ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

Mobile Application Review Classification Using Machine Learning Approach

Laukik Padgaonkar[1],Sahit Jain[1], Swarali Ajgaonkar[1], Rahul Londhe[1], Prof. K.S.Balbudhe[2]

B.E. Student, Dept. of IT, PVG's COET, Pune, India[1]

Asst. Professor, Dept. of IT, PVG's COET, Pune, India [2]

ABSTRACT:A vast number of people use mobile apps in their daily life. Based on their experience with a mobile application, they post reviews of the application. The number of these reviews can reach millions and this makes it extremely difficult and time consuming for the developers to study the reviews and take appropriate actions depending on the reviews for further development. Thus, this project aims to perform automatic classification of mobile application reviews using Machine Learning. The main technique used behind the automatic classification is Supervised Machine Learning. The mobile reviews are classified into four categories: Problem Description, Feature Enhancement, Spam and Other. For this purpose, the features used for training the classifier are: Unigrams, Bigrams, Star Ratings, Sentiment Score of the reviews. The machine learning algorithms used to classify the reviews are: Naive Bayes Classifier, Support Vector Machines, Decision Trees and Random Forests.

KEYWORDS:Machine learning algorithm, Classification

I. INTRODUCTION

The proposed system will be used to classify the application review of an application on the play store. The data set used for this purpose will consist of text reviews, emojis & star ratings. This dataset has been collected from a website heedzy.com, which provides lifetime reviews of apps in "csv" format. For the purpose of classification, Supervised Machine Learning Techniques will be used. From collected datasets contain 4 main features for model training have been extracted: Unigram, Bigram, Star rating and Sentiment of the review. The target or classes are Problem Discovery, Feature Enhancement, Spam and Other. Based on the training dataset the model will be trained to predict the input review data's category. The training dataset has been generated after cleaning the review data set. The data cleaning process includes conversion to lower case, removal of digits, removal of unnecessary spaces, spell-correction, removal of stopwords. After the cleaning process, sentiment analysis has been performed and then the review dataset will be converted to unigram, bigram, star ratings, sentiment score. These features and the training dataset will be used to train the model. The Machine Learning model then compares the input features with the features in training datasets. Based on this comparison the classifier will put this review into appropriate category type. If the reviewer has posted a review like "This application is good but it has a bug while opening a tab" then it will come under the class of problem discovery. If the review is like, "The application could be better if it performed same extra tasks" then it comes under the class of feature enhancement. If the review said " Make easy money by clicking on some link" then it would come under spam. According to these requirements, the features will be associated with the respective targets. These features and targets will be used to train various Supervised Machine Learning models. The machine learning algorithms which will be used for the same are Naive Bayes Classifier, Support Vector Machines, Decision Trees and Random Forests.

II. RELATED WORK

Research on app-review classification in English had been conducted by several studies [1][2][3][4][5]. Reference [5] is the improvement studies use similar method. The difference was on the features used for classification. Reference [4] was also the improvement of [3] where in [4] the application was made for the app-review classification made in [3].

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

In determining the classes of app-reviews, those five studies had several differences. Reference [1] used 9 classes: bug report, feature request, feature strength, featureshortcoming, praise, complaint, usage scenario and other, while [3][4] used only 4 classes: bug report, feature request, user experience and star rating. Reference [2][5] used five classes: problem discovery, feature request, information seeking, information giving, and other. Problem discovery class in [2][5] has the same meaning with bug report in [1], [3][4].

We can see that these five studies agree on two classes: bug report and feature request as the classes of reviews that were important.

Due to lack of standard dataset, those studies used different dataset that they build by themselves. Reference [1] create dataset of 4,550 text reviews with multi-label classes and the number of reviews for the 9 classes defined was unbalanced. Reference [1][4] use the same dataset created in [3] that consisted of 4,400 reviews that single-label and unbalanced dataset. Reference [2] use 1390 reviews that were single label that also unbalanced and [5] create dataset of 852 reviews that single-label and unbalanced.

Dataset was preprocessed before being used for classification. All five studies conducted stemming and stopword removal on the preprocessing stage. Reference [2][5] are also tokenizing the sentences since they are classifying the reviews at sentence level granularity.

After that, feature extraction was done. Reference [1] use 8 groups of features: unigram with tf-idf weighting, star rating, text length, number of uppercase and lowercase, number of exclamation character, @, and space, average word length, and the percentage of number of words with positive and negative sentiment. Reference [3][4] use similar features: unigram, star rating, text length, tense type, and SentiStrength score, but in [4] they used additional feature bigram. Reference [2][5] use the same features: unigram, linguistic pattern, sentiment score in range [-1, 1]. This sentiment score was calculated by using sentiment analysis classification as the prior step.

Those five works compare the performance of several machine learning algorithms in classifying the app review text. All previous work but [2] conduct experiments using Naïve Bayes and Logistic Regression. References [1][2] also used Support Vector Machine (SVM). Beside of that all but [1] used Decision Tree family. In addition, [1] also used neural network.

III.PROPOSED METHOD

The work begins with collection of review dataset from www.heedzy.com for two particular apps - Facebook and Instagram. The reviews are collected from the open source Google Play Store platform. The collected dataset will be cleaned and then harvested to create the classifier's training features. The dataset cleaning process includes conversion of reviews to lower case, removing digits, removing punctuation marks, emoji replacement, removal of extra white spaces and removal of stopwords. Sentiment analysis is then performed on the cleaned dataset and the sentiment score of each review will added to the dataset accordingly. The features: Unigrams, Bigrams, Star Ratings and Sentiment Score are then extracted from the cleaned and pre-processed dataset.

After manual identification of which feature corresponds to which class, the training dataset and testing dataset will be generated. The training dataset will be used to train the Machine Learning Model (Classifier). Four algorithms - Naive Bayes Classifier (NBC), Support Vector Machine (SVM), Decision Trees (DT) and Random Forest (RF) will be used for training the model. The trained model will then take a single review or a review dataset as an input and give an output with corresponding class label for each input review. The performance of each classification model will be compared and the best algorithm or technique will be used in the final system.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

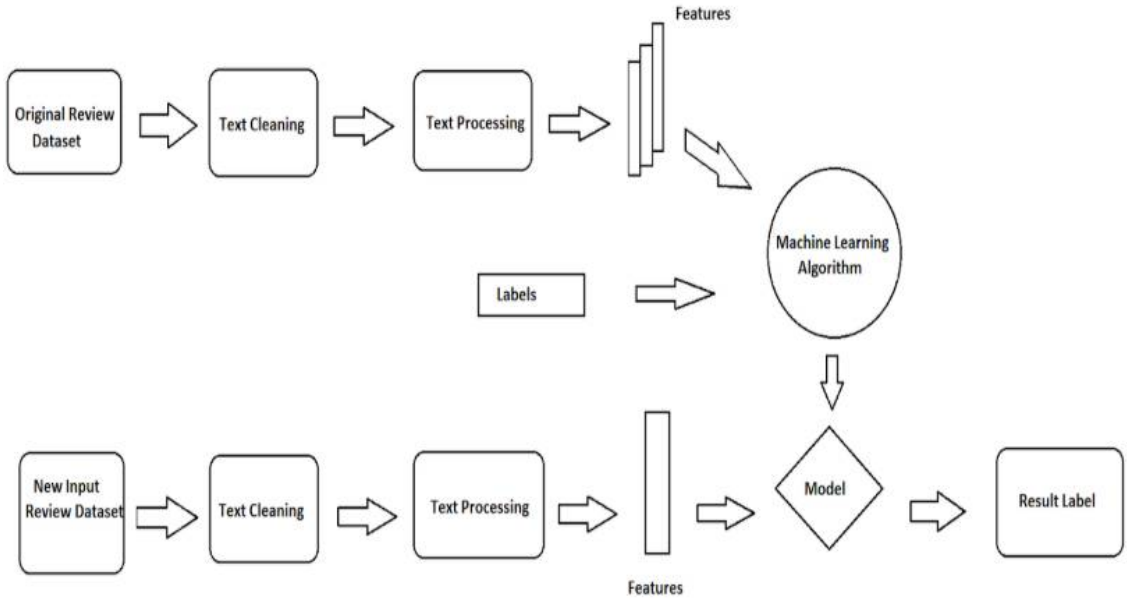
International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

System Architecture



IV. EXRIMENTAL RESULTS

A. Dataset

We obtained 50,000 reviews of two application: Instagram and Facebook each, present on the Google App Store through heezdy.com. The dataset in a ".csv" format contained seven fields: Source, Date, Title, Content, Name, Rating and Version. Providing the Name of the application, the date when the review was posted, the title of the review, the actual review, name of the user who submitted the review, star rating given by the user varying from 0 to 5 and the version of the application for which the review was posted.

B. Model Training

We took 3000 labelled reviews, vectorized them using TFIDF vectorization technique and used this data to train four classification models :Random Forest Classifier, Decision Tree Classifier, Naive Bayes Classifier, Classifier and Linear Support Vector Machine.

We extracted four main features from the dataset :

- Unigram
- Bigram
- Star Rating
- Sentiment Score

We trained the model for 20 times with 2250 reviews and then achieved the following average.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

Training Accuracies :

- Random Forest: 0.79
- Decision Tree: 0.78
- Naive Bayes: 0.66
- Linear SVM: 0.78

We tested the model for 20 times with 2250 reviews and then achieved the following average.
Training Accuracies :

- Random Forest: 0.90
- Decision Tree: 0.88
- Naive Bayes: 0.67
- Linear SVM: 0.87

The highest accuracies for the models were obtained using single feature. As per the test results, the most informative feature was the Unigrams. The models trained using Unigrams only had the highest accuracies followed by Bigrams, Unigrams + Star Rating, Unigram + Sentiment Score and Unigrams + Bigrams.

V. CONCLUSION

As our test results, we conclude that for classifying Problem Description, Feature Enhancement, Spam and Other, Random Forest Classifier is better than Naïve Bayes Classier, especially when dataset is unclean contains redundancies and garbage character. This argument is supported by the achieved test results where in the Testing Accuracy of Random Forest Classifier is on an average 91% which is significantly higher than that of Naïve Bayes Classifier which is 67 %. Our secondary conclusion is that the use of single feature, namely Unigrams, proved to be more effective than using combination of one or more features.

REFERENCES

[1]. Y. Ekanata, I. Budi, " Mobile Application Review Classification for the Indonesian Language Using Machine Learning Approach," 4th International Conference on Computer and Technology Applications, pp. 117-121, 2018.
[2]. E. Guzman, M. El-Haliby, and B. Bruegge, "Ensemble Methods for App Review Classification: An Approach for Software Evolution (N)," 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), , pp. 771–776, 2015.
[3]. S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can i improve my app? Classifying user reviews for software maintenance and evolution," IEEE 31st International Conference on Software Maintenance and Evolution, ICSME 2015 - Proceedings, pp. 281–290, 2015.
[4]. W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews," IEEE 23rd International Requirements Engineering Conference, RE 2015 - Proceedings, pp. 116–125, 2015.
[5]. W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik, "On the automatic classification of app reviews," Requir. Eng., vol. 21, no. 3, pp. 311–331, 2016.
[6]. S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "ARdoc: app reviews development oriented classifier," Proc. 2016 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng. - FSE 2016, pp. 1023–1027, 2016.
[7]. R. Feldman and J. Sanger, "The text mining handbook: advanced approaches in analyzing unstructured data," Imagine, vol. 34, pp. 410, 2007.
[8]. C. M. Bishop, Pattern Recognition And Machine Learning. 2006.
[9]. V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 4, pp. 42–47, 2012.
[10]. F. Pedregosa and G. Varoquaux, Scikit-learn: Machine learning in Python, vol. 12. 2011.
[11]. Google, "Write a review on Google Play." [Online]. Available: https://support.google.com/googleplay/answer/4346705. [Accessed: 15-Nov-2018].
[12]. I. Alfina, D. Sigmawaty, F. Nurhidayati, and A. N. Hidayanto, "Utilizing Hashtags for Sentiment Analysis of Tweets in The Political Domain," in 9th International Conference on Machine Learning and Computing, 2017, pp. 43–47. 2017.