



Performance evaluation of machine learning models on large dataset of android applications reviews

Ali Adil Qureshi¹ · Maqsood Ahmad² · Saleem Ullah¹ · Muhammad Naveed Yasir³ · Furqan Rustam⁴ · Imran Ashraf⁵

Received: 16 March 2022 / Revised: 6 June 2022 / Accepted: 4 February 2023 /

Published online: 18 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

With an ever-increasing number of mobile users, the development of mobile applications (apps) has become a potential market during the past decade. Billions of users download mobile apps for divergent use from Google Play Store, fulfill tasks and leave comments about their experience. Such reviews are replete with a variety of feedback that serves as a guide for the improvement of existing apps and intuition for novel mobile apps. However, application reviews are challenging and very broad to approach. Such reviews, when segregated into different classes guide the user in the selection of suitable apps. This study proposes a framework for analyzing the sentiment of reviews for apps of eight different categories like shopping, sports, casual, etc. A large dataset is scrapped comprising 251661 user reviews with the help of ‘Regular Expression’ and ‘Beautiful Soup’. The framework follows the use of different machine learning models along with the term frequency-inverse document frequency (TF-IDF) for feature extraction. Extensive experiments are performed using preprocessing steps, as well as, the stats feature of app reviews to evaluate the performance of the models. Results indicate that combining the stats feature with TF-IDF shows better performance and the support vector machine obtains the highest accuracy. Experimental results can potentially be used by other researchers to select appropriate models for the analysis of app reviews. In addition, the provided dataset is large, diverse, and balanced with eight categories and 59 app reviews and provides the opportunity to analyze reviews using state-of-the-art approaches.

Keywords Opinion mining · Sentiment analysis · Mobile apps reviews · Google Play Store

1 Introduction

With an ever-increasing number of mobile users, hundreds of thousands of mobile applications (apps) have been developed for Android users. The Google Play Store is currently the biggest online mobile apps store, where more than 2.6 million free and paid applications are

✉ Furqan Rustam
furqan.rustam1@gmail.com

Extended author information available on the last page of the article.

accessible as of 10 January 2022 [5]. On the Google Play Store, users can choose a mobile app for their personal use from different categories and download it to their mobile devices. Mobile app usage gain popularity over the last few years due to its easy-to-use features, simplicity, and mobility. People use these apps for a large variety of daily tasks from health stats, and sports, to business transactions. People share their experiences of using particular apps and provide feedback in the form of reviews and ratings. Application stores such as the Google Play Store and Apple Applications Store give application developers a unique consumer feedback system in the form of application reviews which can be used for different purposes [22]. Reviews often contain suggestions to enhance features, complaints, suggestions to bring more customized options, and innovative ideas. However, analyzing these reviews has several associated challenges. Firstly, app stores include a large number of reviews, which require great effort and a substantial amount of time to examine. Secondly, the caliber of reviews has a wide range of helpful and innovative advice to affront comments. Thirdly, a review tells the different app features that are positive, negative, or neutral, however, sentiment analysis is required for that purpose.

Star rating is a useful numerical value assigned to mobile apps by the users. Since rating represents the average of all the ratings granted by the app users and combines the negative, neutral and positive features in the comments, it reduces the scope of user feedback. Also, technically star rating is not accurate and does not represent a true representation of a mobile app as the user may not be satisfied with an app and still award four stars to the app [6]. Recently, a lot of research has been done to tackle the problem of review anomalies to rank the online reviews based on the sentimental analysis [39, 50]. Existing approaches to sentiment analysis using app reviews lack in several aspects. First, often the data selected for experiments has a small number of samples which limits the scope of the study. Second, predominantly, existing studies utilize reviews related to a single app or a few apps, and mobile app diversity is not covered. Third, with a limited amount of data, the performance evaluation of the models is not exhaustive. This study overcomes these limitations and performs comprehensive experiments in this regard. Keeping in mind the scope and potential of app reviews over app rating, this study leverages the mobile app reviews from Google Play Store and performs sentiment analysis.

This study proposes a framework to perform the sentiment analysis and make the following contributions

- A framework is contrived for analyzing the sentiments of users' reviews of mobile apps on the Google Play Store. The framework utilizes many machine learning models and investigates their performance for the task at hand with the term frequency-inverse document frequency (TF-IDF).
- A large dataset is built that contains the app reviews and comprises 251661 reviews in total for eight different categories of mobile apps. For each category, different apps are selected for review collection. To resolve the problem of dataset imbalance, a similar number of reviews are collected for the dataset.
- Performance of machine learning models is investigated including logistic regression (LR), random forest (RF), multinomial Naive Bayes (MNB), K nearest neighbor (KNN), and support vector machine (SVM) with regards to accuracy, precision, recall, and F1 score. The influence of preprocessing is also analyzed where stats features are also used during this process like the number of words, characters, special characters, etc.

The remainder of this paper is organized into four sections. The literature review is presented in Section 2 followed by the data collection process and proposed methodology in Section 3. Results are presented in Section 4 while Section 5 gives the conclusion.

2 Literature review

With the introduction of mobile apps for Android and iOS, diverse use of such apps has been observed from common daily tasks to business financial transactions. Positive ratings of such apps greatly help the developers which strive to provide more functionality for these apps. Reviews contain suggestions for improvements, criticism of existing problems, and potential solutions to solve these problems, so analyzing the reviews can provide significantly important information. Sentiment analysis has gained large attraction during the past few years and has been utilized to mine opinions from social media text. Similar to other domains, sentiment analysis has been utilized for analyzing mobile app reviews.

Several machine learning algorithms are used to predict the rating of an app on Google Play Store and concluded that the machine learning algorithm like linear regression elegant tree provides the best rating prediction results [6, 33]. The authors investigated how Twitter uses complementary data to drive the creation of mobile apps. A total of 30793 apps were analyzed over six months to see whether there were any links between the number of tweets and program testimonials. Using a machine learning classifier, topic modeling, and subsequent crowdsourcing, the study extracted 22.4% more feature requests and 12.89% another problem report from Twitter [36].

By grouping fine-grained features into more, and relevant features, a refined accuracy for review classification is obtained in [51]. The authors compared the results of peer analysis to 7 apps they downloaded from the Apple and Google app stores. In the proposed method, the app developer can look into users' reviews regarding a specific feature and filter views that are unrelated. The method provides a 91% accuracy score and 73% recall rate. The authors analyze reviews for using two Android apps in [34]. The study use reviews of two recent apps, one from the Brain and Puzzle category and the second from the personalization category. SAS 12.1 is used for performing sentiment analysis on the collected 600 reviews. Performance comparison reveals that the rule-based model is more precise and effective for sentiment analysis. The study [14] asserts that due to a diverse range of apps and related reviews, models being domain or subject dependent, a single model is not suitable to perform sentiment analysis. As a result, different types of classification models are combined to overcome this limitation and benefit from each others' merits, resulting in improved sentiment classification performance.

Authors combined app reviews and ratings in [16] to study the relationship between user-reported cases and app ratings. In addition, important factors with a high influence on apps' perceived quality are studied. The focus is placed on topics that are highly correlated to app rating. The study indicates that users' feedback on apps' bugs has a direct relation to the app's lower rating. The study [54] identifies key features that are associated with apps' high ratings and present a novel approach using app description and user reviews for finding such key features. For this purpose, natural language processing, and machine learning approaches have been leveraged. Results indicate that an average F1 of 78.13% can be achieved with the proposed approach.

Twitter provides complementary information to support mobile app development. By analyzing a total of 30793 apps over six weeks, the authors found strong correlations between the number of reviews and tweets for most apps in [30]. Moreover, through applying machine learning classifiers, topic modeling, and subsequent crowd-sourcing, the author successfully mined 22.4% of additional feature requests and 12.89% additional bug reports from Twitter. The author also found that 52.1% of all feature requests and bug reports were discussed in both tweets and reviews. In addition to finding common and unique information from Twitter and the app store, sentiment and content analyses are also performed

for 70 randomly selected apps. From this, the author found that tweets provided more critical and objective views on apps than reviews from the app store.

The author introduces a scalable system to help analyze and predict Android apps' compliance with privacy requirements in [60]. The proposed system is not only intended for regulators and privacy activists but also meant to assist app publishers and app store owners in their internal assessments of privacy requirement compliance. The authors performed exploratory data analysis on the data collected from the Google Play Store data for feature relationship in [29]. The purpose is to dive deeper into discovering relationships of specific features such as how the number of words in an app name, for instance, affects installs, to use them to find out which apps are more likely to succeed. Using these extracted features and the recent sentiment of users the authors predicted the success of an app soon after it is launched into the Google Play Store.

The authors in [24] show that 23.3% of users request further features for the apps, i.e., comments through which users either suggest new features for an app or express preferences for the re-design of already existing features of an app. One of the challenges app developers face when trying to make use of such feedback is the massive number of available reviews. Through this work, the authors provided a process by designing a mobile app review analyzer (MARA), a prototype for automatic retrieval of mobile app feature requests from online reviews.

In [21], the authors aim to measure the extent of the sentiment analysis results given by customers to Go-Jek through the comments. Customers' opinions are taken to get positive, negative, or neutral comments. Go-Jek is one of the most popular providers of online transportation services in Indonesia that has now grown to become the on-demand mobile platform and the leading application that provides a full range of services ranging from transportation, logistics, payments, and food delivery services, and various other services.

The author examines reviews at the update level to better understand how users perceive bad updates in [22]. The study focuses on the top 250 bad updates (i.e., updates with the highest increase in the percentage of negative reviews relative to the prior updates of the app) from 26726 updates of 2526 top free-to-download apps in Google Play Store. The authors find that feature removal and user interface issues have the highest increase in the percentage of negative reviews. Bad updates with crashes and functional issues are the most likely to be fixed by a later update.

The author applies a conjoint study approach in [38]. The author conducts the research to quantify the monetary value that users place on their friends' personal information. Utilizing the scenario of social app adoption, the authors further investigate the impact of the comprehensiveness of shared profile information on valuation and varying the data collection context, i.e., friends' information is not relevant, or is relevant to app functionality.

3 Proposed methodology

This section describes the methodology used to acquire the dataset, its visualization, the preprocessing techniques applied for dataset selection, feature extraction to implement classification techniques, and the proposed methodology.

Figure 1 shows the methodology adopted for this study. The Google Play store data is scraped by using the regular expression and the raw data of star ratings and comments are acquired for apps from different categories. Secondly, the stats features are implemented on the raw data i.e., counting the number of words, characters, special words, case sensitivity, etc. Thirdly, the data is preprocessed by using the removal of many words such as

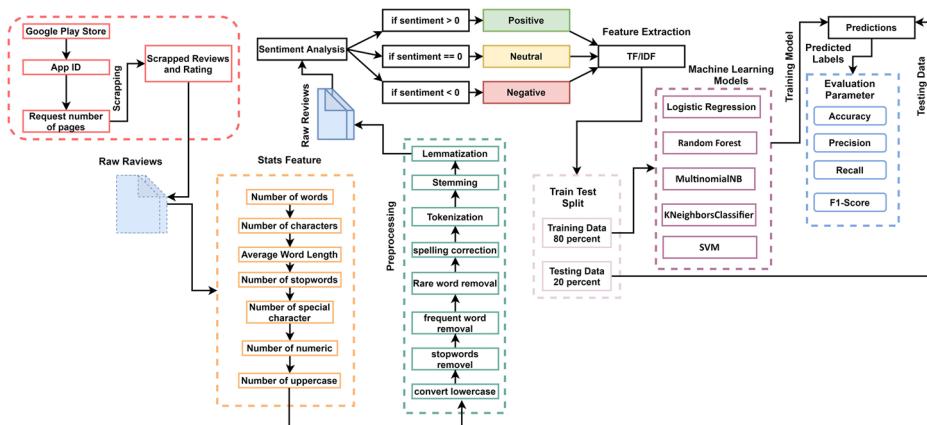


Fig. 1 The flow of Google Play Store applications reviews classification

frequent words, rare word stop words, and conversion of lowercase with correction of misspelled words. Fourthly, the sentimental analysis is performed on the data labeled using the TextBlob where the sentiments are assigned based on sentiment polarity. Fifthly, the TF-IDF features are extracted from the preprocessed data to train the machine learning models used in this study. In the later stage, the features are used with the models for training, testing, and validation with LR, MNB, RF, KNN, and SVM. The results are analyzed based on F1 score, recall, precision, and accuracy.

3.1 Data collection

This study collects a large dataset comprising 234453 user reviews, scraped from the Google Play Store. The scrapping process is depicted in Fig. 2. The ‘request’ library is used to extract the data from the Google Play Store. The ‘request’ is a Python package that allows users to submit HTTP requests. The data were extracted with the labeling like headers files, multipart files, and parameters files using Python libraries. This labeling helped to identify the category of the application. The number of mobile applications followed the category. The reviews and rating data are translated in the form of sentimental analysis (labeled as positive, negative, and neutral). The data is further classified by using machine-learning algorithms.

The regular expression (re) is a character sequence that aids in matching or finding other strings or groupings of strings by employing a specified syntax stored in a pattern. The ‘re’ is essentially a character series that assists in searching for matching patterns in the text [37]. We used the Beautiful Soup for data scrapping [10]. This is a library of python, which is used to extract information from the HTML and XML documents. The review data is divided

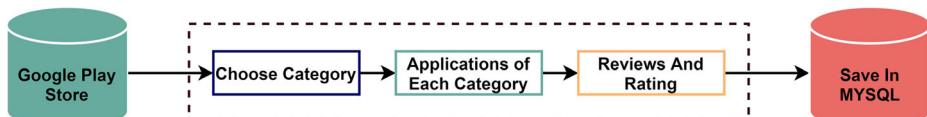


Fig. 2 The scrapping process of Google Play Store applications reviews

into eight different categories depending on their usability for different tasks and include ‘action’, ‘casual’, ‘communication’, etc. Figure 3 represents the name and percentage ratio of each category in the dataset. Each category further has reviews and ratings for different apps.

The detail of the scrapped data into general categories with the number of consumer reviews for each category is given in Table 1. A total of 24000 reviews are gathered for the ‘Action’ category with 4001 reviews each for six mobile apps including ‘Bush Rush’, ‘Metal Soldiers’, ‘Real Gangster Crime’, ‘Talking Tom Gold Run’, etc. The complete details of each category and its comprising apps are provided in Table 1.

3.2 Stats feature using reviews

The user reviews raw data is extracted from the Google Play Store for further processing. The stat feature counts the number of words, characters, average word length, stop words, special characters, numeric and upper case. The objective of using the stats feature for the collected reviews is to analyze the patterns for positive, negative, and neutral reviews. The following steps are performed for stats features. Figure 4 shows the stats feature.

3.2.1 Number of words

Initially, the number of words is counted from each user review. It is observed that the negative users’ reviews have a lesser number of words than the positive reviews.

3.2.2 Number of characters

The number of characters is counted from each review. This process is performed by calculating the length of each review. Similar to the number of words, the number of characters for negative reviews is short.

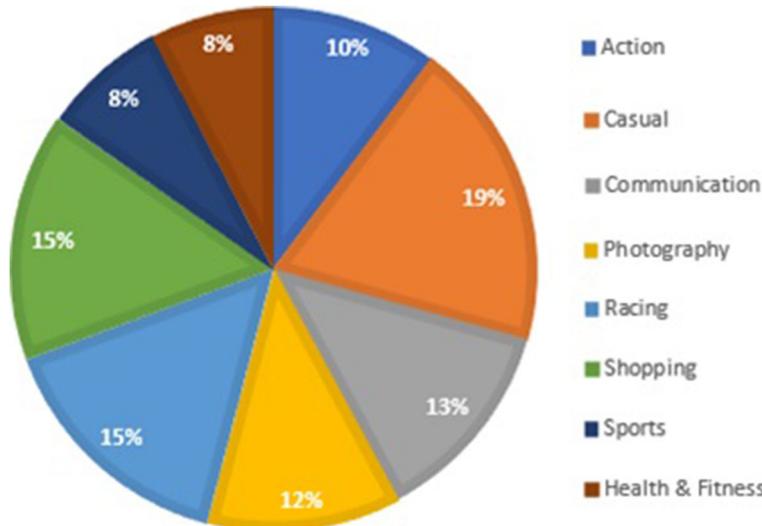
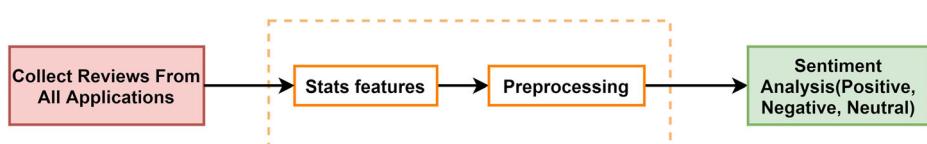


Fig. 3 Percentage of each category in the dataset

Table 1 Detailed dataset information regarding categories and apps

App Name	Reviews	App Name	Reviews
Casual			
Angry Bird POP	4481	Dolphin Brower	3001
Boards King	4481	Fire Fox Brower	3001
Bubble Shooter	4481	Google Duo	3001
Candy Crush Saga	4481	Hangouts Dialer	3001
Farm Heroes Super Saga	4481	KakaoTalk	3001
Hay Day	4481	LINE	3001
Minion Rush	4481	Messenger Lite	3001
My Talking Tom	4481	Opera Mini Brower Beta	3001
Pou	4481	UC Brower Mini	3001
GardenScapes	4481	WeChat	3001
Racing			
Asphalt Nitro	4481	AliExpress	4481
Beach Buggy Racing	4481	Amazon for Tablets	4481
Bike Mayhem Free	4481	Bikroy	4481
Dr. Driving 2	4481	Flipkart Online Shopping App	4481
Extreme Car Driving	4481	Lazada	4481
Racing Fever	4481	Mynta Online Shopping App	4481
Racing in Car 2	4481	Snapdeal Online Shopping App	4481
Trial Xtreme 4	4481	Shopclues	4481
Action			
Bush Rush	4001	FVolleyball Champions 3D	3000
Metal Soldiers	4001	Real Cricket™18	3000
Real Gangster Crime	4001	Real Football	3000
Talking Tom Gold Run	4001	Score! Hero	3000
Warship Battle	4001	Table Tennis 3D	3000
Zombie Frontier 3	4001	Tennis	3000
Action			
B612 - Beauty & Filter Camera	4001	Home Workout - No Equipment	4481
BeautyCam	4001	Lose Belly Fat In 30 Days	4481
BeautyPlus	4001	Lose Weight Fat In 30 Days	4481
HD Camera	4001	Water Drink Reminder	4481
Motorola Camera	4001		
Music Video Maker	4001		
Sweet Snap	4001		

**Fig. 4** Stats feature of Google Play store applications reviews

3.2.3 Average word length

The average word length is calculated at this level for each collected review. The average word length makes the system more smooth and expressive. The sum of words has been calculated from each review and these words are divided by the total length of the review to obtain the average word length.

3.2.4 Number of stop words

The stop word calculation is helpful to get extra information i.e. ‘and’, ‘a’, ‘the’, ‘is’, ‘are’, etc. The stop words are calculated by applying the Natural language toolkit (NLTK) library. The implementation of NLTK is used for the machine learning algorithms [12].

3.2.5 Number of special characters

There are different special characters used in users’ reviews. The number of hashtags is calculated and extracted. Moreover, some more information is extracted from the reviews. Hashtags always appear at the beginning of a word, however, other special characters may appear in the middle, or other places in a sentence like ‘—’, question mark, ‘@’, etc.

3.2.6 Number of numeric values

The number of numerical values is calculated which users mention shortening the length of reviews, for example, ‘4’ is often used instead of ‘for’. Although, numeric values are often removed from the text, using them could provide additional information.

3.2.7 Number of uppercase words

Most of the time the user expresses his feelings in the form of capital words like Anger and anger is often written in UPPERCASE words. So, extracting such information may be helpful to analyze the sentiments.

3.3 Preprocessing steps

Regarding the use of machine learning models for sentiment analysis, the data needs to be cleaned before training the models, as, it helps to increase the training accuracy and improves the performance of machine learning models [43]. Performance is improved as the noise, unnecessary and redundant data, that do not contribute to predicting the target class, are removed [26]. Figure 5 shows the steps followed in the preprocessing.

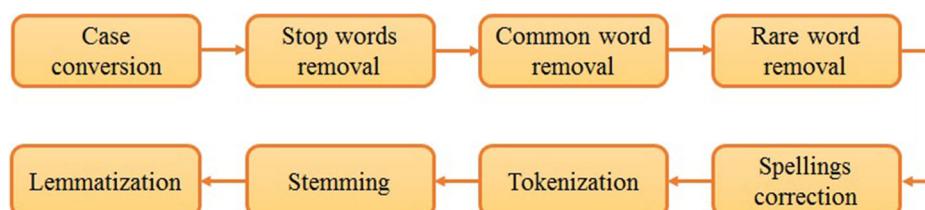


Fig. 5 Preprocessing steps followed in this study

3.3.1 Lower case

In the first step, the dataset is transformed into the lower case. Replication of the same words in the dataset is omitted. For instance, the terms ‘Excellent’ and ‘excellent’ are treated differently when determining the word count.

3.3.2 Removal of stop words

Removal of stop words is essential in Natural Language Processing (NLP) tasks and text analysis [2]. This process can be followed routine using a pre-defined library or using a list of stop words.

3.3.3 Common word removal

In this step, commonly used words are removed from the reviews. A collection of 10 most frequently occurring words were analyzed and removed.

3.3.4 Rare words removal

In this step, the rare words from the user reviews are removed. Due to the scarcity, noise dominates the link between uncommon and other words. Unusual words are replaced with the general word to increase the count of the words.

3.3.5 Spelling correction

Many real-world NLP application problems rely on misspelling detection and repair modules to function properly [28]. Textblob library is used for spelling correction because this step is more potent in reducing the copies of words by preprocessing.

3.3.6 Tokenization

The process of breaking text into a list of tokens that are used as whole words or part of words called subwords is known as tokenization [4]. Tokenization is used for the distribution of the user reviews in the sequence of words or sentences, which are transformed into a blob and then into a string of words by using the ‘textblob’ library.

3.3.7 Stemming

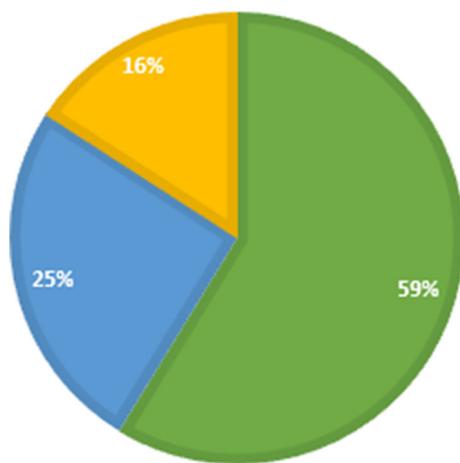
Stemming is a typical need of NLP, as well as a preprocessing step in text mining applications [52]. By using a simple rule-based approach, the words are transformed into their root form, like removing ‘ing’, ‘ly’, ‘s’, and so on. Porter Stemmer is used from the NLTK library for this process.

3.3.8 Lemmatization

The technique of deriving the dictionary form of a word (e.g. swim) given one of its inflected forms is known as lemmatization (e.g. swims, swimming, swam, swum) [28]. Rather than draining the suffices, the phrase is converted to its origin phrase by lemmatization. As a result, lemmatization is the best choice. Before part-of-speech (PoS) tagging, lemmatization

Fig. 6 Sentiments ratio of Google Play store applications reviews

■ Positive ■ Neutral ■ Negative



is required for morphological analysis and the elimination of inflections by returning the base of the word without the ends [18].

3.4 Dataset annotation

The sentiment is assigned to a review with respect to its weight and a different threshold value is used [39]. Textblob library of Python is used for this purpose [1]. Textblob is a Python module for handling textual data. It provides a straightforward application programming interface (API) for language processing tasks such as noun word extraction, tagging, sentiment analysis, and more. TextBlob can be used for classification purposes for large training data sets with many dimensions. Figure 6 shows the distribution of sentiments for the collected dataset. It can be seen that 58% of people have positive feedback, 25% gave neutral comments while only 16% gave negative feedback.

3.5 Feature extraction

The reviews are organized and cleaned by using the preprocessing steps and now can be used for feature extraction. Figure 7 shows the process of feature extraction for sentiment analysis of app reviews. TF-IDF is used to extract features of the specific words from the reviews dataset. TF-IDF is the most commonly used feature for text analysis. Each term in a document is given a weightage depending on its TF and IDF [44].

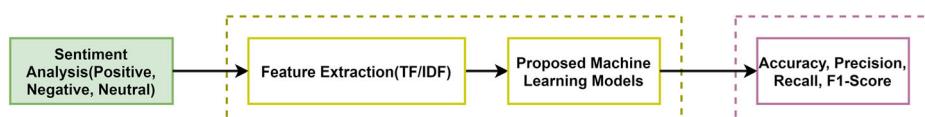


Fig. 7 Feature extraction for sentiment analysis

TF determines the frequency of each unique term in a document and can be used as follows [56]

$$TF(T) = \frac{\text{No. of time } T \text{ appears in a document}}{\text{Total number of terms inside document}} \quad (1)$$

TF-IDF punishes frequent terms and assigns higher weights to those terms that appear less likely in a given corpus. It is calculated as [56]

$$IDF = \log \frac{\text{Total number of documents}}{\text{No. of documents through term } t \text{ in it}} \quad (2)$$

In the end, TF-IDF can be obtained by multiplying TF and IDF.

3.6 Machine learning models for reviews classification

The implementation of the various supervised machine-learning algorithms such as LR, RF, SVM, and KNN is used in this research. The machine learning algorithms are refined for better performance using different hyperparameters and a list of all parameters is provided in Table 2.

3.6.1 Logistic regression

LR is preferred here because LR shows better performance for binary classification and text categorization tasks [20]. Contrary to linear regression which produces continuous numerical values, LR modifies its output by applying the logistic sigmoid function to yield a probability value that may then be translated to two or more discrete groups. For our dataset, the random_state=0 and multi_class='ovr' options are utilized. The following equation is used to find logistic regression [41]

$$Y(x) = \frac{L}{1 + e^{(-n(v-v_0))}} \quad (3)$$

where the natural algorithm basis is e (also known as Euler Number), the sigmoid midpoint's x-value is v_0 , the greatest value of the curve is L , and the steepness of the curve is represented by n .

LR has been used for a variety of tasks including clinical studies [57], finance applications [19], meteorological and ecological data [3], tomography [47] and many other [7, 50].

Table 2 Parameter information of supervised machine learning algorithms

Algorithms	Parameters
LR	random_state=0 , multi_class='ovr'
RF	random_state=1,n_jobs= -1, n_estimators=1000, max_features=4
MNB	Default parameters
KNN	n_neighbors=3
VM	kernel='linear', probability=True, C=1

3.6.2 Random forest

RF is a machine-learning algorithm used for classification. The bagging approach is used in this model to train several decision trees using different bootstrap samples [17]. RF's basic premise is that it is computationally cheap for creating a tiny decision tree with few attributes. The trees merge by merging or taking the majority vote to construct a single, powerful learner if there is the possibility to build multiple tiny, weak decision trees simultaneously. RF is frequently discovered to be the most accurate learning algorithm. For this experiment dataset, random state=1, n_jobs= -1, n_estimators=1000, and max_features=4 parameters are used. The equation for finding RF is given below [45]

$$P = \text{mode}\{T1(y), T2(y), \dots, Tm(y)\} \quad (4)$$

where p is the final prediction by majority voting of decision trees, while $T1(y)$, $T2(y)$, $T3(y)$, and $Tm(y)$ are the number of decision trees participating in the prediction procedure.

RF has been used for several applications related to text analysis like sentiment analysis and text classification [26, 53], COVID-19 pandemic [58], e-commerce [35], and prediction tasks [59].

3.6.3 Multinomial Naive Bayes

MNB is a probabilistic classifier that relies on the Bayes theorem's properties and assumes that the features are highly independent. One of the benefits of this algorithm is that it just needs a small amount of training data to compute the parameters for prediction. Due to the independence of characteristics, the single variance of the characteristic is determined rather than the maximum covariance matrix. For each one, the conditional likelihood for a given textual analysis d and a class c is $P(c|d)$. This likelihood can be determined using the Bayes theorem, according to the following equation [40].

$$P(c|d) = \frac{p(d|c) * p(c)}{p(d)} \quad (5)$$

where $P(c|d)$ is posterior probability, $P(d|c)$ is a likelihood, $P(c)$ is class prior probability, and $P(x)$ is predictor prior probability.

Applications of MNB include text classification [42], apps review [11], sentiment classification [32], and so on.

3.6.4 K nearest neighbor

KNN is a non-parametric supervised learning approach that categorizes data points into a certain category using the training set. It collects information on all training instances and categorizes fresh cases based on their similarity. For our dataset, the n_neighbors=3 option is used with the following distance metric

$$\Pr(Y = j|X = x_0) = \frac{1}{k} \sum_{i \in N_0} I(y_i = j) \quad (6)$$

where N_0 is the set of K-nearest views, and an indicator variable $I(y_i = j)$ that evaluates to 1 if a particular observation (x_i, y_i) is N_0 belongs to a class j , and 0 otherwise.

KNN shows better performance for applications where the feature space is small as it uses all the samples for the training. Despite its simplicity, KNN has been used for several

applications like market prediction [27], medical imaging [9], signal classification [25], and COVID-19 prediction [8, 55].

3.6.5 Support vector machine

SVM is a non-parametric binary linear classifier that relies on a collection of mathematical functions [46]. SVM divides the data into classes by drawing a line or a hyperplane [44]. The SVM algorithm interprets each assessment in vectorized form as a data point in space. The fundamental concept behind this approach is to create a model specified by the hyperplane w , which is utilized to analyze the complete vectorized data. The hyperplane is regarded as ideal when it divides the data with the maximum distance between samples of different classes. For our dataset, kernel='linear', probability=True, and C=1 parameters are used.

$$f(x) = \text{sgn}(w^T x + b) \quad (7)$$

where $w = \text{sum} .ia .ix .iy .i$ is zero for all cases, but the support vectors (those lying exactly at the separating hyperplane), and 1,-1 are the labels.

SVM has been widely used in multifarious domains including bioinformatics [13], text analysis [48], hydrology [15], computational biology [31], financial forecasting [49] and disease prediction [23].

4 Results and discussion

The performance of various algorithms has been examined for the collected dataset. Experiments are performed on a Jupyter notebook and Python is used to implement the machine learning models. Performance is determined with respect to F1 score, precision, recall, and accuracy.

4.1 Accuracy of models

Table 3 shows the results of machine learning models regarding accuracy. It can be observed that accuracy varies both with the model, as well as for different categories of apps. For example, the performance of SVM is the best for all app categories, however, the highest

Table 3 Accuracy results of machine learning algorithms

Classifiers	Action	Casual	Communication	Health and Fitness
LR	0.92	0.92	0.91	0.90
RF	0.93	0.91	0.90	0.89
MNB	0.89	0.87	0.86	0.71
KNN	0.81	0.77	0.68	0.65
SVM	0.94	0.93	0.93	0.92
Classifiers	Sports	Racing	Shopping	Photography
LR	0.91	0.93	0.92	0.93
RF	0.92	0.94	0.92	0.93
MNB	0.71	0.90	0.88	0.89
KNN	0.69	0.81	0.76	0.81
SVM	0.93	0.95	0.93	0.95

accuracy of 0.95 is for the ‘photography’ app category. KNN shows the lowest accuracy of 0.65 with the ‘Health and Fitness’ category, as shown in Fig. 8. MBN also has poor performance with a 0.71 accuracy score for the ‘Sports’, and ‘Health and Fitness’ categories. The performance of models varies as the nature of words used for reviews regarding different categories is different.

Despite variations in the performance of machine learning models, SVM shows consistently better performance for all app categories with an accuracy score higher than 0.92. LR and RF also show better performance with accuracy above 0.90, except for RF accuracy of 0.89 for the ‘Health and Fitness’ category.

4.2 F1 score of machine learning models

The F1 score ranges from 0 to 1 and is regarded more important performance evaluation metric as compared to accuracy, precision, and recall. It considers both precision and recall

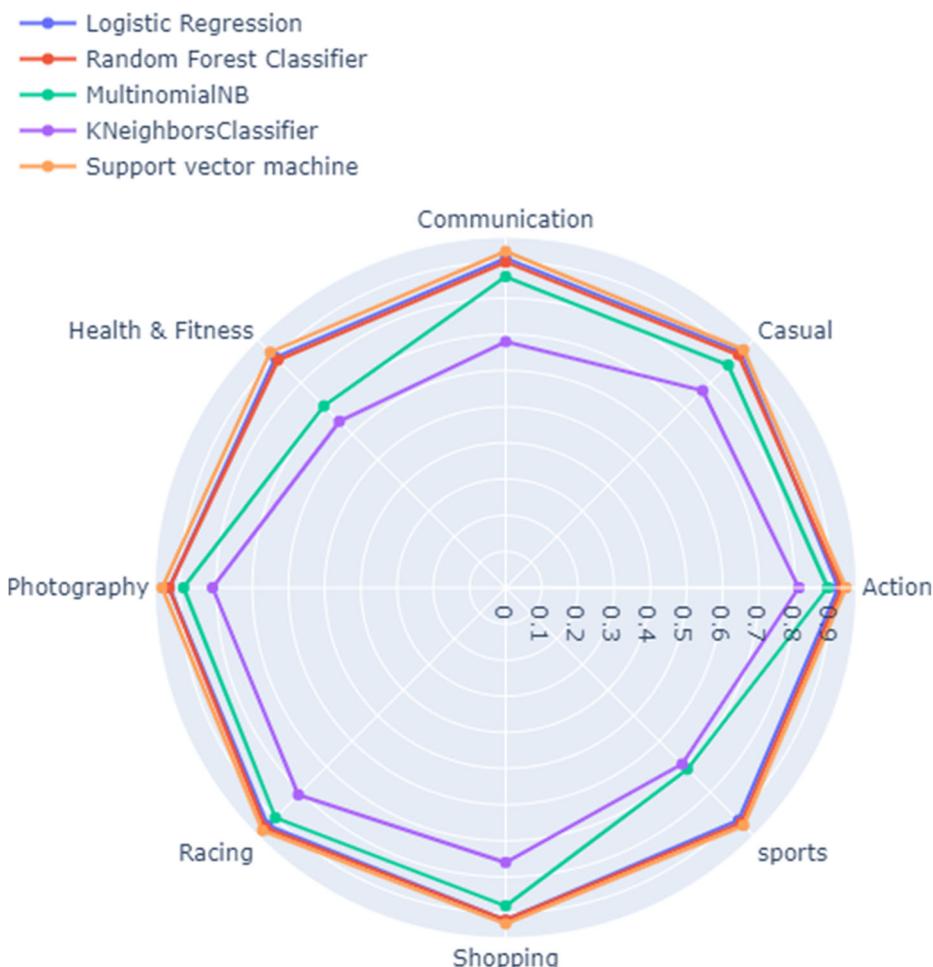


Fig. 8 Graphical visualization of accuracy results

and provides a more balanced analysis of a model, especially when the dataset is imbalanced. Table 4 shows the results for the F1 score of all machine learning models used in this study. Results indicate that the highest F1 score of 0.93 is obtained by SVM for ‘Sports’ and ‘Racing’ app categories. The lowest F1 score is from KNN with 0.54 for the ‘Health and Fitness’ category, similar to its performance regarding accuracy. Figure 9 shows the visual illustration of the F1 score for all models. It indicates that on average the performance of many models is good, except for KNN and MBN which show substantially poor performance. Often the performance of KNN is poor when it comes to the large dataset which is the case for the current study as it contains more than 200000 reviews. KNN shows better performance with small datasets.

4.3 Recall results of machine learning models

The recall is a measure of completeness that indicates the proportion of true positive occurrences of a class identified as such. Table 5 provides the results of each machine learning model for each app’s category regarding review classification. Similar to its performance for accuracy and F1 score, SVM shows superior performance regarding recall score for the review classification with the highest 0.92 scores for ‘Sports’ and ‘Racing’ categories. Figure 10 further illustrates the poor performance of KNN and MNB regarding recall with the lowest recall of 0.50 and 0.55 for MNB and KNN, respectively for the same category.

4.4 Precision of reviews classification

Contrary to recall, precision refers to the proportion of a class’s instances that are accurately classified as positive. Figure 11 shows the graphical illustration of the results of machine learning models regarding the precision while detailed results for every category are provided in Table 6. Results corroborate that SVM outperforms all machine learning models regarding precision, for app review classification with the highest precision of 0.95 for the ‘Racing’ category. The precision score of SVM is higher than 0.91 for all categories except for ‘Health and Fitness’ with a 0.89 precision score.

Table 4 F1 score results of machine learning algorithms

Classifiers	Action	Casual	Communication	Health and Fitness
LR	0.89	0.90	0.88	0.77
RF	0.91	0.89	0.85	0.74
MNB	0.85	0.83	0.80	0.50
KNN	0.74	0.72	0.60	0.54
SVM	0.92	0.91	0.90	0.84
Classifiers	Sports	Racing	Shopping	Photography
LR	0.90	0.91	0.90	0.86
RF	0.90	0.92	0.90	0.85
MNB	0.72	0.86	0.86	0.79
KNN	0.67	0.77	0.70	0.71
SVM	0.93	0.93	0.92	0.91



Fig. 9 Graphical visualization of F1 score results

Table 5 Recall results of machine learning algorithms

Classifiers	Action	Casual	Communication	Health and Fitness
LR	0.86	0.88	0.86	0.74
RF	0.89	0.86	0.83	0.70
MNB	0.81	0.80	0.77	0.50
KNN	0.70	0.70	0.58	0.55
SVM	0.91	0.91	0.89	0.82
Classifiers	Sports	Racing	Shopping	Photography
LR	0.89	0.89	0.88	0.82
RF	0.89	0.90	0.89	0.81
MNB	0.70	0.82	0.84	0.74
KNN	0.66	0.74	0.66	0.67
SVM	0.92	0.92	0.91	0.89

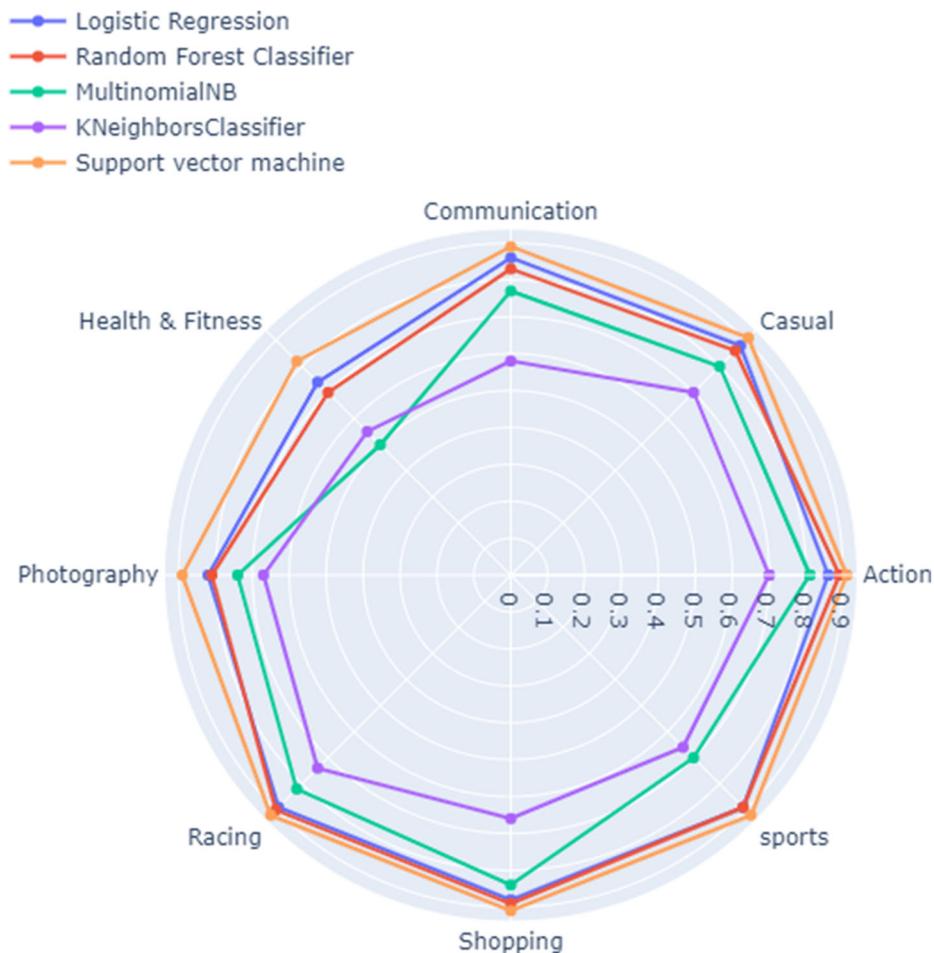


Fig. 10 Graphical visualization of recall results

The performance of SVM is followed by RF with superb results for every category. Actually, on average, the performance of RF is better than SVM except for the fact that the highest precision belongs to SVM. The lowest precision of 0.64 is achieved by the KNN for the ‘Health and Fitness’ category.

4.5 Discussions

This study analyzes the performance of several machine learning models regarding Google App reviews. Reviews analysis has been an important research area for the past few years due to wide use of mobile apps for different tasks like sports, finance, entertainment, fitness, etc. Users are always interested in the mobile apps which best serve their needs without interruptions and bugs. From this perspective, the classification of users’ reviews regarding different apps can guide the users to select an appropriate app. However, such reviews

- Logistic Regression
- Random Forest Classifier
- MultinomialNB
- KNeighborsClassifier
- Support vector machine

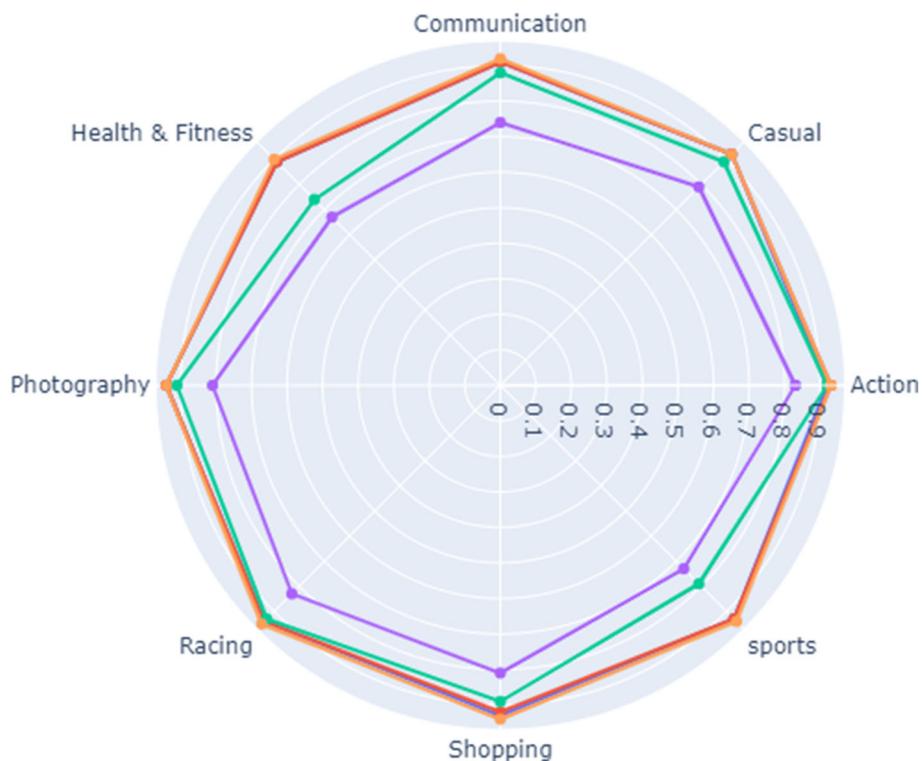


Fig. 11 Graphical illustration of precision results

Table 6 Precision results of machine learning algorithms

Classifiers	Action	Casual	Communication	Health and Fitness
LR	0.92	0.92	0.91	0.89
RF	0.94	0.92	0.91	0.91
MNB	0.92	0.89	0.88	0.74
KNN	0.83	0.79	0.74	0.67
SVM	0.93	0.92	0.92	0.89
Classifiers	Sports	Racing	Shopping	Photography
LR	0.93	0.94	0.93	0.93
RF	0.93	0.94	0.92	0.94
MNB	0.79	0.93	0.89	0.91
KNN	0.73	0.83	0.81	0.81
SVM	0.94	0.95	0.94	0.94

are diverse, contain noise and unwanted characters and of different length which makes their analysis a challenging task. This study leverages several well-known machine learning models for app reviews classification and investigate their performance. It thus helps the researcher to select the best available model when dealing with app review classification. For this purpose, a large dataset of app reviews containing 251661 reviews is scrapped which can help researchers study such reviews. In addition, various dimensions of the reviews are explored regarding the performance of machine learning models. For example, stats features like number of words, average word length, etc. tend to yield better accuracy. This can guide the researchers to explore the metadata of the app reviews and perform further experiments. Similarly, the computational complexity of the models can be targeted where the reviews can be clustered into groups and its influence on computational time and accuracy can be analyzed.

5 Conclusion and future work

With an increasing number of mobile phones, the development and deployment of mobile apps have become a potential market where users select and use apps to perform a large variety of tasks. The choice of an app is greatly influenced by the users' reviews posted for a particular app, in addition to using reviews for apps refining with existing and new features. Classifying such reviews into positive and negative would greatly help new users in the selection of appropriate apps. However, existing studies do not perform extensive experiments due to the lack of a large dataset. This study overcomes this limitation and scraps a large dataset comprising 251661 reviews for eight different app categories and fifty-nine apps. An approximately equal number of reviews are gathered to avoid the problem of imbalanced dataset. The dataset is used for experiments with several machine learning models like LR, RF, MNB, KNN, and SVM following preprocessing, stats feature collection and TF-IDF. Results indicate that SVM shows superior classification results obtaining a higher than 0.93 score for accuracy, precision, recall, and F1 score. Results suggest that the use of machine learning models with cleaned review can yield high accuracy results. Moreover, utilizing the stats features like the number of words, average word length, use of capital words, etc. for training the models show better performance. The performance of models varies regarding different app categories, as the nature of reviews is different for each category, however, the performance of SVM, LR, and RF proves to be more consistent. The study considers equal number of samples regarding app review categories and the impact of dataset imbalance is not studied. The collected dataset is large and provides the opportunity to perform analysis using other machine learning models on a large variety of reviews regarding app categories and apps. We intend to enlarge the dataset further by incorporating more app categories. Also, creating clusters and examining the link between the apps' reviews and ratings is also under consideration. For future studies, we intend to consider meta data to be included in the classification process to analyze its impact on the performance of machine learning models.

Data Availability The data is available upon request.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. A G Ganie SD, Lone MA (2021) Detection of semantic obsessive text in multimedia using machine and deep learning techniques and algorithms. *J Theor Appl Inform Technol* 99(11)
2. Abdul Rahman ANC, Abdullah IH, Zainudin IS, Tiun S, Jaludin A (2021) Domain-specific stop words in malaysian parliamentary debates 1959–2018. *GEMA Online J Lang Stud* 21(2):1–27
3. Alshqaq SS, Ahmadini AA, Abuzaid AH (2021) Some new robust estimators for circular logistic regression model with applications on meteorological and ecological data. *Math Probl Eng* 2021:15. Article ID 9944363. <https://doi.org/10.1155/2021/9944363>
4. Alyafeai Z, Al-shaibani MS, Ghaleb M, Ahmad I (2021) Evaluating various tokenizers for arabic text classification
5. Appbrain free versus paid android apps (2022). <http://www.appbrain.com/stats/free-and-paid-android-applications>. Accessed 10 Jan 2022
6. Aralikatte R, Sridhara G, Gantayat N, Mani S (2018) Fault in your stars: an analysis of android app reviews. In: Proceedings of the acm india joint international conference on data science and management of data. p 57–66
7. Ashraf I, Hur S, Park Y (2018) Magio: Magnetic field strength based indoor-outdoor detection with a commercial smartphone. *Micromachines* 9(10):534
8. Ashraf I, Alnumay WS, Ali R, Hur S, Kashif Bashir A et al (2021) Prediction models for covid-19 integrating age groups, gender, and underlying conditions. *Comput Mater Continua* 67(3):3009–3044
9. Assegie TA (2021) An optimized k-nearest neighbor based breast cancer detection. *J Robot Control (JRC)* 2(3):115–118
10. Behmanesh I, Moaveni B, Lombaert G, Papadimitriou C (2015) Hierarchical bayesian model updating for structural identification. *Mech Syst Signal Process* 64:360–376
11. Bhatia M, Kumar A, Beniwal R (2021) An optimized classification of apps reviews for improving requirement engineering. *Recent Adv Comput Sci Commun (Formerly Recent Patents Comput Sci)* 14(5):1390–1399
12. Bird S (2005) NLTK-Lite: Efficient Scripting for natural language processing. In: Proceedings of the 4th international conference on natural language processing (ICON). Allied Publishers Private Limited, pp 11–18. Allied Publishers
13. Byvatov E, Schneider G (2003) Support vector machine applications in bioinformatics. *Appl Bioinforma* 2(2):67–77
14. Dawoud NN, Samir BB, Janier J (2012) Fast template matching method based on optimized metrics for face localization. In: Proc. of the int'l multiconference of engineers & computer scientists. vol 1. p 700, Hong Kong
15. Deka PC et al (2014) Support vector machine applications in the field of hydrology: a review. *Appl Soft Comput* 19:372–386
16. Di Sorbo A, Grano G, Aaron Visaggio C, Panichella S (2021) Investigating the criticality of user-reported issues through their relations with app rating. *J Softw Evol Process* 33(3):e2316
17. Din Q, Donchev T, Kolev D (2018) Stability, bifurcation analysis and chaos control in chlorine dioxide–iodine–malonic acid reaction. *MATCH Commun Math Comput Chem* 79(3):577–606
18. Ezhilarasi S, Maheswari D (2021) Designing the neural model for POS tag classification and prediction of words from ancient stone inscription script. *Int J Aquatic Sci* 12(3):1718–1728
19. Gao F (2021) Applications of logistic regression and hazard method in accounting and finance research. Springer Books. In: Lee C-F, Lee AC (eds) Encyclopedia of Finance., edition 0, chapter 82. Springer, pp 1919–1937
20. Guzman E, Maalej W (2014) How do users like this feature? A fine grained sentiment analysis of app reviews. In: 2014 IEEE 22nd international requirements engineering conference (RE). Karlskrona, Sweden, pp 153–162. <https://doi.org/10.1109/RE.2014.6912257>
21. Handani SW, Saputra DIS, Arino RM, Ramadhan GFA et al (2019) Sentiment analysis for go-jek on google play store. In: Journal of Physics: Conference Series, vol 1196. IOP Publishing, p 012032, Palembang, Indonesia
22. Hassan S, Bezemer CP, Hassan AE (2018) Studying bad updates of top free-to-download apps in the google play store. *IEEE Trans Softw Eng* 46(7):773–793
23. Huang S, Cai N, Pacheco PP, Narrandes S, Wang Y, Xu W (2018) Applications of support vector machine (svm) learning in cancer genomics. *Cancer Genomics Proteomics* 15(1):41–51
24. Jacob C, Harrison R (2013) Retrieving and analyzing mobile apps feature requests from online reviews. In: 2013 10Th working conference on mining software repositories (MSR), pp 41–44

25. Jiao S, Geng B, Li Y, Zhang Q, Wang Q (2021) Fluctuation-based reverse dispersion entropy and its applications to signal classification. *Appl Acoust* 175:107857
26. Khalid M, Ashraf I, Mehmood A, Ullah S, Ahmad M, Choi GS (2020) Gbsvm: sentiment classification from unstructured reviews using ensemble classifier. *Appl Sci* 10(8):2788
27. Kück M, Freitag M (2021) Forecasting of customer demands for production planning by local k-nearest neighbor models. *Int J Prod Econ* 231:107837
28. Moslem Y, Haque R, Way A (2020) Arabisc: Context-sensitive neural spelling checker. In: Proceedings of the 6th workshop on natural language processing techniques for educational applications. Association for Computational Linguistics, Suzhou, pp 11–19
29. Mueez A, Ahmed K, Islam T, Iqbal W (2018) Exploratory data analysis and success prediction of google play store apps. PhD thesis, BRAC University
30. Nayezi M, Cho H, Ruhe G (2018) App store mining is not enough for app improvement. *Empir Softw Eng* 23(5):2764–2794
31. Noble WS et al (2004) Support vector machine applications in computational biology. *Kernel Methods Comput Biol* 14:71–92
32. Oktaviani V, Warsito B, Yasin H, Santoso R et al (2021) Sentiment analysis of e-commerce application in traveloka data review on google play site using naïve bayes classifier and association method. In: journal of physics: conference series, vol 1943. IOP Publishing, Indonesia, p 012147
33. Pagano D, Maalej W (2013) User feedback in the appstore: An empirical study. In: 2013 21St IEEE international requirements engineering conference (RE). IEEE, Rio de Janeiro, pp 125–134
34. Pal M (2005) Random forest classifier for remote sensing classification. *Int J Remote Sens* 26(1):217–222
35. Pallathadka H, Ramirez-Asis EH, Loli-Poma TP, Kaliyaperumal K, Ventayen RJM, Naved M (2021) Applications of artificial intelligence in business management e-commerce and finance. <https://doi.org/10.1016/j.matpr.2021.06.419>
36. Pappas N, Popescu-Belis A (2013) Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval (SIGIR '13). Association for Computing Machinery, New York, pp 773–776. <https://doi.org/10.1145/2484028.2484116>
37. Prasad GNR (2021) Identification of Bloom's Taxonomy level for the given Question paper using NLP Tokenization technique. *Turkish J Comput Math Educ* 12(13):1872–1875
38. Pu Y, Grossklags J (2015) Using conjoint analysis to investigate the value of interdependent privacy in social app adoption scenarios
39. Qin Y, Wang X, Xu Z (2021) Ranking tourist attractions through online reviews: a novel method with intuitionistic and hesitant fuzzy information based on sentiment analysis. *Int J Fuzzy Syst* 24(2):755–777. PMCID: PMC8224999
40. Ramalingam V, Dandapat A, Raja MK (2018) Heart disease prediction using machine learning techniques: a survey. *Int J Eng Technol* 7(2.8):684–687
41. Rehan M, Rustam F, Ullah S, Hussain S, Mehmood A, Choi GS (2021) Employees reviews classification and evaluation (ERCE) model using supervised machine learning approaches. *J Ambient Intell Humanized Comput* 13:3119–3136
42. Ruan S, Chen B, Song K, Li H (2022) Weighted naïve bayes text classification algorithm based on improved distance correlation coefficient. *Neural Comput Appl* 34(4):2729–2738
43. Rupapara V, Rustam F, Amaar A, Washington PB, Lee E, Ashraf I (2021) Deepfake tweets classification using stacked bi-lstm and words embedding. *PeerJ Comput Sci* 7:e745
44. Rustam F, Ashraf I, Mehmood A, Ullah S, Choi GS (2019) Tweets classification on the base of sentiments for us airline companies. *Entropy* 21(11):1078
45. Rustam F, Mehmood A, Ahmad M, Ullah S, Khan DM, Choi GS (2020) Classification of shopify app user reviews using novel multi text features. *IEEE Access* 8:30234–30244
46. Rustam F, Reshi AA, Mehmood A, Ullah S, On B-W, Aslam W, Choi GS (2020) Covid-19 future forecasting using supervised machine learning models. *IEEE Access* 8:101489–101499
47. Rymarczyk T, Kozłowski E, Kłosowski G, Niderla K (2019) Logistic regression for machine learning in process tomography. *Sensors* 19(15):3400
48. Saad E, Din S, Jamil R, Rustam F, Mehmood A, Ashraf I, Choi GS (2021) Determining the efficiency of drugs under special conditions from users' reviews on healthcare web forums. *IEEE Access* 9:85721–85737
49. Trafalis TB, Ince H (2000) Support vector machine for regression and applications to financial forecasting. In: Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks. IJCNN 2000. Neural Computing: new challenges and perspectives for the new millennium, Como, Italy, 2000, vol 6, pp 348–353, <https://doi.org/10.1109/IJCNN.2000.859420>

50. Umer M, Ashraf I, Mehmood A, Ullah S, Choi GS (2021) Predicting numeric ratings for google apps using text features and ensemble learning. *ETRI J* 43(1):95–108
51. Vinodhini G, Chandrasekaran R (2012) Sentiment analysis and opinion mining: a survey. *Int J* 2(6):282–292
52. Watzlawik M, Valsiner J (2012) The making of magic: cultural constructions of the mundane supernatural. *Oxford Handb Cult Psychol* 2(6):1930–1938. <https://doi.org/10.1093/oxfordhb/9780195396430.013.0038>
53. Wahyuni WA, Saepudin S, Sembiring F (2022) Sentiment analysis of online investment applications on google play store using random forest algorithm method. *Jurnal Mantik* 5(4):2203–2209. Madrid, Spain
54. Wu H, Deng W, Niu X, Nie C (2021) Identifying key features from app user reviews. In: 2021 IEEE/ACM 43Rd international conference on software engineering (ICSE). IEEE, p 922–932
55. Ye H, Wu P, Zhu T, Xiao Z, Zhang X, Zheng L, Zheng R, Sun Y, Zhou W, Fu Q et al (2021) Diagnosing coronavirus disease 2019 (covid-19): Efficient harris hawks-inspired fuzzy k-nearest neighbor prediction methods. *IEEE Access* 9:17787–17802
56. Yousaf A, Umer M, Sadiq S, Ullah S, Mirjalili S, Rupapara V, Nappi M (2020) Emotion recognition by textual tweets classification using voting classifier (lr-sgd). *IEEE Access* 9:6286–6295
57. Zabor EC, Reddy CA, Tendulkar RD, Patil S (2021) Logistic regression in clinical studies. *Int J Radiat Oncol Biol Phys* 112(2):271–277. <https://doi.org/10.1016/j.ijrobp.2021.08.007>
58. Zhan C, Zheng Y, Zhang H, Wen Q (2021) Random-forest-bagging broad learning system with applications for covid-19 pandemic. *IEEE Internet Things J* 8(21):15906–15918
59. Zhao P, Gou G, Liu C, Guan Y, Cui M, Xiong G (2021) TMT-RF: tunnel mixed traffic classification based on random forest. In: Garcia-Alfaro J, Li S, Poovendran R, Debar H, Yung M (eds) Security and privacy in communication networks. SecureComm 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 398. Springer, Cham. https://doi.org/10.1007/978-3-030-90019-9_21
60. Zimmeck S, Wang Z, Zou L, Iyengar R, Liu B, Schaub F, Wilson S, Sadeh N, Bellovin S, Reidenberg J (2017) Automated analysis of privacy requirements for mobile Apps. In: Proceedings 2017 network and distributed system security symposium (proceedings 2017 network and distributed system security symposium). Korea Society of Internet Information. <https://doi.org/10.14722/ndss.2017.23034>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Affiliations

Ali Adil Qureshi¹ · Maqsood Ahmad² · Saleem Ullah¹ · Muhammad Naveed Yasir³ · Furqan Rustam⁴ · Imran Ashraf⁵ 

 **Imran Ashraf**
imranashraf@ynu.ac.kr

Ali Adil Qureshi
aliadil2201@gmail.com

Maqsood Ahmad
maqsoodzee@gmail.com

Saleem Ullah
Saleem.ullah@kfueit.edu.pk

Muhammad Naveed Yasir
naveed.yasir@gmail.com

¹ Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan

² Department of Information Security, The Islamia University of Bahawalpur, Bahawalpur, Punjab 63100, Pakistan

³ Department of Computer Science, University of Narowal, Narowal 51600, Pakistan

⁴ School of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland

⁵ Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea