Global Journal of Modeling and Intelligent Computing (GJMIC)

Vol. 3, No. 1, January, 2023 ISSN: 2767-1917

MACHINE LEARNING BASED EXPENSE TRACKER APPLICATION FOR PERSONAL FINANCE MANAGEMENT

Raj Thakare, SSGMCE Shegaon, India (rajthakare237@gmail.com)
Ninad Thakare, SSGMCE Shegaon, India (ninadthakare1305@gmail.com)
Raj Sangtani, SSGMCE Shegaon, India (rajsangtani1911@gmail.com)
Shubham Bondre, SSGMCE Shegaon, India (sbondre02@gmail.com)
Amitkumar Manekar, SSGMCE Shegaon, India (asmanekar24@gmail.com)

ABSTRACT

This study addresses the issue of tracking daily expenses and proposes a Naive Bayes based Expense Tracker system. The system utilizes Naive Bayes algorithm to detect Bank SMS and read user expenses automatically. This work presents the implementation of Naive Bayes algorithm to classify Bank SMS and also studies the performance of the algorithm in terms of "precision" & "accuracy". Finally compared performance of the "Naive Bayes" Algorithm with other machine learning algorithms. The proposed system can help common people to better manage their finances and make informed decisions about their spending habits.

Keywords: Machine Learning, Personal Finance Management, Expense Tracking, Predictive Modeling, User Interface.

1. INTRODUCTION

The "Expense Tracker" is a mobile application for android users. The app is developed using Kotlin and XML languages in android studio. "Expense Tracker" helps users to keep track of their daily expenses so that they can manage their finances properly and have complete idea of their spending habits. All the expenses are shown with help of Pie Chart and Bar Graph so that users get quick and clear idea of their spendings. Not only this app allows user to enter the expense manually along with amount, category and description of the expense, but also it can automatically detect the bank SMS from your mobile phone and read the credited or debited amount. Firebase is used as the online database for the app, so that even if you uninstall the app, still your data will be saved. To use the "Expense Tracker" app the user will have to sign up using email address and password.

Text classification technique is used to detect Bank SMS accurately. In text classification a category is assigned to a document. SMS are text messages, and we want to classify SMS as "Bank SMS" or "Normal SMS". We use supervised machine learning in text classification and that is why the labelled dataset is required. So, to classify Bank SMS we have used NB (Naive Bayes) supervised ML algorithm, which is mostly used in textual classification. Also, in this study we analysed the performance of the NB (Naive Bayes) algorithm on the basis of "precision" and "accuracy". Also compared performance of NB (Naive Bayes) with other ML algorithms such as SVC (Support Vector Classifier), LR (Logistic-Regression), ETC (Extra-Trees-Classifier), KNC (K-Neighbours-Classifier), RFC (Random-Forest-Classifier) and DTC (Decision-Tree-Classifier).

2. LITERATURE REVIEW

Agarwal et al. (2018) discussed the difficulty of multi-label classification in NLP and text mining, and the achievement of finding a solution for this problem in the less explored research domain of Bengali language. The authors suggest using different word embedding techniques in the future and consider unsupervised or semi-supervised learning models for predicting new labels in a more cost-effective way. Azam et al. (2018) discussed "Feature-Extraction" based "Text-Classification" using KNN (K-Nearest-Neighbour). This algorithm K-NN performed better than Naïve Bayes, and Bagging and Boosting increased accuracy with Naïve Bayes. Bharat et al. (2021) talk about the study Detecting Fake News Using Machine Learning Algorithms authors conclude that feature selection is crucial in a text

classification system, as demonstrated by their results when compared to a system using the same dataset without feature selection. Bhuiyan et al. (2018) provide an impression of different existing spam filtering systems using ML techniques, summarizing their accuracy and effectiveness in filtering email spam. While all the methods are effective, some have better outcomes and others are implementing additional processes to increase their accuracy. Suneera et al. (2020) analyzed ML and DL (Deep Learning) models for text classification on 20Newsgroup data-set. The TF-IDF representation outperformed "Word2Vec" and "BERT" embeddings, and LR (Logistic Regression) and a bi-channel CNN model produced excellent results. Chaurasia et al. (2021) discussed the importance of detecting and eliminating spam messages in SMS and how existing research mainly uses Naive Bayes but with low accuracy. The authors used a python library called EVALML to automate finding the best algorithm and found that logistic regression had the highest accuracy. De Vries et al. (2020) aimed to classify "Aviation Safety Reports" using ML techniques and paper concludes that the RFC (Random-Forest-Classifier) is a feasible option for classifying the occurrence reports, achieving an accuracy of 80%-93%. Deepa et al. (2019) proposed system uses a Convolutional Neural Network for image classification and Python-Tesseract OCR package for text extraction, resulting in better performance compared to existing systems due to the ability to overcome overfitting. Deshmukh et al. (2021) proposed system uses AI techniques to identify fake reviews and the system achieved an accuracy of 80% with precision of 88%, recall score of 72%, and an F1 score of 79% using various evaluation metrics. Gupta et al. (2018) focused on evaluating ML techniques for spam sms detection and compares 8 different classifiers. The CNNC (Convolutional-Neural-Network-Classifier) achieved the maximum accuracy of 99.19% & 98.25% for the two datasets, surpassing traditional classifiers like SVM and NB. Gupta et al. (2019) proposed a ML technique for SMS spam filtering using NB (Naïve-Bayes) and SVM (Support-Vector-Machine) algorithms. The dataset used in the study consisted of 5574 observations of 2 variables, the email content, and the target variable (spam or ham). Naive Bayes outperformed SVM in the study. Katpatal et al. (2018) discussed the problem of detecting spamming activities on Twitter and the need to protect user privacy. The Lfun approach is introduced for spam detection, which involves training classifiers with training data and adding detected spam tweets to the training set over time. Krishnaveni et al. (2021) discussed the issue of SMS spam and proposes a method to detect it using ML classifiers such as NB (Naïve-Bayes) and SVM (Support-Vector-Machine). The study compared the performance of these classifiers and found that SVM had a higher accuracy of 94.32% in detecting spam SMS compared to Naïve Bayes. Kumar et al. (2020) discussed email spam identification using ML algorithm and the study found that Multinomial Naïve Bayes performs best but has limitations, while ensemble methods are useful. Kurniawan et al. (2021) present the use of machine learning algorithms and term weighting schemes hate speech classification in Indonesian Twitter, with SVM and word unigrams performing best. Ma et al. (2020) aimed to classify spam emails from legitimate ones using Naive Bayes and SVM. SVM showed higher accuracy than NB (Naive Bayes) in diverse sizes of training emails. Moldagulova et al. (2017) proposed a ML system that uses KNN (K-Nearest-Neighbours) algorithm in R for classification of textual documents, and it highlights the importance of finding the proper value of k and presents experimental outcomes showing the top accuracy percentage when k changes from 1 to 50. Murugavel et al. (2020) proposed a framework to improve the prediction by the selection of correct attributes that contributes to better data extraction. Nagre et al. (2018) presented a systematic literature review of 13 research papers on SMS spam detection, analysing their proposed techniques, advantages, disadvantages, challenges, and evaluation procedures. Nandhini et al. (2020) discussed methods to build a ML model to identify spam emails using popular algorithms and the random tree outperforms other classification algorithms in spam filtering. Pal et al. (2020) aimed to classify Hindi Poetries using machine learning algorithms. The task is challenging due to the language's morphological variance. The best accuracy achieved was 64%, with Naïve Bayes and Random Forest algorithms performing better than others. Prasad et al. (2021) discussed the problem of email spam and proposes the use of the Naive Bayesian classifier for spam filtering, using the Ling spam dataset for classification. Pooja et al. (2018) showed classification system was developed using the Naïve Bayesian Classifier, which proved to have a low error rate and better accuracy than the Support Vector Machine. After analysing several research studies, Raza et al. (2021) found that the supervised machine learning approach is highly adopted and produces consistent and accurate results. NB (Naive Bayes) and SVM algorithms are in high demand compared to other ML algorithms. Reddy et al. (2021) discussed the effectiveness of spam filtering methods and concludes that Random Forests perform better than other Machine Learning methods in classifying spam messages. The RF (Random-Forest) technique was used for classification of emails in the study, which achieved an accuracy score of 92%. Santoshi et al. (2021) reviewed the use of classification techniques like "SVM", "Random-Classifier" "Bayes-theorem" and "decision-tree" for detecting spammers in social networks. Selvapattu et al. (2020) proposed the classification model is tested on a prepared dataset to measure SMS spam detection performance using accuracy. Naïve Bayes classifier showed the highest accuracy among other classifiers. According to Sethi et al. (2017) NB (Naïve Bayes) outperforms LR (logistic regression) and RF (random forest) algorithms in SMS spam classification with high accuracy of 98.445%. Random forest also performed well and has a shorter running time, but Naïve Bayes

is the best choice. Shradhanjali et al. (2017) proposed a new way for email spam detection to effectively detect spam emails and block them while retaining genuine mail. This classifier gives 98% of accuracy while classifying data-sets. Singh et al. (2018) proposes a technique for email spam identification using SVM, which handles a large number of attributes. This study compares Linear & Gaussian kernel methods and finds that the linear kernel-based scheme outperforms the gaussian kernel in terms of "accuracy" & speed of operation. Taloba et al. (2019) proposed a new approach called GADT, which uses a hybrid of genetic algorithm and decision tree for email spam detection, with the added benefit of using PCA to reduce high dimensionality. Tanvir et al. (2019) discussed a Deep-Learning based model proposed to detect fake news on Twitter using supervised models. The accuracy of the model can be improved by using more complex algorithms, and the study shows that even basic algorithms in AI and ML can have a significant impact on such critical matters as the spread of fake-news. Vijay et al. (2021) discussed the Naïve Bayes classifier correctly classified the majority of ham and spam messages in a test dataset, with a small number of errors in each category. Ziyan et al. (2019) showed that Naïve Bayes had higher accuracy on Spam Data but lower precision than SPAMBASE, where precision was more important for classifying spam emails correctly.

3. METHODOLOGY

To understand the methodology, we need to understand the workflow of the system. Initially, the expense tracker app reads the SMS messages from the user's device & feeds them to the ML model. This ML model accurately identifies bank-related messages and returns them to the app. The app then extracts the credited or debited amount from the user's bank messages and adds the entry of the expense into the database. Finally, the app displays the user's data using various charts.

To build a ML model that detects the bank SMS accurately we need to perform following steps:

I. Importing Libraries
II. Data Cleaning:
III. Data Preprocessing:
IV. Model Building:
V. Evaluation:

I. Importing Libraries:

NumPy: It's a library from python used to deal with arrays, and in addition it provides features of matrices, linear algebra, etc.

Pandas: It's a python library used to deal with datasets.

Matplotlib: This library is used for creating animated, static and interactive visualizations in Python.

Sklearn: Python library with various effective methods for statistical modelling and machine learning.

II. Data Cleaning:

At first, we imported the dataset with the help of pandas' library. We removed all the unnecessary columns from the dataset so that we have only two columns left, one is the text column containing SMS text and other one is target column containing only two values either the text is "Bank SMS" or "Normal SMS". Now as a next step we encoded the target column so that it contains value as either 0 or 1. We checked if there were any duplicate values present in our dataset and then we deleted the duplicate values.

III. Data Preprocessing:

To improve the accuracy of the ML model we preprocessed the data and then fed it to the model. To preprocess the data, we created a function which converted the text of the SMS to the lower case, also it performed the tokenization of the text along with removal of the special characters present in the SMS.

IV. Model Building:

In this step we will build naïve bayes model which is based on Bayes' theorem (Explained in *4. Naïve Bayes Classifier*). After data preprocessing we first imported the TfidfVectorizer (an algorithm that assesses the relevance of words to a given document using their frequency) and applied that on the text column of the dataset. Next, we created the dependent & the independent variables and then dataset was split into the training set & test set in the ratio given as 80 : 20 respectively. The NBC (Naïve-Bayes-Classifier) was imported & trained on the training dataset.

V. Evaluation:

After training of the naive bayes classifier, we evaluated the model on "accuracy" and "precision". At the end we achieved accuracy of 99.33% and precision of 97.67%.

4. NAÏVE BAYES CLASSIFIER

The NBC (Naïve-Bayes-classifier) is used in this study for classification of Bank SMS messages. It is a supervised machine learning technique which is based on "Bayes' theorem". This theorem explains the relationship between conditional probabilities and is a fundamental idea in probability theory.
The formula for Bayes' theorem is:

$$p(a \mid b) = \frac{p(b \mid a) * p(a)}{p(b)} \quad (1)$$

$p(a \mid b)$ gives probability of event 'a' occurring, given that event 'b' has occurred. It is called as the posterior probability.
$p(b \mid a)$ gives probability of event 'b' occurring, given that event 'a' has occurred. It is called the likelihood.
$p(a)$ gives prior probability of event 'a' occurring, which is our initial estimate or belief about the probability of 'a' happening before we have any additional evidence.
$p(b)$ gives prior probability of an event 'b' occurring, which is our initial estimate or belief about the probability of 'b' happening before we have any additional evidence.
The formula calculates probability of 'a' given 'b' by taking the likelihood of 'b' given 'a' and multiplying it with the prior probability of 'a', and then dividing by the total probability of 'b'.

5. USE CASE DIAGRAM

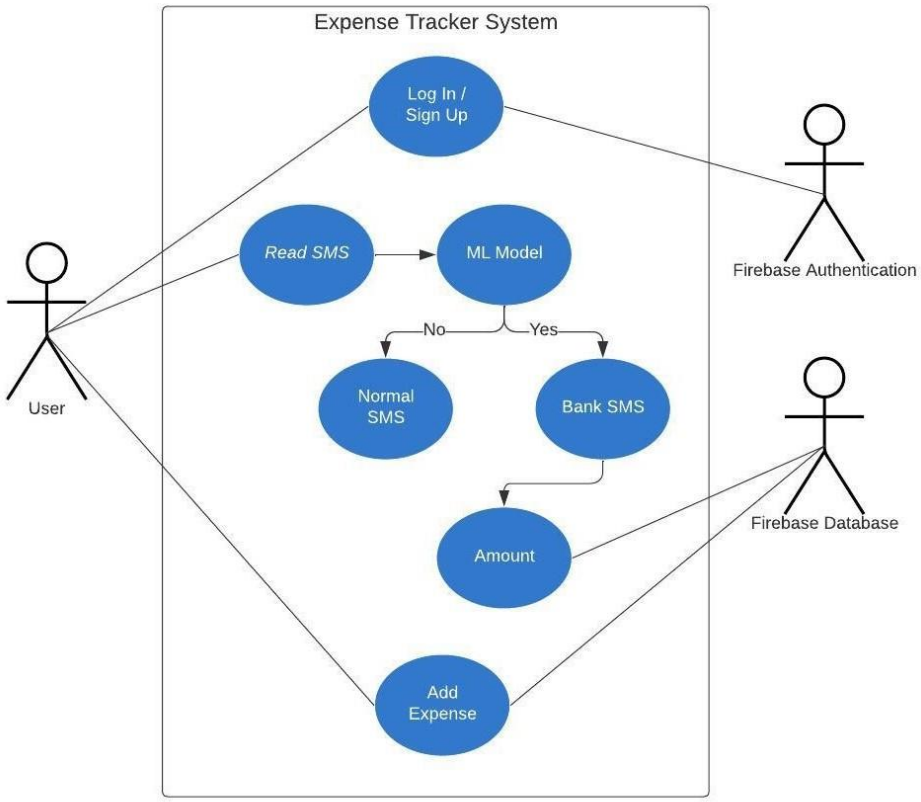

Figure 1: Use Case Diagram

Given Figure 1 shows the use case diagram for the working of expense tracker application. It shows that the expense tracker system allows user to log in / log out with the help of firebase authentication, also the application

reads the user messages and classifies them into bank or normal message with the help of machine learning model which is accessed through API. Firebase real-time database is used to store the use information in online storage.

6. CLASS DIAGRAM

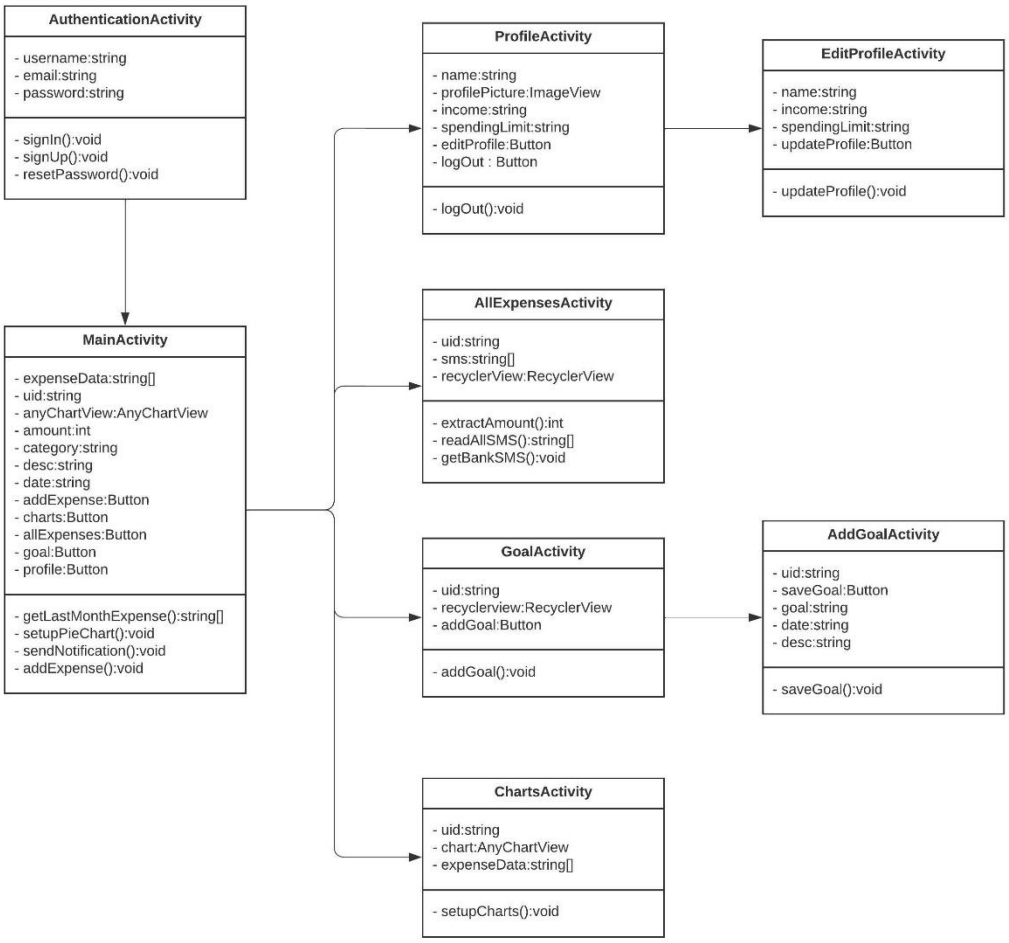

Figure 2: Class Diagram

The above figure 2 shows the class diagram for expense tracker application. This class diagram shows how various classes are related to each other and what are various properties they have and various functions they can perform. The authentication class has the properties like username, email and password etc, this class performs various functions like signIn(), signUp() and resetPassword(). MainActivity class is responsible for the home screen of the app and it has properties such as uid, amount, category, date etc, and it performs functions like setupPieChart(), sendNotification() and addExpense(). ProfileActivity class has attributes such as name, profile picture, income etc, and it has methods such as logout(). Another class related to ProfileActivity class that is EditProfileActivity class and it has same properties as previous once but different function and that is updateProfile(). AllExpensesActivity class has properties such as uid, sms, recyclerview and it performs functions like extractAmount(), readAllSMS() and getBankSMS(). There are also some other classes such as GoalActivity and ChartsActivity which have their own properties and they perform functions such as addGoal() and setupCharts().

7. COMPARISON & ANALYSIS

Algorithm	Accuracy	Precision
NB	0.993377	0.976744
SVC	0.986755	0.976190
ETC	0.986755	0.976190
RFC	0.980132	0.975610
LRC	0.973510	0.975000
KNC	0.986755	0.954545
DTC	0.986755	0.954545

Figure 3: Performance Metrics of Various Algorithms

After evaluating the Naïve bayes model we wanted to check if there is any other algorithm that can perform better than current so we compared its performance with multiple other ML algorithms such as SVC (Support-Vector-Classifier), ETC (Extra-Tree-Classifier), RFC (Random-Forest-Classifier), LRC (Logistic-Regression-Classifier), KNC (K-Neighbors Classifier) and DTC (Decision Tree Classifier). From the figure 2 you can see that after comparing all the algorithms with Naïve bayes, the NB (Naïve bayes) still gives results better that all other algorithms in terms of both "accuracy" and "precision" with accuracy of 99.33% and Precision of 97.67%. Second best performer is Support Vector Classifier with accuracy of 98.67% and precision of 97.61%. Furthermore, the Extra Tree Classifier gives accuracy of 98.67% and precision of 97.61, Random Forest gives accuracy of 98.01% and precision of 97.56%, Logistic Regression gives accuracy of 97.35% and precision of 97.50%, KNC gives accuracy of 98.67% and precision of 95.45% and DTC gives accuracy of 98.67% and precision of 95.45%.

8. RESULT

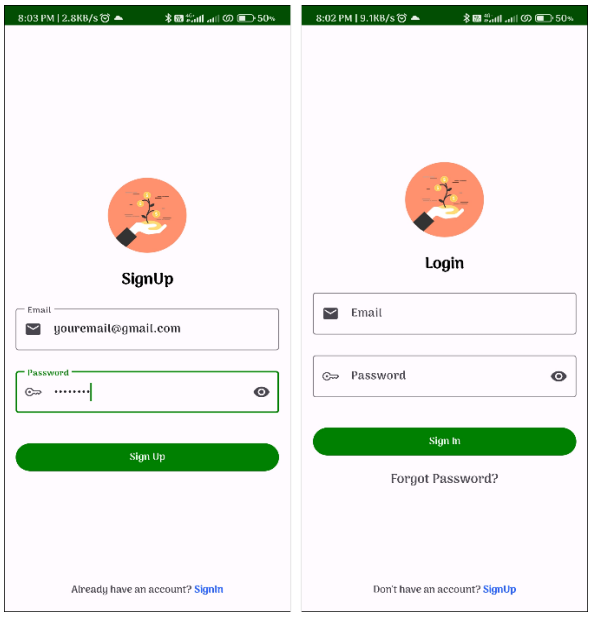

Figure 4: Sign Up and Log In

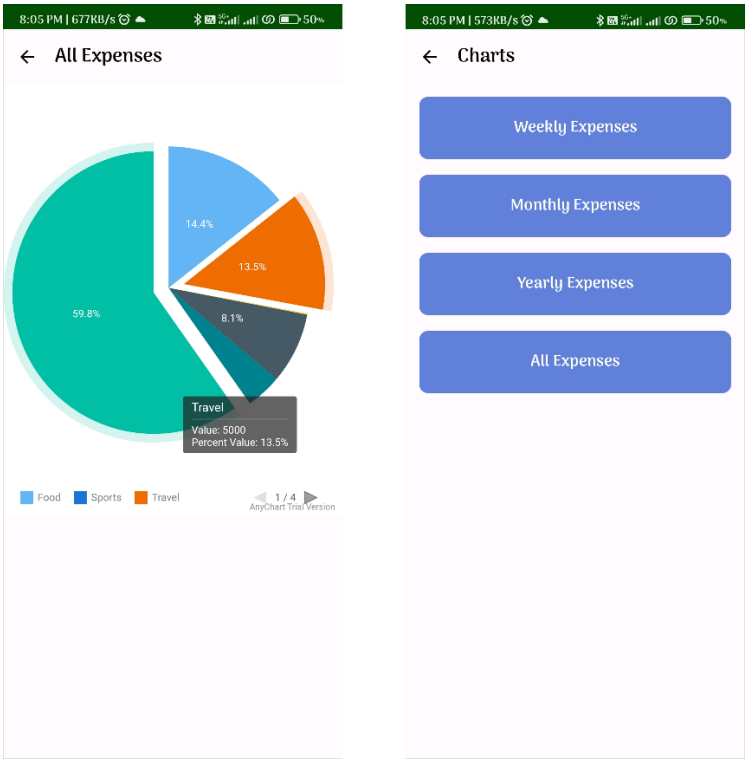

Figure 5: Various Charts

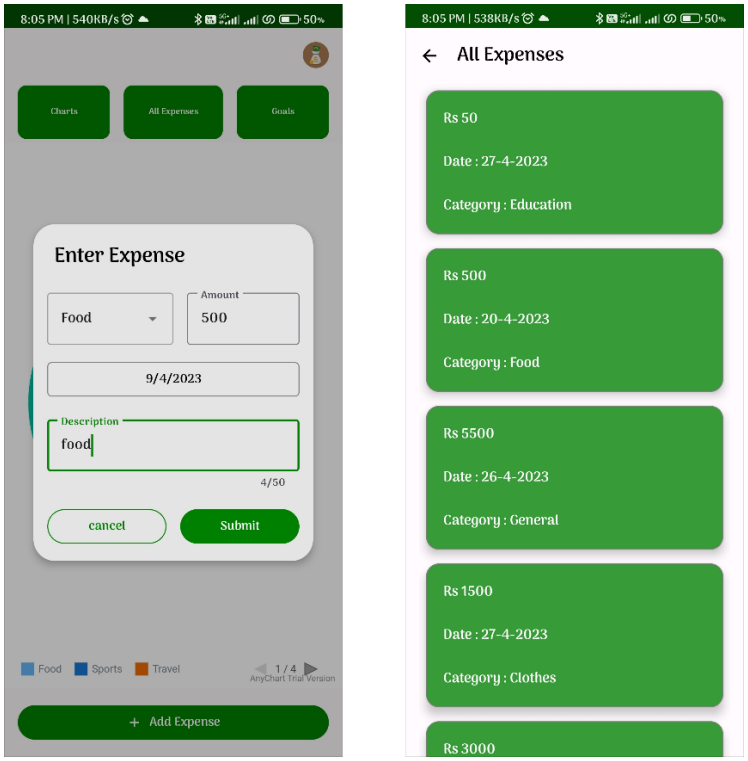

Figure 6: Add and View Expenses

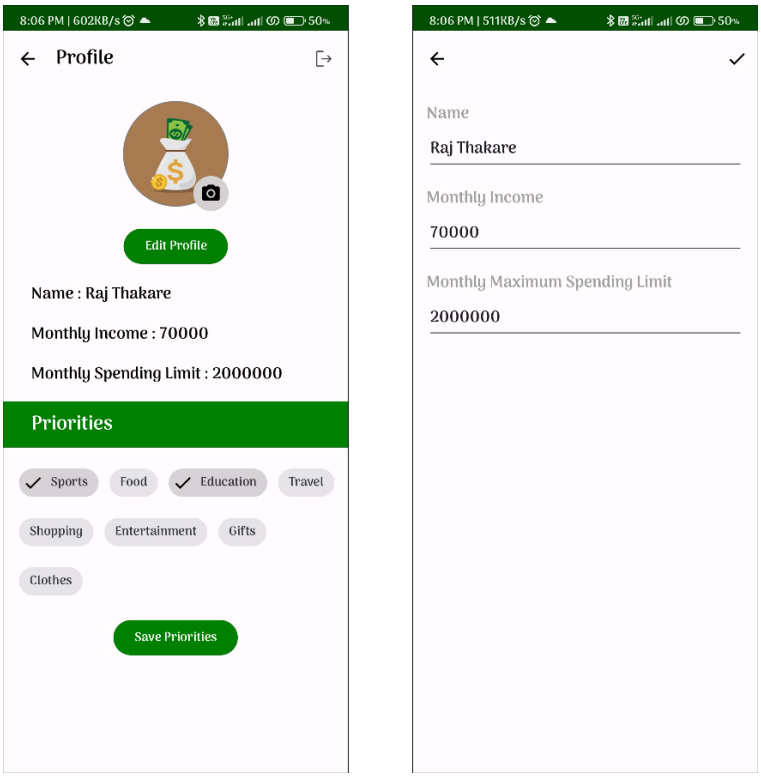

Figure 7: View and Edit Profile

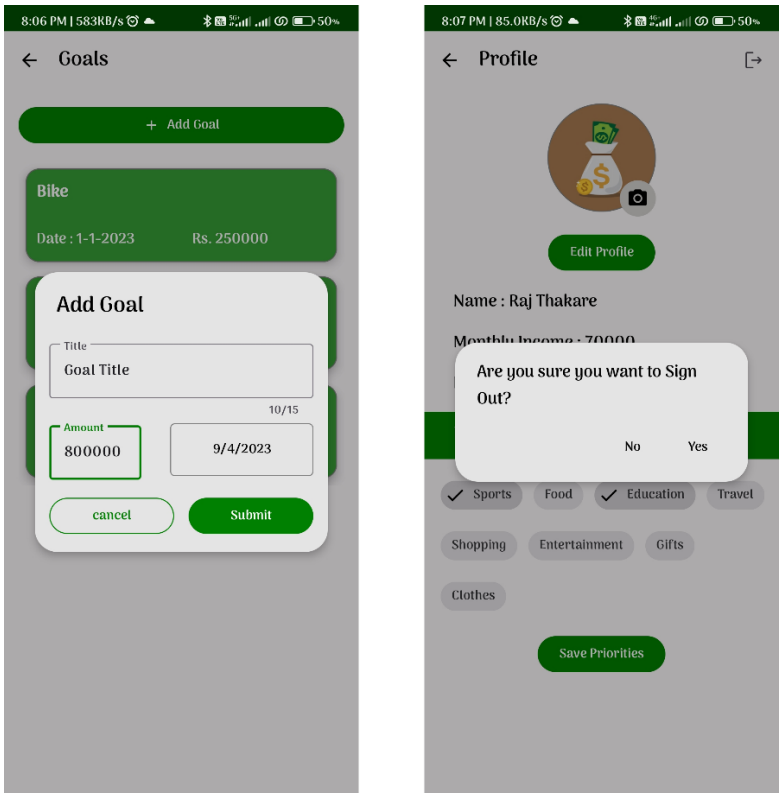

Figure 8: Add Goal and Sign Out

9. CONCLUSION

This study proposed an Android application that helps users track their daily expenses. The application allowed users to manually enter their expenses or automatically detect and read expense messages from the bank with the help of NB (Naïve-Bayes) algorithm. The NB (Naïve-Bayes) algorithm was trained using supervised machine learning to classify SMS as "Bank SMS" or "Normal SMS" accurately. The algorithm's performance was evaluated using precision and accuracy, which were found to be 97.67% and 99.33% respectively. The performance of Naive Bayes was also compared with other machine learning algorithms, and Naive Bayes was found to outperform all the other algorithms. The application was built using Kotlin and XML languages in Android Studio and used Firebase as the online database. The proposed system can help people manage their finances better and make informed decisions about their spending habits. Overall, the system has the potential to help people become more financially aware and responsible.

REFERENCES

Agarwal, K., & Kumar, T. (2018). Email Spam Detection using integrated approach of Naïve Bayes and Particle Swarm Optimization. *Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018).*

Akanda, W. and Uddin, A. (2021), "Multi-label Bengali article classification using ML-KNN algorithm and Neural Network, *Proceedings of the International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD).*

Alom, Z., Carminati, B., & Ferrari, E. (2018). Detecting spam accounts on Twitter. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 856-863.

Azam, M., Ahmed, T., Sabah, F., & Hussain, M. I. (2018). Feature Extraction based Text Classification using K-Nearest Neighbor Algorithm. *IJCSNS International Journal of Computer Science and Network Security*, 18(12), 95. Manuscript received December 5, 2018; Manuscript revised December 20, 2018.

Bharath, G., Manikanta, K.J., Bhanu Prakash, G., Sumathi, R., & Chinnasamy, P. (2021). Detecting Fake News Using Machine Learning Algorithms. *Paper presented at the 2021 International Conference on Computer Communication and Informatics (ICCCI-2021)*, 27 – 29.

Bhuiyan, H., Ashiquzzaman, A., Juthi, T. I., Biswas, S., & Ara, J. (2018). A Survey of Existing E-Mail Spam Filtering Methods Considering Machine Learning Techniques, *Global Journals*.

Suneera, C. M. and Prakash, J. (2020). Performance Analysis of Machine Learning and Deep Learning Models for Text Classification.*2020 IEEE 17th India Council International Conference (INDICON)*, 1-5.

Chaurasia, N., Bharali, P., & Naresh, R. (2021). SMS Spam Detection using Supervised Learning. *Turkish Journal of Computer and Mathematics Education*, 12(11).

De Vries, V. (2020). Classification of Aviation Safety Reports using Machine Learning. *Aerospace Operations Safety Institute, Royal Dutch Aerospace Centre, Amsterdam, The Netherlands. Vincent.de.Vries@nlr.nl*. IEEE.

Deepa, R., & Lalwani, K. N. (2019). Image classification and text extraction using machine learning. *In Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology* [ICECA 2019] (pp. 218-222). IEEE Conference Record # 45616; IEEE Xplore ISBN: 978-1-7281-0167-5. Loyola ICAM College of Engineering and Technology, Chennai, India.

Deshmukh, N., Dhumal, V., Gavasane, N., & Jadhav, S. V. (2021). SPAM DETECTION BY USING KNN ALGORITHM TECHNIQUES. *International Journal of Advance Scientific Research and Engineering Trends*, 6(5), 128. doi:10.51319/2456-0774.2021.5.0029.

Gupta, M., Bakliwal, A., Agarwal, S., & Mehndiratta, P. (2018). A Comparative Study of Spam SMS Detection using Machine Learning Classifiers. *In Proceedings of 2018 Eleventh International Conference on Contemporary Computing (IC3)*, 1-5. Noida, India: IEEE.

Gupta, P., Dubey, R. K., & Mishra, S. (2019). Detecting Spam Emails/Sms Using Naive Bayes and Support Vector Machine. *International Journal of Scientific & Technology Research*, 8(11).

Katpatal, R. & Junnarkar, A. (2018). An Efficient Approach of Spam Detection in Twitter. *In Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018)*.

Krishnaveni, N., & Radha, V. (2021). Comparison of Naive Bayes and SVM classifiers for detection of spam SMS using natural language processing. *International Journal of Scientific Research in Computer Science and Engineering*, 9(2), 2260-2267. DOI: 10.21917/ijsc.2021.0323.

Kumar, N., Sonowal, S., & Yadav, N. (2020). Email Spam Detection Using Machine Learning Algorithms. *In Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA 2020).*

Kurniawan, S., & Budi, I. (2021). Indonesian Tweets Hate Speech Target Classification using Machine Learning. *Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia*. Retrieved May 21, 2021, from IEEE Xplore database.

Ma, T., Yamamori, K., & Thida, A. (2020). A comparative approach to Naive Bayes classifier and Support Vector Machine for email spam classification. *In 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE). IEE*E.

Moldagulova, A., & Sulaiman, R.B. (2017). Using KNN Algorithm for Classification of Textual Documents. *In 2017 8th International Conference on Information Technology (ICIT)*, 1-5. IEEE.

Murugavel, U. and Santhi, R. (2020). K-Nearest Neighbor Classification of E-Mail Messages for Spam Detection. *Indian Journal of Science and Technology*, 13(22). doi: 10.21917/ijsc.2020.0316.

Nagre, S. (2018). Mobile SMS Spam Detection using Machine Learning Techniques. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 5(12), 548. Retrieved from http://www.jetir.org (ISSN-2349-5162), 2018.

Nandhini, S. & Jeen Marseline, K. S. (2020). Performance Evaluation of Machine Learning Algorithms for Email Spam Detection. *Paper presented at the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE*). IEEE. ISBN: 978-1-7281-4142-8.

Navaney, P., Dubey, G., & Rana, A. (2018). SMS Spam Filtering using Supervised Machine Learning Algorithms. *IEEE*.

Pal, K., & Patel, B. V. (2020). Automatic Multiclass Document Classification of Hindi Poems using Machine Learning Techniques. *Paper presented at the International Conference for Emerging Technology (INCET),* Belgaum, India, June 5-7, 2020.

Phani Prasad, J., & Venkatesham, T. (2021). Classification of E-mail Spam with Supervised Machine Learning Naïve Bayesian Classification. *Advances and Applications in Mathematical Sciences*.

Bhatia, P. & Bhatia, K. (2018). Spam Detection using Naive Bayes Classifier. *International Journal of Computer Sciences and Engineering Open Access*, 6(7), 1-6, E-ISSN: 2347-2693.

Raza, M., Jayasinghe, N.D., & Muslam, M.M.A. (2021). A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms. *In 2021 International Conference on Information Networking (ICOIN) IEEE*, 1- 6.

Reddy, K.N. and Kakulapati, V. (2021). Classification of Spam Messages using Random Forest Algorithm. *Journal of Xidian University*.

Santoshi, K.U., Bhavya, S.S., Sri, Y.B., & Venkateswarlu, B. (2021). Twitter Spam Detection Using Naïve Bayes Classifier. *In Proceedings of the Sixth International Conference on Inventive Computation Technologies [ICICT 2021*], 223-227).

Selvapattu, P. and Patil, P.V. (2020). SMS Spam Detection. *IJISET - International Journal of Innovative Science, Engineering & Technology*, 7(6).

Sethi, P., Bhandari, V., & Kohli, B. (2017). SMS spam detection and comparison of various machine learning algorithms.

Shradha & Verma T. (2017). E-Mail Spam Detection and Classification Using SVM and Feature Extraction. *International Journal of Advance Research, Ideas and Innovations in Technology*, 3(3), 1491.

Shrivastava, A., & Dubey, R. (2018). Classification of Spam Mail using different machine learning algorithms. *Paper presented at the 2018 IEEE*.

Siddique, Z. B., Khan, M. A., Ud Din, I., Almogren, A., Mohiuddin, I., & Nazir, S. (2021). Machine Learning Based Detection of Spam Emails. *Scientific Programming*.

Singh, M., Pamula, R., & Shekhar, S. K. (2018). Email Spam Classification by Support Vector Machine. *In 2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 556-560.

Taloba, A. I., & Ismail, S. S. I. (2019). An Intelligent Hybrid Technique of Decision Tree and Genetic Algorithm for E-Mail Spam Detection. *In 2019 IEEE Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 1-6.

Tanvir, A.-A., Mahir, E.M., Akhter, S., & Huq, M.R. (2019). Detecting Fake News using Machine Learning and Deep Learning Algorithms. *In 2019 7th International Conference on Smart Computing & Communications (ICSCC)*.

Tope, M. (2019). Email Spam Detection using Naive Bayes Classifier. *International Journal of Scientific Development and Research (IJSDR*), 4(6), 1-3.

Vijay & Kumar S.(2021). Spam SMS Detection Using Naive Bayes Classifier. *International Journal of Scientific Research and Engineering Development*, 4(1), 561.

Mohammed, Z., Farhaz, M. J A, Irshad, M. P., Basthikodi, M. & Faizabadi, A. R. (2019).A Comparative Study for Spam Classifications in Email Using Naïve Bayes and SVM Algorithm. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 6(5).