbor (KNN)

K-nearest neighbor is an example-based non-parametric classifier; it is one of the simplest and most efficient classifiers used in text classification. KNN is mostly used in text classification for its low calculation time and low computational complexity [25]. KNN classifiers are grouped under the category of lazy learners. In fact, they are called lazy because they postpone the decision making about the test document until meeting all the training documents. Yang and Pederson (1997) [26] were among the first authors investigating the KNN classifier; however, some other researchers have also shown KNN to be effective [26-33]. In more recent research, scientists have attempted to improve the performance of KNN using different methods [34], whereas some researchers have attempted to use feature selection methods to improve the performance of KNN classifier [35, 36].

2.2 Overview of Studies on Text Classification for the Persian Language

In a pioneering study [37], a distributed classification of Persian news articles was proposed using Mapreduce as a programming model and the Hamshahri dataset as the corpus. The results of this study showed an average recall of 63.75% and an average precision of 52.67% [37].

In another study, the researchers used the Learning Vector Quantization (LVQ) algorithm for classifying Persian texts and compared their proposed method with KNN and SVM classifiers. They showed that the LVQ model would perform faster than other algorithms in terms of classifying Persian texts. They have reached an average f-measure of 89% as such [38].

In another model for Persian news text classification, KNN and SVM classifiers and TF-IDF feature weighting approach were used through which Hamshahri dataset was used as the corpus. The authors of this paper showed that KNN would perform better in classifying Persian texts. They have reached an average f-measure of 94% using their proposed model [29].

In another study, the researchers suggested using a thesaurus to improve the SVM classifier for Persian news texts. They used a corpus of news articles collected from different newspapers and Wikipedia and achieved a micro f-measure of 89% [39].

In another study [40], KNN classifier for classifying Persian news texts was proposed using the n-gram model to improve the efficiency of classifier. The authors compared their proposed model with the model in which a thesaurus would be used to improve the performance of the SVM classifier. In their study, the Hamshahri dataset was used to train the classifier; they gained a micro f-measure of 91% [40].

In another study [41], the KNN classifier and the WordNet were used to improve the performance of KNN. In addition, they applied two feature selection algorithms, IG and PCA, and they gained an accuracy of 88.18% using the Hamshahri dataset as the corpus [41].

Another study suggested using a PSA feature selection algorithm to improve the performance of the classifier for Persian text classification [42]. This study used a corpus of Persian news articles conducted by the authors. The authors of this paper also compared their proposed feature selection algorithm with two other feature selection methods, chi-square and correlation coefficient. They have gained an f-measure of 87% for their proposed model.

Table 1. Related Literature on the Persian Language

Paper/Authors	year	Dataset (corpus)	Measurement parameters
[37] Esmaeili et al.	2005	Hamshahri	recall 63.75% precision 52.67%.
[38] Pilevar et al.	2009	Hamshahri2	f-measure 89%
[29] Farhoodi and Yari	2010	Hamshahri2	f-measure 94%.
[39] Maghsoodi and Homayounpour	2011	Sample corpus	f-measure 89%.
[40] Elahimanesh et al.	2012	Hamshahri	f-measure 91%.
[41] Parchami et al.	2012	Hamshahri	accuracy 88.18%
[42] Bagheri et al.	2014	Sample corpus	f-measure 87%
[43] Ahmadi et al.	2016	Bijankhan dataset	accuracy of 87%.
[44] Dastgheib and Koleini	2019	Scholarly articles from RICeST	f-measure 83%

In another study, the authors investigated applying topic models for Persian text classification [43]. They used an SVM classifier for their investigation and reached an accuracy of 87% [44] by using the latent semantic indexing (LSI) model instead of the traditional model of representing texts for text classification called the Vector space model. They used KNN and SVM classifier algorithms to classify the scholarly articles collected from the RICeST[15]Persian articles repository. They showed that using the LSI model would improve the performance of the classifier model. They also reached an f-measure of 83% using their proposed model.

2. Materials and Methods

As shown in Figure 1, building a text classifier usually consists of the following steps:

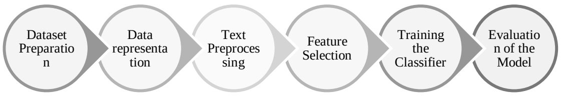

Figure 1. Building a Classifier Model

This paper has followed the steps above to build a model for classifying Persian news articles.

2.1 Dataset Preparation

As mentioned earlier, this research is a case study which uses the Persika corpus to train a classifier. Persika dataset is a corpus of Persian news articles collected from the ISNA news website, one of the most reliable and known news agencies for Persian news. Persika is the only standard news corpus which uses articles from ISNA [45]. Persika contains 11000 news articles categorized under 11 categories. The data in Persika are balanced. That is, each of the 11 classes in Persika consists of 1000 news articles. These 11 classes concern with sports, economy, culture, religion, history, politics, science, society, education, judiciary, and hygiene.

The dataset used in text classification tasks are divided into two parts: the train set and the test set [46]. There are different ways of dividing the dataset into the train and test sets. In this study, we use cross-validation (CV) to do this job. In cross-validation, the dataset is divided into k folds. The model is then built using one fold as the test set and k-1 folds as the training sets [47]. The process is repeated k times so that each of the k folds has been used once as the test set [47]. Then, the average error of these k times repetition is called the cross-validation error, which shows the performance of model. In other words, true positive, true negative, false positive, and false negative for each fold are calculated, and then the evaluation measures (i.e., accuracy, precision, recall, and f-measure) are calculated. In this study, a 10-fold cross-validation method is adopted.

2.2 Data Representation

Data in this study are represented using the Bag-of-Words model (BoW), which is the most common way of text document representation [48]. In this model, a document is represented as a vector $V=\{tw_1, tw_2, \ldots\ldots tw_v\}$., where tw_i is the weight assigned to the term I [49]. In the BOW model, the order of the words is not considered, and only the frequency of each term is considered.

2.3 Data Preprocessing

Since most of the available text documents are in an unstructured form, the text on which the training process is based should be preprocessed. Preprocessing is an essential part of building a classifier model that can positively affect the model's accuracy. In this research, the preprocessing stage consists of three steps, which are shown in Figure 2.

[1] www.ricest.ac.ir

Tokenization
Normalization
Stop Words Removal

Figur 2. Preprocessing

In the tokenization process, a text document is broken into its tokens. In the normalization process, the non-standard tokens and structures in a text document are either removed or standardized. Stop words are frequent words in a text document that do not contain important information. Removing the stop words reduces the complexity of the model and improves the classifier's performance.

2.4 Feature Selection

Text classification usually faces the problem of the high dimensionality of feature space, which is the vast number of terms in a text document [50]. Thus, a process is needed to reduce the dimensions of feature space by choosing more relevant and effective features [51]. This process is called *dimensionality reduction*, also known as *feature selection*. Feature selection reduces the computational complexity of the model and therefore improves the classifier performance [52]. Feature selection can improve the efficiency and accuracy of a text classifier [54]. It is also beneficial in reducing the overfitting (i.e., when a classifier is adjusted to both the dependent characteristics of the training data and the constitutive features of the categories) [5].

There are two main types of feature selection algorithms: Filter methods and Wrapper methods [53]. Wrapper methods use the learning algorithm to evaluate the features. The accuracy of the learning algorithm based on a feature reveals the effectiveness of that feature. Wrapper methods are more time-consuming than filter methods because they have to train a classifier to evaluate each feature and that they work only for a limited set of classifiers. In contrast to wrapper methods, filter methods work independently from the learning algorithm and are less time-consuming. Filter methods measure the importance of each feature using some functions and then select the most essential features.

Since filter methods are more straightforward and less time-consuming than wrapper methods, they are more suitable for text classification tasks in which there is a large number of features. Some of the most popular feature selection algorithms included in the filter group are x^2 statistics (CHI), Information Gain (IG), Mutual Information (MI), and document frequency (DF).

In this paper, we use two feature selection algorithms, namely Mutual Information (MI) and Chi-square (CHI), to see to what extent they improve the efficiency of the classifier and compare the performance of these two feature selection methods. In addition, TF-IDF, which is a term weighting method, is applied before MI and CHI.

2.4.1 Term Frequency-Inverse Document Frequency (TF-DF)

TF-IDF is a crucial method of weighting features in a text document to select the most relevant features. The relevance of the word to the document is calculated by the weight assigned to each term in a text document. TF-IDF is a weighting method that assigns a weight to a term by considering the term frequency and inverse document frequency [1]. In TF-IDF, a word takes a high weight if its frequency in the document is high, and it takes a low weight if the document frequency, the number of training documents containing term t is high [54]. TF-IDF is calculated using the formula below [55]:

$$W_{i,j} = tf_{i,j} * \log \frac{N}{df_i} \tag{1}$$

2.4.2 Chi-square (CHI)

CHI is a recognized statistical test that calculates the correlation between term t and class c_i [56]. In other words, it measures the amount to which the term t and class c_i are correlated. Chi-square outperforms other feature selection algorithms, such as information gain and document frequency [57]. CHI is calculated using the formula below [54].

$$x2\ (t,c) = \frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)} \tag{2}$$

Where N is the number of all training documents, A is the number of documents in class c that contain term t, B is the number of documents that contain term t and are not in class c, C is the number of documents that are in class c and do not contain the term t, and finally D is the number of documents in class c that do not contain the term t [54].

2.4.3 Mutual Information (MI)

MI is a measure calculating the dependency between two variables. These two variables in text classification tasks are a term t and a class c. If the MI between a term t_k and a class c_i is zero, then t_k and c_i are entirely independent.
MI is defined in the following: [58]

$$MI\ (t,c) = \log \frac{p(t,c)}{p(t)*p(c)} \tag{3}$$

2.4.4 Training the Classifier

As mentioned earlier, NB and KNN are among the simplest, most effective, and most applicable algorithms. These two classifier algorithms have not been used and compared for Persian news text classification by applying MI and Chi-square feature selection algorithms. Therefore, in this study, NB and KNN classifiers are used to build the classifier model using MI and chi-square as the feature selection methods to see how these feature selection algorithms improve the efficiency of the model.

2.4.5 Naïve Bayes classifier

As mentioned earlier, the multinomial Naïve Bayes (MNB) classifier is a probabilistic classifier especially designed for text classification. Although in Naïve Bayes the main assumption about the complete independence of the attributes is not true in the real world, it performs surprisingly well in text classification [59,60]. Naïve Bayes classifier uses the Bayes rule to estimate the probability that document d belongs to class C.

This is the so-called Bayes rule on which the Naïve Bayes classifiers are based [9].

$$P(c \mid d) = \frac{P(c)\,P(d|C)}{P(d)} \tag{4}$$

In text classification, a document is usually represented as a vector v={t_1, t_2, ….. , t_k}. Given the fact that for i ≠ j, v_i

and v_j are conditionally independent in terms of the class c. We can rewrite Eq. (3.4) as:

$$P(c \mid d) = P(c) * \frac{\prod_{j=1}^{k} P(v_j|c)}{P(d)} \quad (5)$$

After computing P(c | d), we can construct maximum a posterior (MAP) classifier by selecting the category that maximizes P(c | d) using the formula below [2]:

$$C = argmax_{c \in C} \{P(c \mid d)\} \quad (6)$$

$$C = argmax_{c \in C} \{P(c) * \frac{\prod_{j=1}^{k} P(v_j \mid c)}{P(d)}$$

$$C = argmax_{c \in C} \{P(c) * \prod_{j=1}^{k} P(v_j \mid c)\} \quad (7)$$

There are two models of Naïve Bayes classifier used for text classification: Multivariate Bernoulli model and Multinomial model. In the Multivariate Bernoulli model, the frequency of the terms is ignored because in this model the document is represented by a vector of binary features representing the presence or absence of the words in the text. In this model, a vocabulary V is given. A document is represented with a vector of | V| dimensions [53]. The kth dimension of the word corresponds to word w_k from V and is either 1 or 0, indicating whether word w_k occurs in the document. If document d_i is represented with a vector {t_1, t_2, …. t_v} then we can compute p (d_i | c_j) as:

$$P(d_i|c_j) = \prod_{k=1}^{|v|} P(w_k|c_j)^{t_k} (1 - P(w_k|c_j))^{1-t_k} \quad (8)$$

In the Multinomial model, the frequency of the terms is considered because in this model a document is represented using the bag-of-words model. In this model, the order of the words is not considered. The Multinomial model is more effective when working with large datasets. Therefore, for text classification in which the vocabulary size is large, this model would be better than the Multivariate Bernoulli model.

If the frequency of word w_k in document d_i is represented as N_{ik}, then P (d_i | c_j) can be computed in the following:

$$P(d_i|c_j) = P(|d_i|) \, |d_i|! \prod_{k=1}^{|v|} \frac{P(w_k|c_j)^{n_{ik}}}{n_{ik}\,!} \quad (9)$$

In both models, the probability of class cj, that is P(c_j), can be computed as:

$$P(c_j) = \frac{1+n_j}{1+n_{all}} \quad (10)$$

Where n_j is the number of documents in class c_j and n_{all} is the number of documents in class c_j in the training set D Also, P(w_k | c_j) can be computed in the following:

$$P(w_k|c_j) = \frac{1+n_{cjk}}{n_{all}+n_j} \quad (11)$$

Where n_i is the number of words in class c_j, N_{cjk} is the number of word wk in class $c_{j,}$ and N_{all} is the number of all words in the training set D.

2.4.6 K-Nearest Neighbor (KNN)

As we have mentioned before, KNN is an instance-based classifier with high accuracy in text classification.

The main idea in KNN is comparing the test document with a set of neighboring training sets. In fact, in KNN, the similarity between the test set and the k training sets is calculated using a similarity measure. Then, the test document is labeled with the class by which most of its neighbors are labeled.

As mentioned earlier, for building a KNN classifier, we need to determine a threshold k. The choice of parameter k is an important and effective step in building a KNN classifier. In this study, we use the empirical method of determining k. This method is explained in section 4.

Different similarity measures can be used in KNN classifiers, such as Euclidean distance, Cosine similarity, etc. In this research, the Euclidean distance is used to measure the similarity between the test document and the training documents. Euclidean distance between two documents is calculated using the formula below [59]:

$$s\,(d_i, d_j) = \sum_{f=1}^{n} (d_{if} - d_{jf})^2 \quad (12)$$

3. Results

This section first briefly introduces the evaluation metrics used to evaluate the models. Different classifier models using Naïve Bayes and KNN classifiers and MI and CHI as feature selection algorithms are built and evaluated. Finally, a model for classifying Persian news articles is suggested, and the proposed model is evaluated in terms of accuracy, precision, recall, and f-measure.

3.1 Evaluation Metrics

The evaluation of a document classifier is usually done experimentally. The experimental evaluation of a classifier usually measures its effectiveness, which is its ability to make the right classification decision. There are different metrics for measuring the classification effectiveness, including precision, recall, and f-measure and accuracy. In this paper, the evaluation of classifier models is done in terms of these four evaluation metrics using the contingency table. The contingency table, as shown in Table 2, indicates the distribution of correctly and wrongly classified documents.

Table 2. Contingency Table

Category set C= {c_1, c_2,…..,$c_{\|cl\|}$ }		Expert judgment	
		Yes	No
Classifier judgment	Yes	TP_i	FP_i
	No	FN_i	TN_i

***In the above table, TP is the number of documents that are correctly labeled positive. TN is the number of documents that are correctly labeled negative. FP is the number of documents that are wrongly labeled positive, and FN is the number of documents that are wrongly labeled negative.

***To measure P and R's values, two different methods can be adopted: micro-averaging and macro-averaging. In micro-averaging, P and R are calculated by summing total single decisions about each category [5]. In macro-averaging, firstly, P and R are calculated for each category and the average of the results of the different categories [5]. In this paper, the macro-averaging method is used; therefore, wherever in this paper the word average is used with the evaluation metrics, the macro-averaging method is meant.

3.1.1 Accuracy

The accuracy measure is the ratio of correctly predicted observation to the total observation. The accuracy formula is as follows [5]:

$$A = \frac{TP+TN}{TP+FP+TN+FN} \tag{13}$$

3.1.2 Precision

The precision measure is the ratio of correctly predicted positive observation of the ratio of all the retrieved data. The precision formula is as follows [5]:

$$P = \frac{TP}{TP+Fp} \tag{14}$$

3.1.3 Recall

The recall measure is the ratio of correctly predicted positive observations to all the observations in actual class positive. The recall formula is as follows [5]:

$$R = \frac{TP}{TP+FN} \tag{15}$$

3.1.4 F-measure

The f-measure is the weighted average of precision and recall. It is calculated as follows [5]:

$$F = \frac{2RP}{R+P} \tag{16}$$

3.1.5 Evaluation

The evaluation of models has been done in 4 phases as shown below:

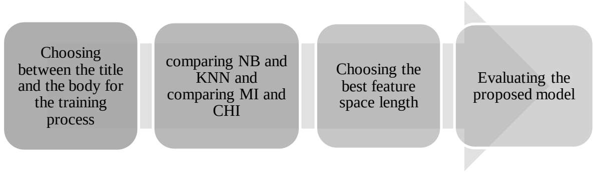
Choosing between the title and the body for the training process
comparing NB and KNN and comparing MI and CHI
Choosing the best feature space length
Evaluating the proposed model

Figure 3. The Evaluation Process

3.1.6 Choosing between the title and the body for the training process

Persika dataset has seven columns, namely news-ID, title, body, date, time, category 1, and category 2. In this paper, we deal with the title and the body of the news articles and the column named category 2. To train the classifier, we can use the title, the body, or both of them. To see which of these three was the most effective way of gaining high accuracy, we compared the use of these three situations. The experimental results showed that using both the title and the body of the news articles can be the most effective ones.

3.1.7 Comparing NB with KNN and comparing MI with CHI

In the second phase of our experiment, we compared NB with KNN classifiers to see which one would outperform the other in classifying Persian news articles.

As the choice of parameter k is an essential part of building a KNN classifier, we used different amounts of 1, 3, 5, 7, and 9 for parameter k to see what amount of k would give the best accuracy of the classifier. Table 3 shows the result of this experimental analysis.

Table 3. Choice of Parameter k

Evaluation Metric	K=1	K=3	K=5	K=7	K=9
Average Accuracy	0.72	0.73	0.75	0.76	0.76
Average Precision	0.72	0.75	0.76	0.76	0.77
Average Recall	0.72	0.73	0.75	0.76	0.76
Average F-measure	0.71	0.73	0.75	0.75	0.76

As shown in Table 3, using nine nearest neighbors among the training set would be the best. Therefore in the rest of this paper, the variable k in the KNN algorithm has a value of 9.

The classifier models were built using NB and KNN classifiers and MI and Chi-square feature selection algorithms for the comparison purpose.

Table 4 shows the performance of the KNN classifier with and without applying feature selection methods.

Table 4. Evaluation of the KNN classifier

Evaluation Metric	KNN.MI.	KNN.CHI.	KNN
Average accuracy	0.76	0.76	0.76
Average Precision	0.77	0.77	0.77
Average recall	0.76	0.76	0.76
Average f-measure	0.76	0.75	0.76

Contrary to expectations, applying MI and CHI does not improve the performance of the KNN classifier.

Table 5 shows the performance of the NB classifier with and without applying feature selection methods.

Table 5. Evaluation of the NB Classifier

Evaluation Metric	NB	NB.CHI.	NB.MI.
Average accuracy	0.73	0.79	0.79
Average Precision	0.64	0.81	0.81
Average recall	0.73	0.79	0.79
Average f-measure	0.67	0.78	0.78

It can be seen from Table 5 that the performance of the NB classifier significantly improves when applying MI and CHI feature selection algorithms. As shown in Table 5, the average precision of Naïve Bayes is improved by about 17%. Its average recall is improved by 6%, its accuracy is improved by 6%, and its f-measure is improved by 11% when applying the MI feature selection method.

Table 6 shows the comparison of the KNN (in its best form) with the NB (in its best form).

Table 6. Comparison of the NB with the KNN

Evaluation Metric	NB.MI.	KNN
Average accuracy	0.79	0.76
Average Precision	0.81	0.77
Average recall	0.79	0.76
Average f-measure	0.78	0.76

As shown in Table 6, the results show that the NB classifier can outperform the KNN when applying feature selection methods and that the MI feature election method can optimize the results.

3.1.8 Choosing the Best Feature Space Length

In the fourth phase of the evaluation process, the best size of feature space is made. For this purpose, the performance of the NB classifier using MI feature selection was evaluated several times through several different subsets of feature space. The subsets were generated by selecting 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 50, and also 80% of the total features. The results are shown in Figure 4.

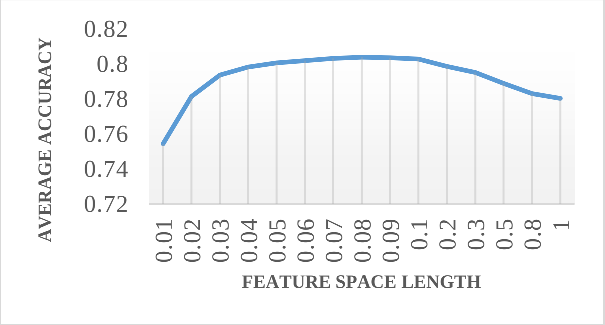

Figure 4. Feature space length

Based on Fig. 4, it can be seen that the performance of the classifier is the best when we use 8% of the feature space. Therefore, the final proposed model in this paper is a classifier model based on the Persika dataset using NB classifier and MI feature selection while selecting 8% of the features. We call this model *PNC* (Persika-based Persian News Classifier).

Table 7 compares the results of the present study with the performance of Naïve Bayes algorithms in the study conducted by Eghbalzadeh, Hosseini, Khadivi, & Khodabakhsh (2012) to show the improvement in the performance of the NB classifier for Persian text classification when applying feature selection algorithms [45].

Table 7. Comparison of the two models for Persian text classification

Evaluation metric	NB	NB-MI
Accuracy	65.22	80

Table 8 compares the best result of the proposed model by Eghbalzadeh, Hosseini, Khadivi, & Khodabakhsh (2012) with the best result of the present study to show how the performance of the model is improved.

Table 8. Comparison of the two models for Persian text classification

Evaluation Metric	KNN (K=1)	NB.MI
Accuracy	70.18	80

As shown in Table 8, the proposed model has an accuracy of 80%, which is about 10% higher than the previously proposed Persika-based model for Persian text classification.

3.1.9 Evaluating the Proposed Model

Finally, to evaluate the proposed model, we computed the precision, recall, and f-measure for each class, along with their macro-averaged values and the average accuracy of all categories.

As shown in Table 7, the proposed model can perform well in classifying different subjects (different classes). For instance, the model can classify news in sport class with an f-measure of 95% and the news in the religion class with an average f-measure of 91%.

The experimental results of the final proposed model are shown in Tables 9 and 10 below.

Table 9. Experimental Results of the Proposed Model

Evaluation Metric	Precision	Recall	F-measure
0.95	0.94	0.97	Sports
0.91	0.94	0.88	Religion
0.82	0.82	0.82	Judiciary
0.82	0.75	0.90	Culture
0.72	0.60	0.90	Politics
0.66	0.55	0.83	Science
0.86	0.97	0.62	Hygiene
0.86	0.92	0.80	Economy
0.78	0.81	0.75	History
0.52	0.40	0.75	Social
0.80	0.97	0.68	Education

It can be seen in Table 8 that the proposed model has an average accuracy of 80% and an average f-measure of 80% with a standard deviation of 0.01. The results show that the proposed model can perform well in classifying Persian news articles; therefore, the Persika corpus as the dataset can help to build a classifier model for Persian news articles.

Table 10. Evaluation of the Proposed Model

Evaluation Metric	Average	Standard Deviation
accuracy	0.80	0.01
precision	0.81	0.01
recall	0.80	0.01
f-measure	0.80	0.01

4. Conclusion

In this study, the main aim was to suggest a classifier model based on the Persica dataset using Naïve Bayes and K-Nearest Neighbor classifiers to see the performance of these classifiers while applying two feature selection algorithms, MI and CHI. Also, the impact of feature space length on the performance of the model was evaluated to see the best length of feature space.

The results of the present study show that using Naïve Bayes classifier alongside the MI feature selection method can give the best precision, recall, f-measure, and accuracy among the evaluated methods. It is also concluded that using 8% of the feature space can result in the best precision, recall, f-measure, and accuracy. Our empirical results also show that the proposed classifier model can automatically classify Persian news articles with the average f-measure of 80% and the average accuracy of 80%.

References

[1]V. K. Vijayan, K. R. Bindu, and L. Parameswaran, “A comprehensive study of text classification algorithms,” in *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*,

2017, vol. 2017.

[2] J. Novoviĉová, A. Malík, and P. Pudil, “Feature selection using improved mutual information for text classification,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3138, 2004, doi: 10.1007/978-3-540-27868-9_111.

[3] Bahasine, S., et al., Feature selection using an improved Chi-square for Arabic text classification. *Journal of King Saud University-Computer and Information Sciences*,. Vol.32, No.2:pp. 225-231, 2020.

[4] J. R. Vergara and P. A. Estévez, “A review of feature selection methods based on mutual information,” *Neural Computing and Applications*, vol. 24, no. 1. 2014, doi: 10.1007/s00521-013-1368-0.

[5] F. Sebastiani, "Machine learning in automated text categorization". *ACM computing surveys (CSUR)*, 2002. 34(1): p. 1-47.

[6] I. Moulinier and J.-G. Ganascia, “Applying an existing machine learning algorithm to text categorization,” 1996.

[7] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, “Natural language processing: An introduction,” *Journal of the American Medical Informatics Association*, vol. 18, no. 5. 2011, doi: 10.1136/amiajnl-2011-000464.

[5] M. K. Dalal and M. A. Zaveri, “Automatic Text Classification: A Technical Review,” *Int. J. Comput. Appl.*, vol. 28, no. 2, 2011, doi: 10.5120/3358-4633.

[6] B. S. Harish, D. S. Guru, and S. Manjunath, “Representation and classification of text documents: A brief review,” *IJCA, Spec. Issue Recent Trends Image Process. Pattern Recognit.*, no. 2, 2010.

[7] Mahinovs, A., et al., Text classification method review. 2007.

[8] R. Jindal, R. Malhotra, and A. Jain, “Techniques for text classification: Literature review and current trends,” *Webology*, vol. 12, no. 2, 2015.

[9] A. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” *AAAI/ICML-98 Work. Learn. Text Categ.*, 1998, doi: 10.1.1.46.1529.

[10] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-Based Learning Algorithms,” *Mach. Learn.*, vol. 6, no. 1, 1991, doi: 10.1023/A:1022689900470.

[11] J.R., Quinlan, *C4. 5: programs for machine learning*. 2014: Elsevier.

[12] C. Cortes, and V. Vapnik, *Support vector machine.* Machine learning, 1995. 20(3): p. 273-297.

[13] M.E. Ruiz, and P. Srinivasan, "Automatic text categorization using neural networks." i*n Proceedings of the 8th ASIS SIG/CR Workshop on Classification Research. 1998.*

[14] P. Domingos and M. Pazzani, “On the Optimality of the Simple Bayesian Classifier under Zero-One Loss,” *Mach. Learn.*, vol. 29, no. 2–3, 1997, doi: 10.1023/a:1007413511361.

[15] J. H. Friedman, (1997). “On bias, variance, 0/1-loss, and the curse-of-dimensionality,” Data Min. Knowl. Discov., vol. 1, no. 1, doi: 10.1023/A:1009778005914.

[16] S. Gil-Begue, C. Bielza, and P. Larrañaga, (2021). "Multi-dimensional Bayesian network classifiers: A survey". Artificial Intelligence Review, Vol. 54, no. 1, (PP. 519-559). doi: 10.1007/s10462-020-09858-x

[17] N. Friedman, D. Geiger, and M. Goldszmidt (1997). Bayesian network classifiers. Machine learning, vol. 29, no. 2, (PP. 131-163). Springer. doi: https://doi.org/10.1023/A:1007465528199

[18] G. Singh, B. Kumar, L. Gaur, and A.Tyagi, "Comparison between multinomial and Bernoulli naïve Bayes for text classification". *In 2019 International Conference on Automation, Computational and Technology Management (ICACTM)* (pp. 593-596). IEEE.

[19] L. Jiang, C. Li, S. Wang, and L. Zhang, “Deep feature weighting for naive Bayes and its application to text classification,” *Eng. Appl. Artif. Intell.*, vol. 52, 2016, doi: 10.1016/j.engappai.2016.02.002.

[20] X. Zhu, Y. J. Ko, S. Berry, K. Shah, E. Lee, and K. Chan, “A Bayesian network meta-analysis on second-line systemic therapy in advanced gastric cancer,” *Gastric Cancer*, vol. 20, no. 4, 2017, doi: 10.1007/s10120-016-0656-7.

[21] J. Li, X. Y. Tong, L. Da Zhu, and H. Y. Zhang, “A Machine Learning Method for Drug Combination Prediction,” *Front. Genet.*, vol. 11, 2020, doi: 10.3389/fgene.2020.01000.

[22] S. Paudel, P. W. C. Prasad, and A. Alsadoon, *Feature Selection Approach for Twitter Sentiment Analysis and Text Classification Based on Chi-Square and Naïve Bayes*, vol. 842, no. 1. 2018.

[23] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, O. Rusli, and Rudy, “News Article Text Classification in Indonesian Language,” in *Procedia Computer Science*, vol. 116, 2017, doi: 10.1016/j.procs. 2017.10.039.

[24] L. Zhang, L. Jiang, C. Li, and G. Kong, “Two feature weighting approaches for naive Bayes text classifiers,” *Knowledge-Based Syst.*, vol. 100, 2016, doi: 10.1016/j.knosys.2016.02.017.

[25] D. Tomar and S. Agarwal, “A survey on data mining approaches for healthcare,” *Int. J. Bio-Science Bio-Technology*, vol. 5, no. 5, 2013, doi: 10.14257/ijbsbt.2013.5.5.25.

[26] J. O. Pedersen and Y. Yang, “A Comparative Study on Feature Selection in Text Categorization,” *Proceeding ICML ’97 Proc. Fourteenth Int. Conf. Mach. Learn.*, 1997, doi: 10.1093/bioinformatics/bth267.

[27]Y. Yang and X. Liu, “A re-examination of text categorization methods,” 1999, doi: 10.1145/312624. 312647.

[28] S. Tan, “Neighbor-weighted K-nearest neighbor for unbalanced text corpus,” *Expert Syst. Appl.*, vol. 28, no.

4, 2005, doi: 10.1016/j.eswa.2004.12.023.

[29] M. Farhoodi and A. Yari, “Applying machine learning algorithms for automatic Persian text classification,” 2010.

[30] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1998, vol. 1398, doi: 10.1007/s13928716.

[31] L. S. Larkey, “Automatic essay grading using text categorization techniques,” *SIGIR Forum (ACM Spec. Interes. Gr. Inf. Retrieval)*, 1998, doi: 10.1145/290941. 290965.

[32] L. S. Larkey, “Patent search and classification system,” 1999, doi: 10.1145/313238.313304.

[33] W. Lam, M. Ruiz, and P. Srinivasan, “Automatic text categorization and its application to text retrieval,” *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 6, 1999, doi: 10.1109/69.824599.

[34] Y. Zhou, Y. Li, and S. Xia, “An improved KNN text classification algorithm based on clustering,” *J. Comput.*, vol. 4, no. 3, 2009, doi: 10.4304/jcp.4.3.230-237.

[35] Y. Bao and N. Ishii, “Combining multiple k-nearest neighbor classifiers for text classification by reducts,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002, vol. 2534, doi: 10.1007/3-540-36182-0_34.

[36] P. Soucy and G. W. Mineau, “A simple KNN algorithm for text categorization,” 2001, doi: 10.1109/icdm.2001.989592.

[37] L. Esmaeili, M. K. Akbari, V. Amiry, and S. Sharifian, “Distributed classification of Persian News (Case study: Hamshahri News dataset),” 2013, doi: 10.1109/ICCKE.2013.6682829.

[38] M. T. Pilevar, H. Feili, and M. Soltani, “Classification of Persian textual documents using learning vector quantization,” 2009,

[39] N. Maghsoodi and M. M. Homayounpour, “Using thesaurus to improve multiclass text classification,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6609 LNCS, no. PART 2, doi: 10.1007/978-3-642-19437-5_20.

[40] M. H. Elahimanesh, B. Minaei-Bidgoli, and H. Malekinezhad, “Improving K-nearest neighbor efficacy for farsitext classification,” 2012.

[41] M. Parchami, B. Akhtar, and M. Dezfoulian, “Persian text classification based on K-NN using wordnet,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7345 LNAI, doi: 10.1007/978-3-642-31087-4_30.

[42] A.Bagheri, M. Saraee, and S. Nadi, "PSA: a hybrid feature selection approach for Persian text classification", Journal of Computing and Security, 2014. 1(4): p. 261-272.

[43] P. Ahmadi, M. Tabandeh, and I. Gholampour, “Persian text classification based on topic models,” 2016, doi: 10.1109/IranianCEE.2016.7585495.

[44] M. B. Dastgheib and S. Koleini, “Persian text classification enhancement by latent semantic space,” *Int. J. Inf. Sci. Manag.*, vol. 17, no. 1, 2019.

[45] H. Eghbalzadeh, B. Hosseini, S. Khadivi, and A. Khodabakhsh, “Persica: A Persian corpus for multi-purpose text mining and natural language processing,” 2012, doi: 10.1109/ISTEL.2012.6483172.

[46] H. Almagrabi, "Predicting the Helpfulness of Product Reviews: a Sentence Classification Approach", 2020, The University of Manchester (United Kingdom).

[47] S. Yadav and S. Shukla, “Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification,” 2016, doi: 10.1109/IACC.2016.25.

[48] H. K. Kim and M. Kim, “Model-induced term-weighting schemes for text classification,” *Appl. Intell.*, vol. 45, no. 1, 2016, doi: 10.1007/s10489-015-0745-z.

[49] T. Wang, L. Liu, N. Liu, H. Zhang, L. Zhang, and S. Feng, “A multi-label text classification method via dynamic semantic representation model and deep neural network,” *Appl. Intell.*, vol. 50, no. 8, 2020, doi: 10.1007/s10489-020-01680-w.

[50] Y. Li, D. F. Hsu, and S. M. Chung, “Combination of multiple feature selection methods for text categorization by using combinatorial fusion analysis and rank-score characteristic,” *International Journal on Artificial Intelligence Tools*, vol. 22, no. 2. 2013, doi: 10.1142/S0218213013500012.

[51] D. Agnihotri, K. Verma, and P. Tripathi, “An automatic classification of text documents based on correlative association of words,” *J. Intell. Inf. Syst.*, vol. 50, no. 3, 2018, doi: 10.1007/s10844-017-0482-3.

[52] G. Kou, P. Yang, Y. Peng, F. Xiao, Y. Chen, and F. E. Alsaadi, “Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods,” *Appl. Soft Comput. J.*, vol. 86, 2020, doi: 10.1016/j.asoc.2019.105836.

[53] J. Chen, H. Huang, S. Tian, and Y. Qu, “Feature selection for text classification with Naïve Bayes,” *Expert Syst. Appl.*, vol. 36, no. 3 PART 1, 2009, doi: 10.1016/j.eswa.2008.06.054.

[54] H. Liu and R. Setiono, “Chi2: feature selection and discretization of numeric attributes,” 1995, doi: 10.1109/tai.1995.479783.

[55] S. Lee, J. Song, and Y. Kim, “An empirical comparison of four text mining methods,” *J. Comput. Inf. Syst.*, vol. 51, no. 1, 2010, doi: 10.1080/08874417.2010.11645444.

[56] J. C. Lamirel, P. Cuxac, A. S. Chivukula, and K. Hajlaoui, “Optimizing text classification through efficient feature selection based on quality metric,” *J.*

Intell. Inf. Syst., vol. 45, no. 3, 2014, doi: 10.1007/s10844-014-0317-4.

[57] J. He, A. H. Tan, and C. L. Tan, “On machine learning methods for Chinese document categorization,” *Appl. Intell.*, vol. 18, no. 3, 2003, doi: 10.1023/A:1023202221875.

[58] J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review,” in *Data Classification: Algorithms and Applications*, 2014.

[59] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, “Inductive learning algorithms and representations for text categorization,” 1998, doi: 10.1145/288627. 288651.

[60] D. D. Lewis and M. Ringuette, “A comparison of two learning algorithms for text categorization,” in *Proceedings of SDAIR94 3rd Annual Symposium on Document Analysis and Information Retrieval*, vol. 33, 1994.