WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Empirical Revelations through Statistical Techniques in Multiclass Text Classification

Hira Arooj [a], Khawar Iqbal Malik [b*]

[a] *Department of Mathematics & Statistics University of Lahore, Sargodha Campus, Sargodha, 40100, Pakistan*
[b] *Riphah School of Computing and Innovation, Riphah International University Lahore Campus, Lahore, 54000, Pakistan*

ABSTRACT

The practice of Text Mining entails the systematic categorization of textual data, offering a means to extract valuable insights from a diverse array of documents. This method, employed for the establishment of taxonomies, leverages the capabilities of "Text Mining" to discern the primary topic or theme encapsulated within a document collection. Given the rich repository of information inherent in textual data, the application of text mining emerges as a potent avenue for the retrieval and discovery of knowledge from myriad informational sources, thereby boasting substantial commercial value. Central to this endeavor is the process of Text Classification, an instrumental mechanism that systematically organizes documents based on predefined categories. This paper endeavors to furnish a comprehensive introduction to the realm of text classification, delineating its intricate processes, and undertaking a comparative analysis with established classifiers such as Naïve Bayes, Logistic Regression, and Support Vector Machines (SVM). The evaluation criteria encompass pivotal aspects including groupware detection, source retrieval, and overall performance accuracy. This empirical investigation seeks to contribute valuable insights into the efficacy of various text classification methodologies, thereby enriching our understanding of their applicability and performance across diverse criteria.

Keywords: Text classification, Statistical techniques, Feature selection, Logistic regression, Naive Bayes, SVM.

1. INTRODUCTION

In recent times, the significance of text mining research has experienced a notable upswing, attributed to the proliferation of electronic documents sourced from diverse origins, encapsulating both unstructured and semi-structured information. The primary objective of text mining is the extraction of valuable information from these textual resources, encompassing operations such as retrieval, classification (including supervised, unsupervised, and semi-supervised approaches), and summarization. This intricate process involves the collaborative integration of Natural Language Processing (NLP), Data Mining, and Machine Learning techniques. These amalgamated methodologies function harmoniously to automate the classification and pattern discovery processes across various document types (Sebastiani & Fabrizio, 2005). The discernment between supervised and unsupervised categorization algorithms is visually represented in Figure 1. This integration of diverse methodologies facilitates a comprehensive exploration of text mining's capabilities in handling the evolving landscape of electronic documents.

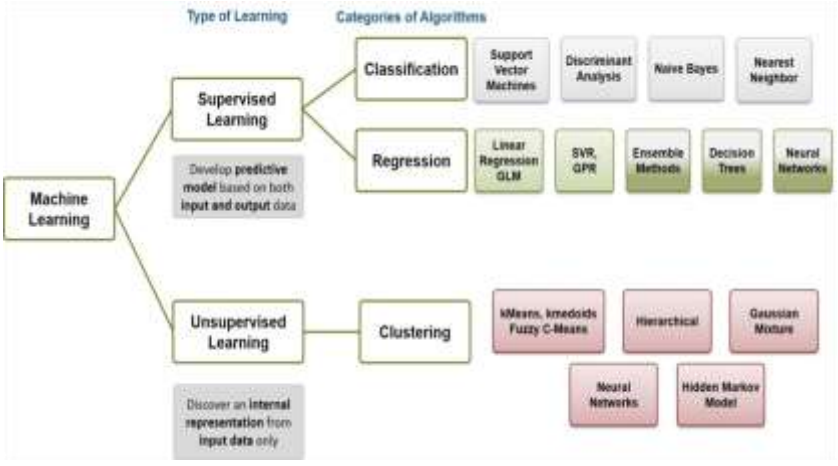


Fig. 1- Types of Classifiers

A pivotal aspect within the domain of text mining involves text classification, wherein automatic systems are constructed, traditionally through manual application of knowledge-engineering techniques. This manual approach entails the formulation of a set of logical rules that articulate expert knowledge on the categorization of documents within predefined categories. For instance, an incoming news story may be automatically labeled with a designated topic such as "sports," "politics," or "art" (Qutab, Malik, & Arooj, Sentiment analysis for Roman Urdu text over social media, a comparative study, 2020). The initiation of a data mining classification task commences with a training set, denoted as D = (d1...dn), comprising pre-labeled documents associated with specific classes (e.g., sport, politics).

The subsequent step involves determining a classification model capable of accurately assigning the appropriate class to a new document within the text domain. Notably, text classification encompasses two facets: single-label and multi-label. A single-label document pertains to only one class, while a multi-label document may belong to more than one class. This paper exclusively focuses on the classification of single-label documents. By concentrating on this specific aspect, the research aims to delve deeper into the nuances of classifying documents associated with a singular label, thus contributing to a more nuanced understanding of text classification processes.

Fig.2- Text classification process has the following stages

2. LITERATURE REVIEW

The study by (Gürcan, 2018) the issue of text classification, which involves the supervised assignment of text documents to predefined categories using natural language processing methods. The focus is on classifying Turkish texts through supervised machine learning methods. The research evaluates the classification performance of various models—Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Support Vector Machine, K-Nearest Neighbor, and Decision Trees—on Turkish news texts. The classification success is examined under different parameters, including the number of training documents.

The analysis specifically targets the classification of news texts into five predefined categories: economy, politics, sport, health, and technology. The study reveals that the Multinomial Naïve Bayes algorithm achieved the highest classification success, reaching approximately 90%. This outcome suggests the effectiveness of the Naïve Bayes probability model as a robust classifier for Turkish texts compared to other methods (Malik, 2022).

In conclusion, the study proposes that the outlined methodology, particularly the Multinomial Naïve Bayes algorithm, can be applied to classify Turkish texts across diverse web platforms, such as social networks, forums, and communication networks. This suggests potential applications for the proposed approach in various contexts, demonstrating its adaptability and effectiveness in classifying Turkish texts for different purposes.

The paper presented by (Raza, et al., 2019)the challenge of selecting the best Software as a Service (SaaS) provider in the context of the widespread adoption of SaaS as a software delivery model. With the increasing distribution of SaaS services due to the growth of edge computing, the importance of making informed choices in provider selection is emphasized. Although major cloud service providers have introduced frameworks and service quality pillars for cloud applications, there is a lack of mechanisms for users to assess how well a service aligns with these defined pillars.

To bridge this gap, the paper proposes a systematic approach involving the analysis of customer reviews associated with SaaS products. The goal is to determine the extent to which a service satisfies the defined service quality pillars. The study employs eleven traditional machine learning classification approaches and a weighted voting ensemble of these classifiers to analyze customer reviews. The evaluation includes 10-fold cross-validation on the training dataset to optimize parameters for each machine learning algorithm, considering the unbalanced sample distribution per class.

The paper applied Friedman test and Nemenyi's post hoc test to identify significant differences in classifier performance during cross-validation. The experimental results are used to conduct a comparative analysis, revealing that the logistic regression model outperforms other individual classifiers. Furthermore, the weighted voting ensemble shows marginal improvement in overall performance. In summary, the study highlights the potential of leveraging machine learning techniques on customer reviews to assess and compare SaaS providers based on their alignment with defined service quality pillars.

The paper addressed by (Casola, et al., 2018) the current limitation in cloud computing, where Service Level Agreements (SLAs) tailored to specific Cloud Service Customers (CSCs) are considered challenging to implement, particularly in terms of security guarantees. Existing cloud SLAs tend to offer uniform guarantees across all services and customers, lacking customization based on specific service characteristics or individual customer needs. This paper introduces a framework designed to overcome this challenge, facilitating the adoption of a per-service SLA model.

The proposed framework enables the automatic implementation of cloud security SLAs customized to the requirements of each customer for specific service instances. The paper outlines the process and software architecture for the implementation of per-service SLAs. To illustrate the feasibility and

effectiveness of the solution, a case study application is presented. This case study involves the provisioning of a secure web container service, demonstrating how the proposed framework can be applied in a real-world scenario.

In summary, the paper contributes to addressing the current gap in cloud computing by introducing a framework that supports the implementation of per-service SLAs, specifically focusing on security terms tailored to individual customer needs. The case study provides practical evidence of the proposed solution's feasibility and effectiveness in ensuring tailored security guarantees for specific cloud services.

K-nearest neighbor (k-NN) classification is a non-parametric classifier commonly used in pattern classification problems, relying on distance measurements between test and training data for classification. This study explores the impact of different distance functions on k-NN performance, particularly in diverse medical domain problems. The investigation focuses on various medical datasets, including categorical, numerical, and mixed types of data. Four distance functions—Euclidean, cosine, Chi-square, and Minkowsky—are individually tested during k-NN classification.

The experimental results indicate that the choice of the distance function significantly influences k-NN performance. The Chi-square distance function proves to be the most effective across all three types of datasets. However, for mixed-type datasets, the cosine and Euclidean (and Minkowsky) distance functions exhibit comparatively lower performance.

In summary, the study demonstrates the impact of distance function selection on the classification accuracy of the k-NN classifier, emphasizing its relevance in medical domain datasets with diverse data types. Notably, the Chi-square distance function emerges as the optimal choice for datasets containing categorical, numerical, and mixed data types (Hu, et al., 2016).

Amidst the rapid expansion of cloud services, choosing the right service from a plethora of functionally similar options poses a significant challenge. The non-functional quality of services plays a pivotal role in service selection and user satisfaction within cloud computing. This study addresses the complexities of selecting a suitable cloud service under unpredictable conditions by introducing a computational framework. The framework integrates the Analytical Hierarchical Process (AHP) and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS).

The study utilizes AHP to establish the architecture for the cloud service selection process, determining criteria weights through pairwise comparisons. Subsequently, the TOPSIS method is employed to derive the final ranking of cloud services based on overall performance. A real-time cloud case study is presented to validate the efficacy of the proposed framework and methodology, demonstrating superior performance compared to existing cloud service selection methodologies. Additionally, sensitivity analysis is conducted to affirm the effectiveness and correctness of the proposed methodology.

In summary, this study introduces a computational framework that combines AHP and TOPSIS for determining the most suitable cloud service. The real-time case study and sensitivity analysis provide empirical evidence of the framework's potential and effectiveness in improving cloud service selection outcomes (Kumar, et al., 2018).

In the context of a service-oriented architecture (SOA) like infrastructure with resource constrained hosting platforms, the discovery and retrieval of an ideal service present significant challenges. This paper proposes a straightforward service selection component for the Internet of Services (IoS). The component is designed to choose services with the highest utility value, considering both Quality of Services (QoS) and Quality of User Experience (QoUE) attributes.

The novelty of the proposed middleware lies in its ability to achieve high configurability and customizability by selecting a subset of middleware functionality based on specific needs. In contrast to conventional service selection approaches, which often require users to provide a set of QoS constraints, this middleware considers QoUE and QoS holistically. The selection approach is grounded in the Multi-Attribute Utility Theory (MAUT) decision-making method.

Experimental results demonstrate the encouraging performance of the middleware, particularly when compared to constraint models and linear programming models. The proposed approach addresses the limitations of conventional methods by incorporating both QoUE and QoS in service selection, contributing to improved decision-making and adaptability in resource-constrained hosting platforms within an SOA-like infrastructure (Balakrishnan, Murugan, Sangaiah, & Kumar, 2017).

In the context of the growing deployment of Web Services (WS) over the Internet, the multitude of available services poses a challenge in selecting the most suitable one for a desired task. Various WS selection approaches have emerged in the past decade to address this issue. This paper introduces a novel WS selection framework that stands out by offering users a flexible and effective means to express their preferences using linguistic terms. It goes beyond the conventional focus on user-specified preferences and enhances WS selection by incorporating user contexts and profiles.

The framework evaluates candidate WS based on an objective score, considering not only the user-specified preferences but also additional preferences derived from the user's context and profile using fuzzy inference rules. This approach aims to improve the overall effectiveness of the selection process. The paper also presents a strategy for prioritizing between the two types of preferences to rank candidate services.

Experimental evaluation conducted on a real case study validates the effectiveness of the proposed strategy. The results demonstrate the framework's ability to provide an enhanced WS selection process by considering user preferences expressed in linguistic terms, as well as incorporating contextual and profile-based preferences, offering a valuable contribution to the field of WS selection in the era of evolving web technologies and increased internet usage (Chouiref, et al., 2016).

Cloud computing operates on a subscription-based computing model, offering a multitude of services from various cloud providers with different qualities. Users, having diverse applications, face challenges in selecting the most suitable service for their needs. The method of comparing services and making

the best selection is considered a complex task. This paper introduces the NSGA_SR approach, which addresses this challenge by incorporating both objective and subjective assessments. The approach models the ranking problem as a multi-objective optimization and employs a non-dominated sorting genetic algorithm to solve it (Qutab, Malik, & Arooj, 2022).

Numerical experiments demonstrate that the NSGA_SR approach surpasses existing methods in terms of flexibility and scalability, particularly as the number of users and services increases. The proposed approach proves effective in optimizing goals, maintaining stability across different generations, and accommodating new quality attributes, services, or supplementary functions without limitations.

In summary, this paper presented an innovative approach, NSGA_SR, to address the challenge of selecting the best cloud service in a subscription-based environment. The approach's ability to integrate both objective and subjective assessments, scalability, flexibility, and adaptability to evolving service landscapes positions it as a promising solution for the complex task of service selection in cloud computing environments (Jahani, Arezoo, Khanli, & Mohammad, 2016).

In the current landscape of cloud computing, customers are faced with numerous cloud services offering similar functionalities but at varying prices and performance levels. Evaluating and ranking these services based on a multitude of quality-of-service attributes presents a significant challenge, considering the trade-offs among different functional and non-functional requirements. This paper introduces a novel approach by proposing modified data envelopment analysis (DEA) and modified super-efficiency data envelopment analysis to assess cloud services' efficiencies, taking user preferences into account.

The modified DEA methods are compared based on sensitivity analysis, adaptability to changes in Decision Making Units (DMUs), suitability for supporting decision-making processes, and modeling of uncertainty. This comparative analysis aims to assist customers in selecting a cloud service that aligns with their specific requirements while fostering healthy competition among cloud service providers.

In summary, this paper presents innovative modifications to DEA methods for evaluating cloud services, providing customers with a comprehensive and tailored approach to selecting the most suitable service based on their preferences and requirements. Additionally, the proposed methodology contributes to creating a competitive environment among cloud service providers, ultimately benefitting customers in the dynamic landscape of cloud computing (Jatoth, et al., 2017).

The adoption of cloud computing technology has surged, offering numerous advantages to organizations. However, the diverse landscape of cloud providers necessitates scientific decision tools for firms to judiciously select the most appropriate vendor. Existing studies primarily focus on technological and cost considerations, often overlooking other influential factors such as competitive pressure and managerial skills. This paper introduces a multi-attribute group decision-making (MAGDM) based scientific decision tool, aiming to provide a comprehensive evaluation of cloud computing vendors by considering a broader range of influencing factors.

The proposed approach advocates the consideration of both objective attributes, like cost, and subjective attributes, such as TOE factors (Technology, Organization, and Environment), in the decision-making process for cloud computing services. The paper presents a new subjective/objective integrated MAGDM approach that combines statistical variance (SV), improved techniques for order preference by similarity to an ideal solution (TOPSIS), simple additive weighting (SAW), and Delphi–AHP. This integration determines the weights of both attributes and decision-makers (DMs), incorporating both objective attribute weights and subjective preferences of DMs, accounting for their identity differences. This holistic approach enhances the accuracy and theoretical validity of decision results.

A numerical example is provided to illustrate the practicality and usefulness of the proposed approach, showcasing its effectiveness as a decision-making tool for firms engaging in cloud computing services. This contribution enriches the theory and methodology surrounding the selection of cloud computing vendors and MAGDM analysis, offering a more robust framework for firms in their decision-making processes (Liu, et al., 2016).

In the rapidly evolving global business environment, the significance of Information Communication Technology (ICT) for firm survival is paramount, with its functions continually gaining importance. Cloud computing, as a transformative force in ICT services, has witnessed rapid growth, offering enhanced functionality to an expanding user base. The critical challenge in this scenario is the selection of a suitable cloud service that aligns with business strategies and objectives. This paper presents a hybrid multi-criteria decision-making model for addressing the cloud service selection problem, employing the Balanced Scorecard (BSC), Fuzzy Delphi Method (FDM), and Fuzzy Analytical Hierarchy Process (FAHP).

The BSC framework forms the basis for the hierarchical structure, incorporating four major perspectives financial, customer, internal business processes, and learning and growth. Decision-making criteria and factors are derived for each BSC perspective. The FDM is employed to identify important decision-making factors within each perspective based on the opinions of decision makers. Subsequently, the FAHP is utilized to compare decision-making criteria and factors, determining their relative importance. It also plays a crucial role in selecting the optimal cloud service among alternatives, considering predetermined weights assigned to decision-making criteria and factors.

The integration of BSC and FAHP as a hybrid multi-criteria decision-making technique facilitates the selection of the best cloud service. The findings offer a systematic approach for decision-making processes in cloud service selection, providing valuable guidance to IT department managers or CTOs in performance evaluation and strategic improvement of companies' capabilities (Lee, Sangwon, Seo, & Kwang-Kyu, 2016).

Cloud services come in various forms, either independently or as a combination of two or more services to meet consumer requirements. Different types of cloud service providers, such as direct sellers, resellers, and aggregators, offer services with varying levels of features and quality. Selecting the most

suitable services involves a multi-criteria evaluation considering both qualitative and quantitative factors, making it a complex decision-making process. This paper proposes a fuzzy hybrid multi-criteria decision-making approach to address this complexity, incorporating both types of factors.

The approach involves the use of triangular fuzzy numbers in pairwise comparison matrices within the Fuzzy Analytic Network Process (Fuzzy ANP). Criteria weights are then determined using Fuzzy Technique for Order Preference by Similarity to Ideal Solution (Fuzzy TOPSIS) and Fuzzy Elimination Et Choix Traduisant la Realité (Fuzzy ELECTRE) methods to rank the alternatives. The application of this strategy is demonstrated through the selection of a cloud-based collaboration tool for designers. Additionally, sensitivity analysis is conducted to showcase the robustness of the proposed approach.

In summary, this paper presented by (Subramanian, Thiruselvan, Savarimuthu, & Nickolas, 2016) a fuzzy hybrid multi-criteria decision-making approach to facilitate the selection of cloud services, considering both qualitative and quantitative factors. The application to the selection of a cloud-based collaboration tool illustrates the practicality of the proposed method, and sensitivity analysis provides insights into the robustness of the approach in handling uncertainties.

3. METHODOLOGY

The initial phase of the classification process involves gathering diverse document types, including HTML, .pdf, .doc, and web content. Subsequently, the preprocessing step transforms text documents into a clear word format. Documents primed for the next phase in text classification are characterized by a multitude of features. The standard sequence of steps includes tokenization, wherein a document is treated as a string and partitioned into a list of tokens. Stop words, such as "the," "a," and "and," are then removed due to their frequent occurrence and lack of significance. Following this, stemming is applied (Fisseha & Yonas, 2011), utilizing an algorithm that converts different word forms into a common canonical form, conflating tokens to their root form (e.g., connection to connect, computing to compute).

The indexing step aims to reduce document complexity, transforming the full-text version into a document vector using the vector space model. While this model represents documents as vectors of words, its limitations include high-dimensional representation, loss of correlation with adjacent words, and a loss of semantic relationships among terms. To address these issues, term weighting methods are employed to assign appropriate weights to terms (Gan, et al., 2017). Various methods for improving the representation of documents are presented, such as representing the semantic relationships between terms using ontology, utilizing N-Grams to extract sequences of symbols, and proposing a new representation for web documents using HTML tags.

After preprocessing and indexing, a crucial step in text classification is feature selection, enhancing the scalability, efficiency, and accuracy of a text classifier. Feature selection involves choosing a subset of features from original documents, typically performed by retaining words with the highest score according to a predetermined measure of importance. Numerous feature evaluation metrics, including information gain, term frequency, Chi-square, and mutual information, are utilized. Additionally, a sampling method is presented, randomly sampling features to create a matrix for classification (Jing, et al., 2015).

Three methods for document classification are discussed: unsupervised, supervised, and semi-supervised methods. The task of automatic text classification has seen significant progress, with machine learning approaches such as Bayesian classifiers, Decision Trees, K-nearest neighbor (KNN), Support Vector Machines (SVMs), and Neural Networks being extensively studied (Arooj & Malik, 2022). Evaluations of text classifiers are typically experimental, focusing on efficiency and effectiveness (Rezaeian, Naeim, Novikova, & Galina, 2020). Performance measures, including Precision, fallout, error, and accuracy, are used for evaluating text categorization.

The evaluation of text classifiers involves data mining techniques, including Naïve Bayes, logistic regression, and SVM algorithms, emphasizing feature space creation and applying various feature selection techniques. The accuracy of these algorithms is improved by selecting appropriate features through specific evaluators and search methods. Naïve Bayes is discussed in detail, encompassing priori and conditional probability (Arooj & Malik, A Control Chart Based on Moving Average Model Functioned for Poisson Distribution, 2020), event models, the multinomial model, and the challenges associated with rare categories. Modified Naïve Bayes and approaches for improving its classification performance are also presented. Despite its ease of implementation and computation, Naïve Bayes may exhibit poor performance when features are highly correlated, emphasizing the importance of suitable feature selection metrics (Wickramasinghe, Indika, Kalutarage, & Harsha, 2021).

For a document d" and a class c" the naïve bays will be

$$p(c|d) = \frac{p(d|c)p(c)}{p(d)} \qquad (1)$$

The application of the Support Vector Machine (SVM) method for text classification is introduced by (Joachims & Thorsten, 1998). Unlike other classification methods, SVM requires both positive and negative training sets. These sets are crucial in the quest to identify a decision surface that effectively separates positive and negative data within an n-dimensional space, referred to as the hyper plane. The support vector is identified as the one wherein representative documents are closest to the decision surface.

$$[\frac{1}{n}\sum_{i=1}^{n} max\,(0,1 - y_i(w, x_i - b))] + \lambda|w|^2 \qquad (2)$$

The SVM classifier method stands out due to its notable effectiveness, particularly in enhancing the performance of text classification (Qin, Yu-ping, Wang, & Xiu-kun, 2009). This effectiveness is achieved by combining Hidden Markov Models (HMM) with SVM. In this approach, HMM serves as a feature extractor, generating a new normalized feature vector that is then fed into SVMs. Through this process, trained SVMs demonstrate the capability

to successfully classify unknown texts. Additionally, the integration of Bayes (Lin & Yi, 2002) is employed to reduce the number of features, effectively reducing the dimensionality. In tackling multi-label class classification, SVM proves to be particularly adept (Qin, Yu-ping, Wang, & Xiu-kun, 2009).

Belonging to the generalized linear model category of statistical models, Logistic Regression is designed to predict a discrete outcome from a set of variables, which can be categorical, numerical, continuous, or dichotomous (Pineda, et al., 2015). When applied to high-dimensional data in natural language text, Logistic Regression encounters computational and statistical challenges. In such scenarios, maximum likelihood estimation frequently faces difficulties. To address these issues, we propose a straightforward Bayesian logistic regression approach. This method incorporates a Laplace prior to prevent over fitting and generates sparse predictive models specifically tailored for text data.

Logistic Regression Model is

$h_\circ(x) = g(\theta^T x) \qquad 0 \leq h_\circ(x) \leq 1$ (3)

$g(z) = \frac{1}{1+e^{-z}}$ (4)

$h_\circ(x) = \frac{1}{1+e^{-\theta^T x}}$ (5)

3.1 Scheme of work

We utilize an extensive dataset comprising Stack Overflow questions and tags, which is openly accessible on this Cloud Storage platform.

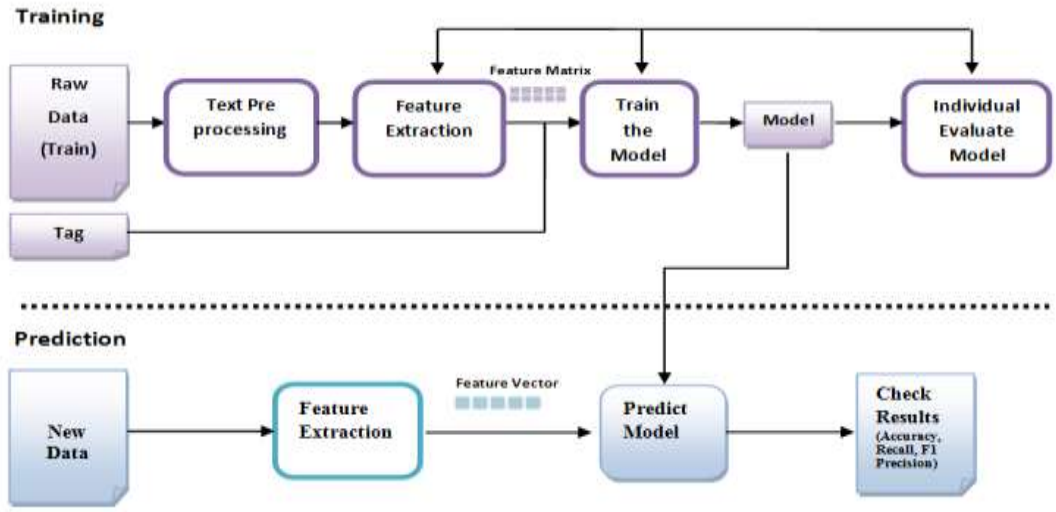


Figure.3- Flow Chart

3.2 Exploring the Data

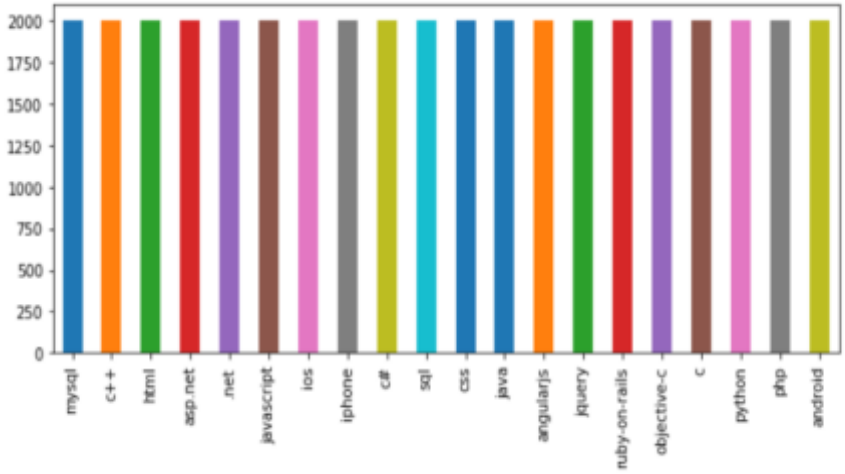
In this Python script, we embark on a comprehensive exploration of a dataset sourced from a CSV file named 'stack-overflow-data.csv.' Employing an array of powerful libraries such as pandas, numpy, gensim, nltk, scikit-learn, matplotlib, and BeautifulSoup, I first read the data into a pandas DataFrame denoted as 'df.' The initial ten rows of the dataset are then displayed, offering a glimpse into its structure. Furthermore, a crucial statistic is unveiled: the cumulative word count across all 'post' entries. This is achieved by applying a lambda function that tokenizes the text and calculates the total word count using the 'split' method. The integration of natural language processing tools, including nltk and BeautifulSoup, along with scikit-learn functionalities, hints at an intricate data preprocessing and feature extraction pipeline, laying the groundwork for subsequent in-depth analyses, potentially within the realms of natural language processing or machine learning.

	post	tags
0	what is causing this behavior in our c# datet...	c#
1	have dynamic html load as if it was in an ifra...	asp.net
2	how to convert a float value in to min:sec i ...	objective-c
3	.net framework 4 redistributable just wonderi...	.net
4	trying to calculate and print the mean and its...	python
5	how to give alias name for my website i have ...	asp.net
6	window.open() returns null in angularjs it wo...	angularjs
7	identifying server timeout quickly in iphone ...	iphone
8	unknown method key error in rails 2.3.8 unit ...	ruby-on-rails
9	from the include how to show and hide the con...	angularjs

Fig 4. Exploring the Dataset

The dataset encompassed nearly 10 million words. Following this, we harmonized the dataset, creating a balanced set that includes all elements observed across all time frames. The achievement of a notable outcome, characterized by low accuracy, high precision, and effective predictions, is contingent on the utilization of a balanced dataset.

Fig. 4- All the classes are well balanced as shown in the above figure.



Following the dataset balancing process, text cleaning techniques are applied, contingent upon the nature of the texts encountered, with the possibility of incorporating more intricate text cleaning steps. In the case of this specific training dataset, the text cleaning phase involves HTML decoding, removal of stop words, conversion of text to lowercase, elimination of punctuation, removal of undesirable characters, and similar processes.

Upon completion of text cleaning and stop word removal, approximately 3 million words remain in the dataset. Subsequently, the dataset is split, and the subsequent steps involve feature engineering. The text documents are transformed into a matrix of token counts using Count Vectorizer, followed by the conversion of a count matrix into a normalized tf-idf representation through tf-idf transformer. Following these transformations, several classifiers are trained using the Scikit-Learn library.

4. PERFORMANCE EVALUATION

4.1 Naive Bayes Classifier for Multinomial Models

Upon obtaining our features, we have the capability to train a classifier for predicting the tag of a post. One effective approach for this task involves using a Naive Bayes classifier, providing a solid baseline. Within scikit-learn, various versions of this classifier are available, with the multinomial variant

accuracy 0.7395

	precision	recall	f1-score	support
java	0.63	0.65	0.64	613
html	0.94	0.86	0.90	620
asp.net	0.87	0.92	0.90	587
c#	0.70	0.77	0.73	586
ruby-on-rails	0.73	0.87	0.79	599
jquery	0.72	0.51	0.60	589
mysql	0.77	0.74	0.75	594
php	0.69	0.89	0.78	610
ios	0.63	0.59	0.61	617
javascript	0.57	0.65	0.61	587
python	0.70	0.58	0.59	611
c	0.79	0.79	0.79	594
css	0.84	0.59	0.69	619
android	0.66	0.84	0.74	574
iphone	0.64	0.83	0.72	584
sql	0.66	0.64	0.65	578
objective-c	0.79	0.77	0.78	591
c++	0.89	0.83	0.86	608
angularjs	0.94	0.89	0.91	638
.net	0.74	0.66	0.70	601
avg / total	0.75	0.74	0.74	12000

Wall time: 955 ms

being particularly suitable for text-related tasks (Awwalu, et al., 2020).

Fig. 5- Precision-Recall-F1 Score for Naïve Bayes Classifier

After the execution of code we receive the above results showed 74% accuracy with 75% precision.

4.2 Linear Support Vector Machine

The optimal algorithm for text classification is the Linear Support Vector Machine, and the results are presented below.

accuracy 0.7891666666666667

	precision	recall	f1-score	support
java	0.74	0.68	0.71	613
html	0.85	0.93	0.89	620
asp.net	0.87	0.95	0.91	587
c#	0.81	0.80	0.80	586
ruby-on-rails	0.74	0.88	0.80	599
jquery	0.77	0.41	0.53	589
mysql	0.82	0.68	0.74	594
php	0.70	0.95	0.81	610
ios	0.82	0.56	0.66	617
javascript	0.72	0.59	0.65	587
python	0.71	0.65	0.68	611
c	0.81	0.87	0.84	594
css	0.77	0.79	0.78	619
android	0.83	0.86	0.85	574
iphone	0.81	0.80	0.81	584
sql	0.71	0.68	0.69	578
objective-c	0.81	0.90	0.85	591
c++	0.84	0.96	0.89	608
angularjs	0.87	0.95	0.91	638
.net	0.77	0.89	0.83	601
avg / total	0.79	0.79	0.78	12000

Wall time: 1.26 s

Fig. 6- Precision-Recall-F1 Score for SVM

We achieve an increased accuracy rate of 79%, marking a 5% improvement over Naive Bayes.

4.3 Logistic Regression

A straightforward and comprehensible classification algorithm is Logistic Regression, which can be easily extended to handle multiple classes. Upon executing the code, the performance accuracy results for Logistic Regression are as follows.

accuracy 0.783

	precision	recall	f1-score	support
java	0.70	0.62	0.66	613
html	0.91	0.91	0.91	620
asp.net	0.97	0.94	0.95	587
c#	0.78	0.77	0.78	586
ruby-on-rails	0.77	0.81	0.79	599
jquery	0.59	0.58	0.58	589
mysql	0.77	0.76	0.76	594
php	0.82	0.86	0.84	610
ios	0.70	0.72	0.71	617
javascript	0.61	0.59	0.60	587
python	0.64	0.63	0.64	611
c	0.83	0.83	0.83	594
css	0.78	0.78	0.78	619
android	0.85	0.85	0.85	574
iphone	0.80	0.83	0.81	584
sql	0.65	0.65	0.65	578
objective-c	0.82	0.84	0.83	591
c++	0.91	0.91	0.91	608
angularjs	0.96	0.94	0.95	638
.net	0.78	0.83	0.80	601
avg / total	0.78	0.78	0.78	12000

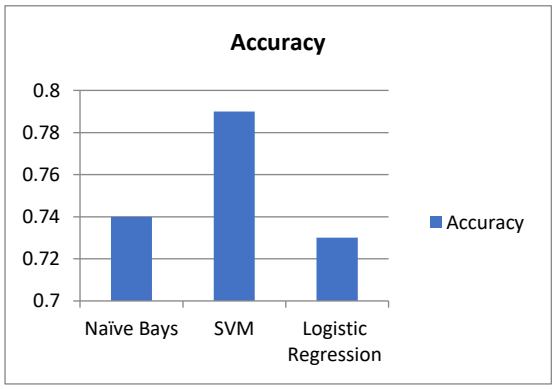
Wall time: 981 ms

Fig. 7- Precision-Recall-F1 Score for Logistic Regression

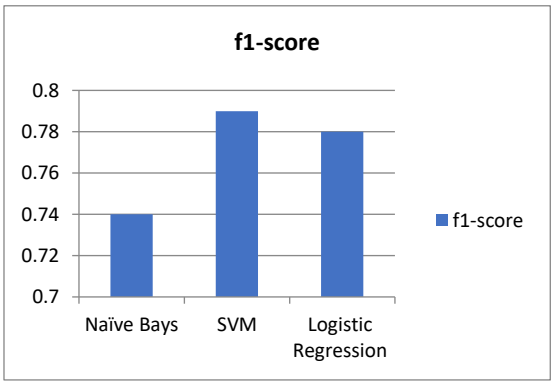
The outcome yields an accuracy score of 78%, indicating a 4% improvement compared to Naive Bayes and a 1% decrease compared to SVM.

Graphical Comparison between Naïve bays, SVM, Logistic regression

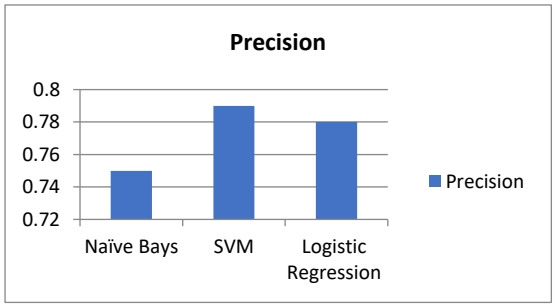
Accuracy Comparison



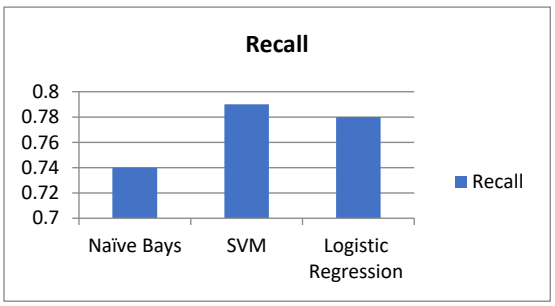
F1-Score Comparison



Precision Comparison



Recall Comparison



The comparative analysis between Naïve Bayes, SVM, and Logistic Regression provides a clear and detailed overview. It reveals that SVM exhibits high peaks in terms of accuracy, precision, recall, and F1 score. However, it is noteworthy that SVM has a longer wall time due to its algorithmic complexity, taking approximately 1.26 seconds for execution. In contrast, Naïve Bayes requires 955 milliseconds, and Logistic Regression takes 981 milliseconds to execute the results.

5. CONCLUSIONS

In this case study, we conducted an analysis of various classifiers using a specific dataset, evaluating their performance across precision, recall, F1 score, accuracy, and wall time parameters. The results indicate that SVM outperforms Logistic Regression and Naïve Bayes classifiers in terms of accuracy, with cumulative precision values of 79%, 74%, and 78%, respectively. The increasing utilization of documented data underscores the growing necessity for text mining, machine learning, and natural language processing techniques to organize, extract patterns, and derive knowledge from documents. This review delves into available literature, examining document representations, and presenting an analysis of feature selection methods and classification algorithms. The discussion highlights that there is no one-size-fits-all representation scheme or classifier applicable across all applications. The performance of different algorithms varies based on the dataset. Nevertheless, it can be concluded to a certain extent that SVM demonstrates superior accuracy compared to the other two classifiers.

References

Arooj, H., & Malik, K. I. (2020). A Control Chart Based on Moving Average Model Functioned for Poisson Distribution. International Journal of Current Science Research and Review, 3(10), 104-112.

Arooj, H., & Malik, K. I. (2022). Double Moving Average Control Chart for Autocorrelated Data. Scientific Inquiry and Review, 6(2), 1-20.

Awwalu, Jamilu, Umar, Aisha, N., Ibrahim, Sani, M., . . . Francisca, O. (2020). A Multinomial Naãive Bayes Decision Support System For Covid-19 Detection. FUDMA Journal of Sciences, 4(2), 704-711.

Balakrishnan, Murugan, S., Sangaiah, & Kumar, A. (2017). Integrated quality of user experience and quality of service approach to service selection in internet of services. International Journal of Grid and Utility Computing, 8(4), 282-298.

Casola, Valentina, Benedicti, De, A., Modic, Jolanda, . . . Umberto. (2018). Per-service security SLAs for cloud security management: model and implementation. International Journal of Grid and Utility Computing, 9(2), 128-138.

Chouiref, Zahira, Belkhir, Abdelkader, Benouaret, Karim, . . . Allel. (2016). A fuzzy framework for efficient user-centric Web service selection. Applied Soft Computing, 41, 51-65.

Fisseha, & Yonas. (2011). Development of Stemming Algorithm for Tigrigna Text. Addis Ababa University: Addis Ababa, Ethiopia.

Gan, Xiaoyu, Fernandez, C, I., Guo, Jie, . . . Jianguo. (2017). When to use what: Methods for weighting and aggregating sustainability indicators. Ecological indicators, 81, 491-502.

Gürcan, F. (2018). Multi-class classification of turkish texts with machine learning algorithms. In 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT (pp. 1-5). IEEE.

Hu, Li-Yu, Huang, Min-Wei, Ke, Shih-Wen, . . . Chih-Fong. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus, 5(1), 1-9.

Jahani, Arezoo, Khanli, & Mohammad, L. (2016). Cloud service ranking as a multi objective optimization problem. The Journal of Supercomputing, 72, 1897-1926.

Jatoth, Chandrashekar, Gangadharan, GR, Fiore, & Ugo. (2017). Evaluating the efficiency of cloud services using modified data envelopment analysis and modified super-efficiency data envelopment analysis. Soft Computing, 21, 7221--7234.

Jing, Liping, Tian, Kuang, Huang, & Z, J. (2015). Stratified feature sampling method for ensemble clustering of high dimensional data. Pattern Recognition, 48(11), 3688-3702.

Joachims, & Thorsten. (1998). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer.

Kumar, Ranjan, R., Mishra, Siba, Kumar, & Chiranjeev. (2018). A novel framework for cloud service evaluation and selection using hybrid MCDM methods. Arabian Journal for Science and Engineering, 43, 7015-7030.

Lee, Sangwon, Seo, & Kwang-Kyu. (2016). A hybrid multi-criteria decision-making model for a cloud service selection problem using BSC, fuzzy Delphi method and fuzzy AHP. Wireless Personal Communications, 86, 57-75.

Lin, & Yi. (2002). Support vector machines and the Bayes rule in classification. Data Mining and Knowledge Discovery, 6, 259-275.

Liu, Sen, Chan, TS, F., Ran, & Wenxue. (2016). Decision making for the selection of cloud vendor: An improved approach under group decision-making with integrated weights and objective/subjective attributes. Expert Systems with Applications, 55, 37-47.

Malik, K. I. (2022). Urdu news content classification using machine learning algorithms. Lahore Garrison University Research Journal of Computer Science and Information Technology, 6(1), 22-31.

Pineda, A. L., Ye, Ye, Visweswaran, Shyam, Cooper, . . . Rich, F. (2015). Comparison of machine learning classifiers for influenza detection from emergency department free-text reports. Journal of biomedical informatics, 58, 60-69.

Qin, Yu-ping, Wang, & Xiu-kun. (2009). Study on multi-label text classification based on SVM. In 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery (pp. 300-304). IEEE.

Qutab, I., Malik, K. I., & Arooj, H. (2020). Sentiment analysis for Roman Urdu text over social media, a comparative study. arXiv preprint arXiv:2010.16408.

Qutab, I., Malik, K. I., & Arooj, H. (2022). Sentiment classification using multinomial logistic regression on roman Urdu text. Int. J. Innov. Sci. Technolgy, 4, 223-335.

Raza, Muhammad, Hussain, Khadeer, F., Hussain, Khadeer, O., . . . Zia. (2019). A comparative analysis of machine learning models for quality pillar assessment of SaaS services by multi-class text classification of users' reviews. Future generation computer systems, 101, 341-371.

Rezaeian, Naeim, Novikova, & Galina. (2020). Persian text classification using naive bayes algorithms and support vector machine algorithm. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 8(1), 178-188.

Sebastiani, & Fabrizio. (2005). Text categorization. Encyclopedia of database technologies and applications, 683-687.

Subramanian, Thiruselvan, Savarimuthu, & Nickolas. (2016). Cloud service evaluation and selection using fuzzy hybrid MCDM approach in marketplace. International Journal of Fuzzy System Applications (IJFSA), 5(2), 118-153.

Wickramasinghe, Indika, Kalutarage, & Harsha. (2021). Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft Computing, 25(3), 2277-2293.