MOVIE RECOMMENDATION SYSTEMS USING DECISION TREE AND COMPARE PREDICTION ACCURACY WITH NAIVE BAYES BASED COLLABORATIVE FILTERING

Chamala Karthik Reddy[1], Terrance Frederick Fernandez[2*]

Article History: Received: 12.12.2022 **Revised:** 29.01.2023 **Accepted:** 15.03.2023

.

Abstract

Aim: The aim of the research article is to improve the accuracy of movie recommendation systems using a novel Decision Tree (D-Tree) algorithm in comparison with a Naive Bayes (NB) algorithm.
Materials and methods: The dataset used in this paper was collected from the Movie lens database. The sample size for the movie recommendation system was sample 20 (Group 1 = 10 and Group 2 = 10) and the calculation was performed utilizing G-power 0.8 with alpha and beta qualities of 0.05 and 0.2 with a confidence interval of 95%. The movie recommendation system is performed by the Decision Tree (D-Tree) classifier with a number of samples (N=10) and Naive Bayes (NB) model with a number of samples (N=10).
Results: The Decision Tree (D-Tree) classifier has a 90.88 percent higher accuracy rate when compared to the accuracy rate of the Naive Bayes (NB) model, which is 82.56 percent. The study has a significance value of p=0.024. **Conclusion:** The Decision Tree (D-Tree) classifier provides better outcomes in terms of accuracy rate when compared to the Naive Bayes (NB) model for movie recommendation systems.

Keywords: Innovative Recommendation System, Movie Recommendation, Novel Decision Tree, Naive Bayes, Machine Learning.

[1]Research Scholar, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India, Pincode:602105.
[2*]Project Guide, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India, Pincode:602105.

1. Introduction

A movie recommendation is significant in our social lives since it has the ability to provide greater enjoyment (Wang et al. 2017). Users can recommend a set of movies based on their interests or the attractiveness of the films. There are numerous recommendation systems available on the market for recommending various things to users (Shah et al. 2017). Durability, input separability, overspecialization, and cold-start issues plague these systems, leading in poor quality recommendations and limited coverage. A novel Decision Tree (D-Tree) classifier and the Naive Bayes algorithm are described in this study as part of the suggested methodology, with comparison findings demonstrating that the proposed D-Tree improves the accuracy, quality, and adaptability of the movie recommendation system (Jha, Gaur, and Thakur 2022). The proposed D-Tree method appears to be promising for solving very large-scale issues, and high-quality recommendations can be anticipated as a result of the experiments. Many real-time applications, such as e-commerce and marketing, use recommendation programmes; they are also used in many other areas, such as healthcare system, aviation, and farming (Mustaqeem, Anwar, and Majid 2020; Archana and Saranya 2020).

Over the last few decades, a plethora of recommendation systems have been developed. Different methodologies are used in these systems, such as collaborative approaches, content-based approaches, utility-based approaches, hybrid approaches, and so on. There are 135 research publications on IEEE Xplore, and 170 articles on Google Scholar. Hirdesh Shivhare et al. (Gupta, Shivhare, and Sharma 2015) developed an integrative strategy for building a movie recommendation system by combining fuzzy c-means clustering and a genetic algorithm-based weighted similarity measure. MovieGEN, an expert system for movie recommendation, was introduced by Gaurangi Tilak et al. (Eyrun, Gaurangi, and Nan 2008). On the basic principle of a hybrid recommendation method, they constructed the system utilizing machine learning and clustering algorithms. By integrating a Naive Bayes classification strategy with a collaborative filtering recommendation approach, writers (Ghazanfar and Prugel-Bennett 2010) suggested a novel switching hybrid recommendation approach. Ahmed et al. (Ahmed et al. 2017) showed how movie suggestions might be difficult and different from other types of recommendations. Park, Hong, and Cho (Park, Hong, and Cho 2007) suggested a personalized recommendation system in which Bayesian Networks reflect users' preferences. The variables are determined from a dataset, however the Bayesian Network topology was created by an analyst. K Jakhar created a collaborative filtering system that is used in conjunction with a Support Vector Machine based classifier to identify people with similar interests and model their likes and dislikes more correctly than collaborative filtering alone (Jakhar, Sharma, and Sharma 2016). S Ryu and K Han et al. (Ryu et al. 2010) suggested an agent-based recommendation model that can cut down on analysis time when new users or services enter the system and offer more user-centric services. On a large sample of the MovieLens dataset, a Gershman et al. (Gershman et al. 2010) suggested decision tree based recommendation system outperformed the quality of recommendations given mostly by the extracted features separation criteria.Our team has extensive knowledge and research experience that has translated into high quality publications(Mohan et al. 2022; Vivek et al. 2022; Sathish et al. 2022; Kotteeswaran et al. 2022; Yaashikaa, Keerthana Devi, and Senthil Kumar 2022; Yaashikaa, Senthil Kumar, and Karishma 2022; Saravanan et al. 2022; Jayabal et al. 2022; Krishnan et al. 2022; Jayakodi et al. 2022)

The key drawback of the current method is that the recommendation will be incorrect if the content is insufficient to discriminate between two things with the same set of qualities. When data is scant, as is common with web-related items, its accuracy suffers (V et al. 2020). To address this issue, this research suggested a novel Decision Tree (D-Tree) algorithm that improves prediction accuracy and addresses the issue of cold-start items. The Naive Bayes (NB) method is used to compare the findings. The proposed methodology aims to improve the movie recommender system's functionality and quality. Even with a sparse dataset, the proposed system outperforms the competition.

2. Materials and Methods

The study was carried out at Saveetha School of Engineering. The segmentation dataset was collected from the Movie Lens repository. This research uses two different methods: the new Decision Tree and the Naive Bayes approach. It entails two sample sets of ten samples each and a total of twenty samples, with a pretest power of 0.8. The test size for training the D-TREE is around 20% of the whole dataset, with the remaining 80% utilized for the training dataset. For the movie recommendation system, Python software is used to obtain the results. Using earlier results from (Salehi and Nakhai Kamalabadi 2013) at clinicalc.com, the sample size was calculated with a threshold of 0.05, G power of 80%, and confidence interval of 95%.

Naïve Bayes

It is a Bayes' Theorem-based approach. According to the Naive Bayes algorithm, the presence of one characteristic in a group has no influence on the availability of other features in that class. This framework is simple to build and is very useful when dealing with large datasets. Because of its simplicity, Naive Bayes is known to perform better even with the most powerful classification algorithms.

$$P\ C|A\ = P(A|C) * \frac{P(C)}{P(A)} \qquad (1)$$

Pseudocode

Step 1: Input: movie recommendation system_Input Features
Step 2: Assign training and testing dataset for movie recommendation system
Step 3: Output: Classification on movie recommendation system
Step 4: Function: Naïve Bayes (Input features I)
Step 5: Go over the Training data.
Step 6: For each class, compute the mean and standard deviation of the predictor variables.
Step 7: In each class, use the gauss density formula to compute the probability of p i.
Step 8: Estimate the likelihood for each class until all predictor variables (p 1,p 2,...p n) have been calculated.
Step 9: Determine the probability for each class.
Step 10: Maximize your chances.
Step 11: Return Classification outcomes of movie recommendation system

Decision Tree

A decision tree is a branch in which each inner (non-leaf) node represents an attribute test, each branch indicates the test's outcome, and each terminal (leaf) node provides a classification model (Choi et al. 2012). Based on the qualities of the input, the decision tree creates a predictive model that maps the information to a predicted value. Each arc from a parent to a child node indicates a possible value or collection of values for that attribute, and each internal node in the tree correlates to an attribute. The tree is built from the ground up, starting with the root of the tree and the data set. The root is given an attribute, and curves and sub-nodes are produced for each set of values. The information set is then separated into values, with each child node receiving only the amount of the training dataset that matches to the feature value given by the child node's path. Using decision trees to build recommendation models has a number of advantages, including speed, understandability, and adaptability in analyzing different types of input data sources.

Pseudocode

Step 1: Input: movie recommendation system_Input Features
Step 2: Assign training and testing dataset for movie recommendation system
Step 3: Output: Classification on movie recommendation system
Step 4: Function: Decision Tree (Input features I)
Step 5: if tree is of the form Leaf(movie_detect) then
Step 6: return detect
Step 7: else if tree is of the form Node(Input features I,left,right) then
Step 8: if Input features I = no in test point then
Step 9: return D-Tree_Test(left,test point)
Step 10: else
Step 11: return D-Tree_Test(right,test point)
Step 12: end if
Step 13: end if
Step 14: Return Classification outcomes for movie recommendation system

Statistical Analysis

Python software is used to generate the results (Downey, n.d.). A monitor with a resolution of 1024x768 pixels was required to train these datasets (10th gen, i5, 12GB RAM, 500 GB HDD). NB and D-TREE algorithms are statistically analyzed using SPSS software (Yockey 2017). The mean, standard deviation, and standard error mean statistical significance between the groups were determined using the independent sample t test, followed by a comparison of the two groups using SPSS software. Accuracy is a dependent variable, while D-TREE and NB are independent variables.

3. Results

The accuracy rate of the D-TREE classifier is compared to that of the NB classifier in Figure 1. The D-TREE classifier has a higher accuracy rate of 90.88 when compared to the NB classifier, which has 82.56. The D-TREE classifier is significantly different from the NB classifier ($p<0.05$ independent sample test). On the X-axis, D-TREE and NB accuracy rates are plotted. Y-axis: Mean accuracy rate for keyword identification, ± 1 SD with 95 percent confidence interval. Table 1 presents the evaluation metrics of the comparison of the D-TREE classifier with the NB classifier. The D-TREE classifier has a 90.88 accuracy rate, whereas the NB classifier has 82.56, respectively. In all parameters, the D-TREE classifier outperforms the NB in the recommendation of movies, with a higher accuracy rate. Table 2 displays the statistical computations for the D-TREE and NB classifier, such as mean, standard deviation, and standard error mean. In the t-test, the accuracy rate parameter is used. The D-TREE classifier has a mean accuracy rate of 90.88, while the NB classifier has 82.56, respectively. The standard deviation of D-TREE is 0.50393 and the NB algorithm is 1.67839. The standard error mean of D-TREE is 0.49248 and the NB algorithm is 1.78293. Table 3 shows the statistical computations for independent samples of D-TREE compared to the NB classifier. The significance value for accuracy rate is 0.024. An independent sample T-test is applied for comparison of D-TREE and NB algorithms with a confidence interval as 95% and level of significance as 0.77838. This independent sample test consists of significance as 0.001, significance (2-tailed), mean difference, standard error difference, and lower and upper interval difference.

4. Discussion

A novel Decision Tree and Naïve Bayes algorithms are implemented, and its output accuracy is analyzed and compared. The performance evaluation and comparison of proposed D-Tree and conventional NB algorithms for movie recommender systems have been carried out under various conditions to study their capability and convergence speed. From the results of this study, it is proved that D-Tree has performed better than the NB recommended system. The D-Tree has an accuracy of 90.88% whereas the NB has an accuracy of 82.56%. Over the last few years, numerous recommendation systems have been presented. S Agrawal (Agrawal and Jain 2017) presented an enhanced strategy for movie recommendation using a support vector machine, which has a 92 percent accuracy. MovieGEN, an optimization approach for movie recommendation, was introduced by Jain et al. (Jain et al. 2018). On the premise of a hybrid recommendation method, they constructed the system utilizing machine learning and cluster analysis. M. Kumar et al. (Kumar et al. 2015) evaluated four feature selection approaches (MI, IG, CHI, and DF) as well as five methods of learning (D-Tree, K-nearest neighbor, winnows classifier, Naive Bayes, and SVM) on a movielens database with a 90.4 percent accuracy. Jian Wei1 et al. (Wei et al. 2017) suggested a decision tree algorithm-based technique. Entire cold start difficulties (where no rating data is given) and incomplete cold start difficulties can both be solved using the proposed method. Only for total cold start problems did this strategy perform better, with an exactness of 89.8%. The developed D-Tree approach has the disadvantage of requiring more memory than the existing Naive Bayes strategy. It will be important to focus on the recommended approach's memory consumption in the future. To test the proposed approach, only distinct movie lens datasets were employed. It can also be combined with the Film Affinity and Netflix libraries, with the results coming later.

5. Conclusion

The proposed model exhibits the Decision Tree and Naïve Bayes algorithms, in which the D-TREE algorithm has the highest values. The accuracy rate of D-TREE algorithm is 90.88% higher compared with the NB algorithm, which has an accuracy rate of 82.56%. The accuracy rate of the D-TREE algorithm is efficient when compared with the NB algorithm, which has lower values in the movie recommendation system.

Declarations

Conflict of Interests

No conflict of interest in this manuscript.

Authors Contributions

Author name was involved in data collection, data analysis, manuscript writing. Author guide name was involved in conceptualization, data validation, and critical review of the manuscript.

Acknowledgment

The authors would like to express their gratitude towards Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Formerly known as Saveetha University) for providing the necessary infrastructure to carry out this work successfully.

Funding

We thank the following organizations for providing financial support that enabled us to complete the study.

1. Sankar Industries
2. Saveetha University
3. Saveetha Institute of Medical and Technical Sciences.
4. Saveetha School of Engineering.

6. References

Agrawal, Shreya, and Pooja Jain. 2017. "An Improved Approach for Movie Recommendation System." In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 336–42. ieeexplore.ieee.org.

Ahmed, Muyeed, Anirudha Paul, Mir Tahsin Imtiaz, Md Zahid Hassan, Shawon Ashraf, and Rashedur M. Rahman. 2017. "TV Series Recommendation Using Fuzzy Inference System, K-Means Clustering and Adaptive Neuro Fuzzy Inference System." In 2017 13th

International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 1540–47.

Archana, K., and K. G. Saranya. 2020. "Crop Yield Prediction, Forecasting and Fertilizer Recommendation Using Voting Based Ensemble Classifier." SSRG Int. J. Comput. Sci. Eng 7. https://www.academia.edu/download/63354060/IJCSE-V7I5P101_120200518-129172-1avb6kf.pdf.

Choi, Keunho, Donghee Yoo, Gunwoo Kim, and Yongmoo Suh. 2012. "A Hybrid Online-Product Recommendation System: Combining Implicit Rating-Based Collaborative Filtering and Sequential Pattern Analysis." Electronic Commerce Research and Applications 11 (4): 309–17.

Downey, Allen B. n.d. "Python for Software Design." Python for Software Design. https://doi.org/10.1017/cbo9780511813832.002.

Eyrun, A., T. Gaurangi, and L. Nan. 2008. "Moviegen: A Movie Recommendation System." Hewlett-Packard, August.

Gershman, Amir, Amnon Meisels, Karl-Heinz Lüke, Lior Rokach, Alon Schclar, and Arnon Sturm. 2010. "A Decision Tree Based Recommender System." 10th International Conferenceon Innovative Internet Community Systems (I2CS)--Jubilee Edition 2010--. https://dl.gi.de/handle/20.500.12116/19012.

Ghazanfar, Mustansar, and Adam Prugel-Bennett. 2010. "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering." https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.302.5301&rep=rep1&type=pdf.

Gupta, Anshul, Hirdesh Shivhare, and Shalki Sharma. 2015. "Recommender System Using Fuzzy c-Means Clustering and Genetic Algorithm Based Weighted Similarity Measure." In 2015 International Conference on Computer, Communication and Control (IC4), 1–8.

Jain, Kartik Narendra, Vikrant Kumar, Praveen Kumar, and Tanupriya Choudhury. 2018. "Movie Recommendation System: Hybrid Information Filtering System." Intelligent Computing and Information and Communication, 677–86.

Jakhar, K., V. K. Sharma, and S. Sharma. 2016. "Collaborative Filtering Based Recommendation System Augmented with SVM Classifier." Int. J. Sci. Eng. Technol. https://www.academia.edu/download/56729542/document.pdf.

Jayabal, Ravikumar, Sekar Subramani, Damodharan Dillikannan, Yuvarajan Devarajan, Lakshmanan Thangavelu, Mukilarasan Nedunchezhiyan, Gopal Kaliyaperumal, and Melvin Victor De Poures. 2022. "Multi-Objective Optimization of Performance and Emission Characteristics of a CRDI Diesel Engine Fueled with Sapota Methyl Ester/diesel Blends." Energy. https://doi.org/10.1016/j.energy.2022.123709.

Jayakodi, Santhoshkumar, Rajeshkumar Shanmugam, Bader O. Almutairi, Mikhlid H. Almutairi, Shahid Mahboob, M. R. Kavipriya, Ramesh Gandusekar, Marcello Nicoletti, and Marimuthu Govindarajan. 2022. "Azadirachta Indica-Wrapped Copper Oxide Nanoparticles as a Novel Functional Material in Cardiomyocyte Cells: An Ecotoxicity Assessment on the Embryonic Development of Danio Rerio." Environmental Research 212 (Pt A): 113153.

Jha, Govind Kumar, Manish Gaur, and Hardeo Kumar Thakur. 2022. "A Trust-Worthy Approach to Recommend Movies for Communities." Multimedia Tools and Applications, 1–28.

Kotteeswaran, C., Indrajit Patra, Regonda Nagaraju, D. Sungeetha, Bapayya Naidu Kommula, Yousef Methkal Abd Algani, S. Murugavalli, and B. Kiran Bala. 2022. "Autonomous Detection of Malevolent Nodes Using Secure Heterogeneous Cluster Protocol." Computers and Electrical Engineering. https://doi.org/10.1016/j.compeleceng.2022.107902.

Krishnan, Anbarasu, Duraisami Dhamodharan, Thanigaivel Sundaram, Vickram Sundaram, and Hun-Soo Byun. 2022. "Computational Discovery of Novel Human LMTK3 Inhibitors by High Throughput Virtual Screening Using NCI Database." Korean Journal of Chemical Engineering. https://doi.org/10.1007/s11814-022-1120-5.

Kumar, M., D. K. Yadav, A. Singh, and V. K. Gupta. 2015. "A Movie Recommender System: Movrec." International Journal of. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.6037&rep=rep1&type=pdf.

Mohan, Kannan, Abirami Ramu Ganesan, P. N. Ezhilarasi, Kiran Kumar Kondamareddy, Durairaj Karthick Rajan, Palanivel Sathishkumar, Jayakumar Rajarajeswaran, and Lorenza Conterno. 2022. "Green and Eco-Friendly Approaches for the Extraction of Chitin and Chitosan: A Review." Carbohydrate Polymers 287 (July): 119349.

Mustaqeem, Anam, Syed Muhammad Anwar, and Muhammad Majid. 2020. "A Modular Cluster Based Collaborative Recommender System for Cardiac Patients." Artificial Intelligence in

Medicine 102 (January): 101761.

Park, Moon-Hee, Jin-Hyuk Hong, and Sung-Bae Cho. 2007. "Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices." In International Conference on Ubiquitous Intelligence and Computing, 1130–39. Springer.

Ryu, Sanghyun, Kang-Hak Han, Hyunsu Jang, and Young Ik Eom. 2010. "User Adaptive Recommendation Model by Using User Clustering Based on Decision Tree." In 2010 10th IEEE International Conference on Computer and Information Technology, 1346–51. ieeexplore.ieee.org.

Saravanan, A., P. Senthil Kumar, B. Ramesh, and S. Srinivasan. 2022. "Removal of Toxic Heavy Metals Using Genetically Engineered Microbes: Molecular Tools, Risk Assessment and Management Strategies." Chemosphere 298 (July): 134341.

Sathish, T., R. Saravanan, V. Vijayan, and S. Dinesh Kumar. 2022. "Investigations on Influences of MWCNT Composite Membranes in Oil Refineries Waste Water Treatment with Taguchi Route." Chemosphere 298 (July): 134265.

Shah, Kunal, Akshaykumar Salunke, Saurabh Dongare, and Kisandas Antala. 2017. "Recommender Systems: An Overview of Different Approaches to Recommendations." In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 1–4. ieeexplore.ieee.org.

V, Deepak, V. Deepak, S. John Justin Thangaraj, and M. Rajesh Khanna. 2020. "An Improved Early Detection Method of Autism Spectrum Anarchy Using Euclidean Method." 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). https://doi.org/10.1109/i-smac49090.2020.9243361.

Vivek, J., T. Maridurai, K. Anton Savio Lewise, R. Pandiyarajan, and K. Chandrasekaran. 2022. "Recast Layer Thickness and Residual Stress Analysis for EDD AA8011/h-BN/B4C Composites Using Cryogenically Treated SiC and CFRP Powder-Added Kerosene." Arabian Journal for Science and Engineering. https://doi.org/10.1007/s13369-022-06636-5.

Wang, Xibin, Fengji Luo, Chunyan Sang, Jun Zeng, and Sachio Hirokawa. 2017. "Personalized Movie Recommendation System Based on Support Vector Machine and Improved Particle Swarm Optimization." IEICE Transactions on Information and Systems 100 (2): 285–93.

Wei, Jian, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. "Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items." Expert Systems with Applications 69 (March): 29–39.

Yaashikaa, P. R., M. Keerthana Devi, and P. Senthil Kumar. 2022. "Algal Biofuels: Technological Perspective on Cultivation, Fuel Extraction and Engineering Genetic Pathway for Enhancing Productivity." Fuel. https://doi.org/10.1016/j.fuel.2022.123814.

Yaashikaa, P. R., P. Senthil Kumar, and S. Karishma. 2022. "Review on Biopolymers and Composites – Evolving Material as Adsorbents in Removal of Environmental Pollutants." Environmental Research. https://doi.org/10.1016/j.envres.2022.113114.

Yockey, Ronald D. 2017. SPSS® Demystified: A Simple Guide and Reference. Routledge.

Tables and Figures

Table 1. The evaluation metrics of the D-TREE classifier with the NB classifier has been calculated. The D-TREE classifier has a 90.88 accuracy rate, whereas the NB classifier has 82.56, respectively. In all parameters, the D-TREE classifier outperforms the NB in the classification of movie recommendation, with a higher accuracy rate.

SI.No.	**Test Size**	**ACCURACY RATE**	
		D-TREE	NB
1	Test1	88.23	80.10
2	Test2	88.54	80.23
3	Test3	88.36	80.19

4	Test4	89.34	80.92
5	Test5	89.12	81.92
6	Test6	90.56	81.01
7	Test7	90.35	81.85
8	Test8	90.36	82.28
9	Test9	90.45	82.58
10	Test10	90.54	82.34
Average Test Results		90.88	82.56

Table 2. The statistical calculation such as mean, standard deviation and standard error mean for D-TREE and NB algorithm. Accuracy rate parameter used in the t-test. The mean accuracy rate of D-TREE is 90.88% and the NB algorithm is 86.956%. The Standard Deviation of D-TREE is 0.50393 and the NB algorithm is 1.67839. The Standard Error mean of D-TREE is 0.49248 and NB algorithm is 1.78293.

Group		**N**	**Mean**	**Standard Deviation**	**Standard Error Mean**
Accuracy Rate	NB	10	82.56	1.67839	1.78293
	D-TREE	10	90.88	0.50393	0.49248

Table 3: The statistical calculations for independent samples tested between D-TREE and NB algorithm. The significance value for accuracy rate is 0.024. Independent samples T-test is applied for comparison of D-TREE and NB algorithm with the confidence interval as 95% and level of significance as 0.36838. This independent sample test consists of significance as 0.001, significance (2-tailed), mean difference, standard error difference, and lower and upper interval difference.

Group		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval (Lower)	95% Confidence Interval (Upper)
	Equal	6.07	0.02	14.77	14	.001	13.81037	0.36838	14.89062	14.21192

Accuracy Rate	variances assumed	4	4	2						
	Equal variances not assumed			10.010	10.919	.001	12.51033	0.31612	10.30911	10.38112

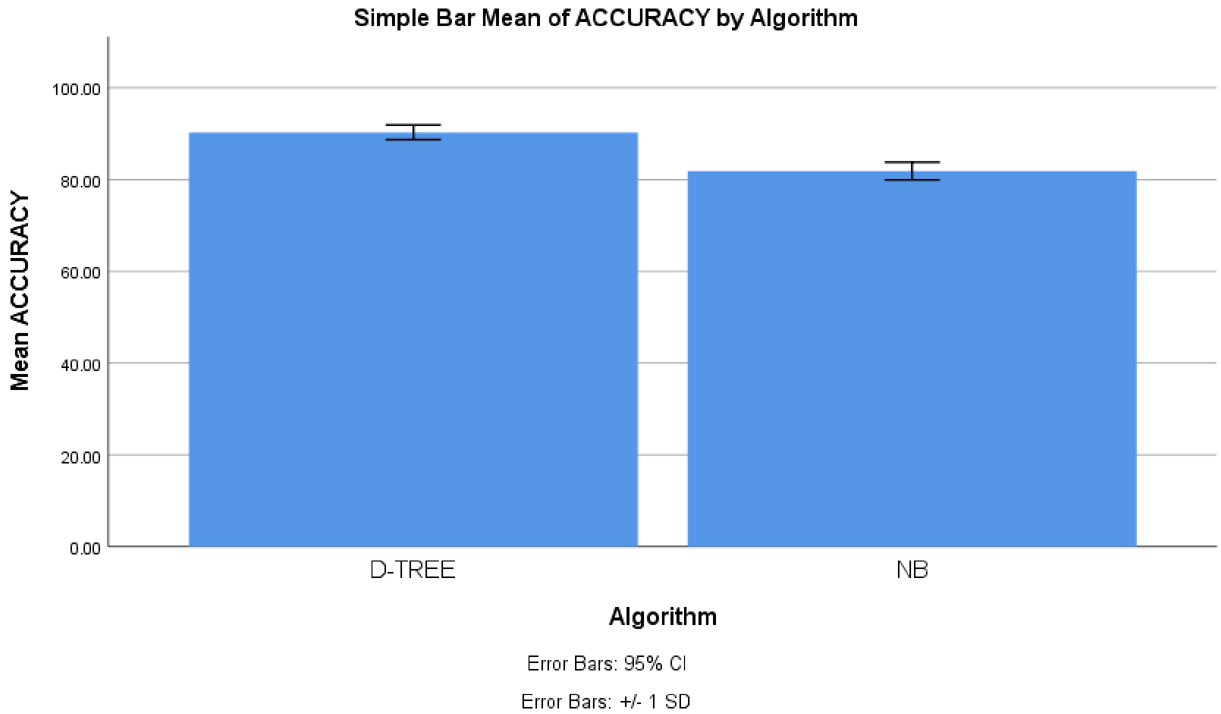

Fig 1. Simple Bar graph for D-TREE classifier accuracy rate is compared with NB model. The D-TREE classifier is higher in terms of accuracy rate 90.88 when compared with NB model 82.56. There is a significant difference between D-TREE classifier and NB model ($p<0.05$ Independent sample test). X-axis: NB model accuracy rate vs D-TREE classifier Y-axis: Mean of accuracy rate, for identification of keywords ± 1 SD with 95 % CI.