Check for updates

The Investigation of Feasibility Related to AI algorithms in VR for Improving Customer Satisfaction and Immersion

Yiren Shen

Business Analysis and Information Technology, Rutgers University, NJ 08854, USA
ys799@rutgers.edu

Abstract. After years of extensive development, the accessibility of Virtual Reality (VR) technology has reached a point where it is now within reach of ordinary individuals, allowing them to engage and derive pleasure from immersive experiences. This rapid proliferation of VR brands in the market has prompted businesses to explore avenues for enhancing customer immersion in order to elevate overall satisfaction. Consequently, this scholarly article investigates the feasibility of employing various Artificial Intelligence (AI) algorithms, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), eXtreme Gradient Boosting (XGBoost), Deep Neural Networks (DNN), Naive Bayes, and Random Forest, to analyze and improve VR applications. By evaluating crucial metrics such as F1-score, Recall, Precision, and accuracy for each model, the findings of this study reveal that the accuracy of these algorithms consistently hovered around 0.21. Notably, XGBoost analysis was supplemented with a feature importance table, which identified duration, age, and motion sickness as the primary influencing factors. Furthermore, the Learning Curve analysis demonstrated that KNN and Random Forest models exhibited signs of overfitting, whereas Naive Bayes and SVM models exhibited signs of underfitting. In light of these results, it is apparent that none of the individual AI models explored in this research are well-suited for the comprehensive analysis of VR applications.

Keywords: Machine Learning, Virtual Reality, Deep Learning

1 Introduction

Virtual Reality (VR) integrates immersive experiences into all aspects of everyday applications, and currently, in addition to the familiar VR games, VR is also widely recognized for its applications in education and training, and psychotherapy [1]. VR products are recently moving toward being thinner, lighter, more comfortable, easier to wear, lower power consumption, and easier to operate, where users are also expecting more practical devices with better user experience to emerge. VR creates an immersive and realistic experience through a series of technological means to create virtual scenes and simulate various sensory inputs to trick the brain, which allows people to immerse themselves and interact with the virtual world. [2]. In this scenario,

© The Author(s) 2023
P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_26

the user's immersion level is important but difficult to quantify. How to predict the degree of user experience based on various aspects of the device's parameters is critical and conducive to product improvement, and is therefore considered in this paper.

Artificial intelligence is an interdisciplinary scientific and technological domain dedicated to the study of computational systems capable of simulating and executing intelligent tasks analogous to human cognition. During the middle of the 20th century, machine learning, an important subset of AI, arose, enabling computers to learn and perform better via data-driven techniques. It allows computers to learn from data and improve performance automatically by building mathematical models and algorithms. Supervised learning is a prevalent machine learning technique, which, given input samples and corresponding labels (i.e., known outputs), allows a computer to learn a model from them that can be used to predict or classify new inputs [3, 4]. There are many common supervised learning algorithms, namely Support Vector Machines (SVM), Random forest and K-Nearest Neighbor (KNN). These can facilitate in predicting numerous issues in daily lives. For instance, since the Naive Bayes classification algorithm can be applied to categorize disease datasets like diabetes and heart disease, AI can calculate and compare disease data from several methods to decide whether a patient has the disease [5]. AI can also be used in the gaming field to accurately assess player levels for game developers, with the main purpose of bringing a better gaming experience and controlling the level of difficulty. Through AI's matching mechanism, game companies can accurately assess player level, match teammates and opponents of similar level, ensure fairness in matchmaking, and improve players' gaming experience [6]. On top of level matching, players may also have corresponding needs for social interaction. By analyzing players' user profiles through AI algorithms and then matching them accordingly, players' higher-level pursuits can be satisfied [5]. As a result, AI's algorithm can not only assist individuals in analyzing hospital testing data, but it can also boost player pleasure in the game, bringing income to the company. However, when considering the integration of AI algorithms within the domain of VR, it becomes imperative to undertake a comprehensive investigation into the applicability and effectiveness of these algorithms in predicting and facilitating analytical processes within this context.

To investigate the possibility of using AI algorithm in predicting the level of user experience when using VR devices, a Kaggle dataset was used in this study for analysis. Among others, KNN, SVM, Extreme Gradient Boosting (XGBoost), Naive Bayes, Random Forest, and Deep Neural Network (DNN) algorithms were evaluated for this dataset. Furthermore, the feature importance of the dataset is also investigated based on XGBoost and Random forest.

2 Method

2.1 Dataset Preparation

In the project, the Virtual Reality Experiences dataset from Kaggle was employed [7]. A dataset of user experiences in VR settings make up the dataset. It contains information about user preferences, emotional states, and physiological reactions like heart rate and skin conductance etc. The dependent variable in the dataset represents the subjective level of immersion reported by the user progressively increasing immersion with a scale of 1 to 5, which quantifies the user's immersion in the VR experience. The original dataset recorded the experience of 1, 000 users, a total of 1000 rows by 7 columns. In terms of the pre-processsing part, several steps were undertaken. Firstly, categorical variables, namely 'Gender' and 'VRHeadset,' were transformed into binary variables utilizing one-hot encoding. The data set is then split into independent features (x) and target variables (y) to aid in dividing it into training and testing sets.

2.2 Machine Learning Models

Various classifiers including KNN, SVM, XGBoost, DNN, Naive Bayes, and Random Forest are used to train on the data. On the test set, each model's performance is assessed. Confusion matrices are also generated for each model. For 'XGBoost', the feature importance is also plotted. The results show classification reports of different models, including precision, recall, f1-score, and accuracy for each class (0, 1, 2, 3, 4) as well as overall. Learning curves are plotted for each model. These charts demonstrate how the size of the training set affects the model's properties on the training and validation sets, which can facilitate in understanding whether the model is overfitting or underfitting.

K-Nearest Neighbors. KNN is the process of determining the k closest data to a newly given data, where whichever of these k data contains the most data in a category is regarded to belong to that category [8].

Support Vector Machine. SVM separates different classes of data samples by finding an optimal hyperplane or nonlinear transformation [9].

Extreme Gradient Boosting. The multi-threaded implementation of regression trees in XGBoost is built on the C++ programming language, building on the foundation of the original gradient boosting approach to significantly speed up model training and improve prediction accuracy [10].

Naive Bayes. It based on a statistical approach to classification prediction by means of prior probabilities and conditional probabilities. Its advantages are that the model is simple to use, fast, efficient on large data sets, and effective in handling missing or incomplete data. The model in this study employed scikit-learn in Python to implement a Naive Bayes after selecting GaussianNB as the Gaussian distribution [11].

Random Forest. The primary premise of random forests is to first classify several decision trees and then train them using a randomly selected subset of features. Random forests then obtain final predictions by voting or averaging [12].

Deep Neural Network. DNN is an effective neural network model [13, 14], which can achieve the property of approximating an arbitrary function by adding an activation function with nonlinear characteristics between the connections of layer and layer neurons and increasing the number of layers of neurons. The argmax function is used to convert these probabilities to class labels [15].

3 Results and Discussion

3.1 The Performance of Various Models

From Table 1 and Fig. 1, it can be observed that KNN model has achieved the accuracy of 0.21, with the f1-scores ranging from 0.14 to 0.29 across different classes. This indicates that the model's performance is quite varied across classes and overall, not particularly strong. The 0-4 in the first column represent the five different categories in the immersion level, while the accuracy in the last row represents the accuracy of the individual models.

Table 1. The performance based on KNN model training.

	Precision	Recall	F1-score	Support
0	0.23	0.37	0.29	49
1	0.16	0.22	0.19	41
2	0.28	0.23	0.25	35
3	0.19	0.11	0.14	37
4	0.22	0.11	0.14	38
Accuracy			0.21	200

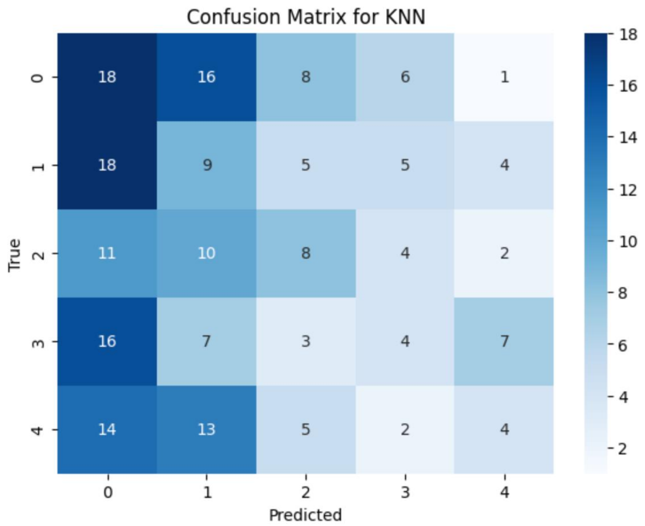

Fig. 1. The confusion matrix based on KNN (Photo/Picture credit: Original).

Table 2 and Fig. 2 shows that SVM model also has quite varied performance across classes, with f1-scores ranging from 0.14 to 0.25, and an overall accuracy of 0.20. This indicates that the model's predictive performance is below par.

Table 2. The performance based on SVM model training

	Precision	Recall	F1-score	Support
0	0.28	0.10	0.15	49
1	0.20	0.32	0.25	41
2	0.22	0.14	0.17	35
3	0.17	0.35	0.23	37
4	0.21	0.11	0.14	38
Accuracy			0.20	200

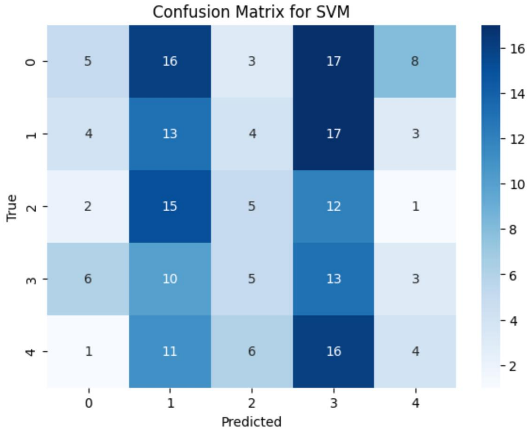

Fig. 2. The confusion matrix of SVM (Photo/Picture credit: Original).

Table 3 and Fig. 3 demonstrate that XGBoost algorithm exhibits comparable performance to the KNN and SVM models, as evidenced by f1-scores ranging from 0.15 to 0.25 across classes, accompanied by an overall accuracy of 0.215. The precision, recall, and f1-scores are all fairly similar across classes, thereby implying consistency but not marked accuracy in the model's predictions. From Fig. 4, it can be displayed that the feature importance, duration, age and motion sickness show the most influential features.

Table 3. The performance based on XGBoost model training

	Precision	Recall	F1-score	Support
0	0.26	0.18	0.21	49
1	0.29	0.10	0.15	41
2	0.25	0.26	0.25	35
3	0.17	0.32	0.22	37
4	0.16	0.18	0.17	38
Accuracy			0.20	200

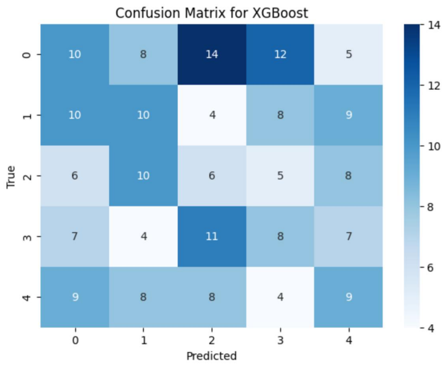

Fig. 3. The confusion matrix of XGBoost (Photo/Picture credit: Original).

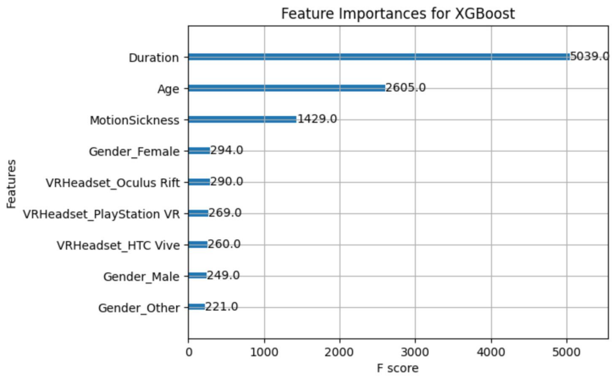

Fig. 4. The feature importance of XGBoost (Photo/Picture credit: Original).

Table 4 and Fig. 5 indicate the Naïve Bayes model exhibits an overall accuracy of 0.205, accompanied by f1-scores between 0.15 to 0.25. The recall values and precision are also fairly balanced across classes, suggesting a consistent performance across different classes, albeit without reaching notably elevated levels of accuracy.

Table 4. Naive Bayes model training

	Precision	Recall	F1-score	Support
0	0.26	0.18	0.21	49
1	0.29	0.10	0.15	41
2	0.25	0.26	0.25	35
3	0.17	0.32	0.22	37
4	0.16	0.18	0.17	38
Accuracy			0.20	200

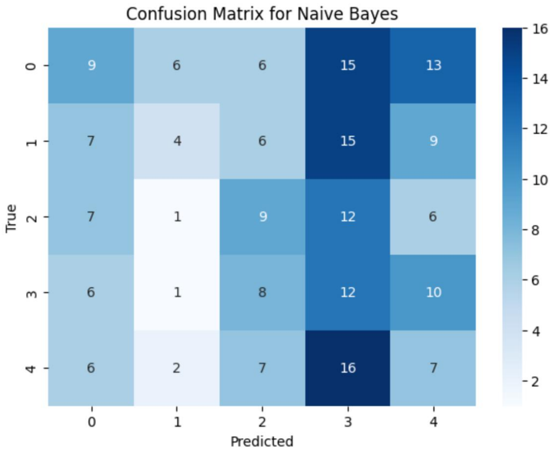

Fig. 5. The confusion matrix of Naïve Bayes (Photo/Picture credit: Original).

From Table 5 and Fig. 6, it illustrates that Random Forest model has an overall accuracy of 0.215, with f1-scores ranging from 0.14 to 0.28. This range of f1-scores suggests a performance that is comparable to the other models under consideration. Moreover, the observed consistency across classes implies that the model maintains a reliable pattern in its predictions. However, it is noteworthy that the model does not demonstrate notable accuracy beyond this consistency, suggesting a room for improvement in its predictive capabilities.

Table 5. Random Forest model training

	Precision	Recall	F1-score	Support
0	0.33	0.24	0.28	49
1	0.22	0.24	0.23	41
2	0.24	0.23	0.23	35
3	0.16	0.22	0.19	37
4	0.14	0.13	0.14	38
Accuracy			0.21	200

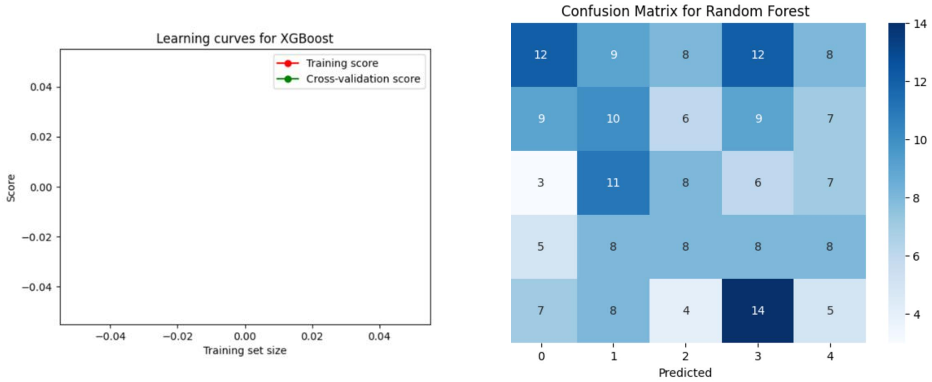

Fig. 6. The confusion matrix of Random Forest (Photo/Picture credit: Original).

The DNN model presents in Table 6 prove an atypical performance, characterized by an overall accuracy of 0.245, surpassing that of the other models examined. However, the model only correctly classifies instances of class 0, with a perfect recall of 1.00 for class 0 but 0.00 for the other classes. This indicates that the model is highly biased towards class 0 and fails to correctly classify instances of other classes. The argmax function is used to convert these probabilities to class labels.

Table 6. DNN model training

	Precision	Recall	F1-score	Support
0	0.24	1.00	0.39	49
1	0.00	0.00	0.00	41
2	0.00	0.00	0.00	35
3	0.00	0.00	0.00	37
4	0.00	0.00	0.00	38
Accuracy			0.24	200

3.2 The Learning Curve of the Model

Fig. 7 illustrates how increasing the training set affects the model's outcome on the two sets. This can facilitate understanding whether the model is overfitting or underfitting.

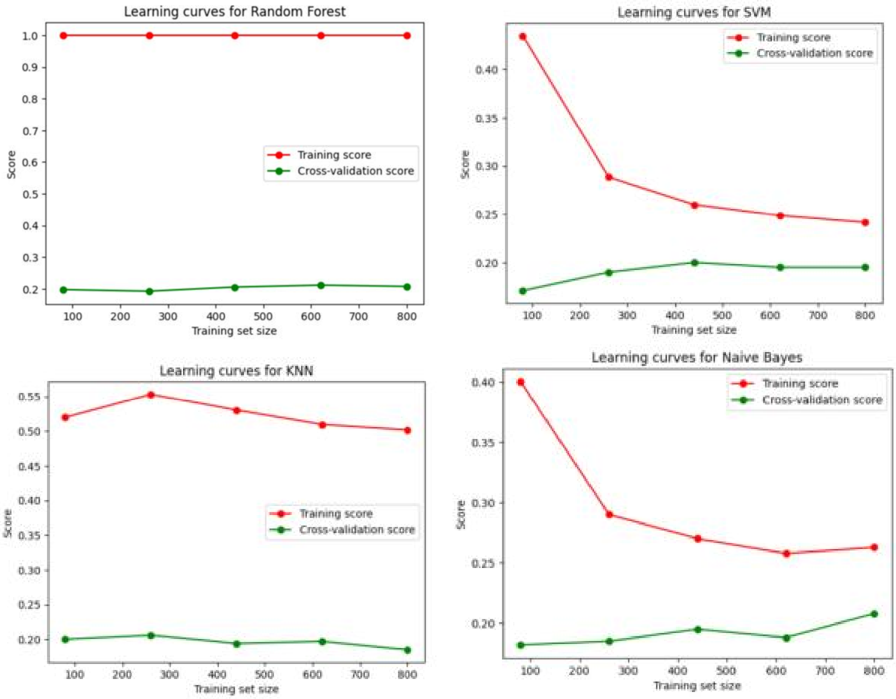

Fig. 7. Learning Curve of KNN, SVM, XGBoost, Naïve Bayes, Random Forest (Photo/Picture credit: Original).

The KNN model shows a training score that decreases slightly as more data is added. Given that the test result is consistently low and low across all training set sizes, it is possible that the model is overfit to the training set of data. The SVM model reveals a consistent decrease in training score as the size of the training data increases, indicating that the model is learning with more data. However, the test scores are quite low and remain more or less constant. This could be a sign of underfitting, indicating that the model might be too simple to capture the complexity in the data. Similar to SVM, the Naïve Bayes model training score are low and show minimal variation, which suggests the model might be underfitting the data. The overfitting pattern is reflected in the random forest model because the training data has a training score of 1.0 for all training set sizes. The test scores are slightly higher than the other models but are still quite low, reinforcing the idea that the model might be overfitting. The XGBoost model appears to have an issue as the training and test scores are showing as 'nan' (Not a Number). This might be due to a problem in the model fitting process or it might be an issue with the data used to train the model.

4 Conclusion

This paper endeavors to enhance VR design, user comfort, and customization by investigating the physical and emotional reactions of users in various VR scenarios. This study involves using KNN, SVM, XGBoost, DNN, Naive Bayes, and Random Forest algorithms to explore whether algorithms in AI can be applied to VR to improve customer satisfaction and immersion. The experimental results demonstrated that the accuracy of each model is not considerably high, and duration, age and motion sickness are the most influential features from the analysis of feature importance. Furthermore, the Learning curves show that some models are over-fitted and some models are under-fitted. Since VR is just now beginning to be offered on the market, it should be planned to change the algorithm in the future to locate models with a better rate of accurate analysis, which will help improve user immersion.

References

1. Girvan, C.: What is a virtual world? Definition and classification, Educational Technology Research and Development, 1087–1100 (2018).
2. Xiong, J., Hsiang, E. L., He, Z. et al.: Augmented reality and virtual reality displays: emerging technologies and future perspectives, Light Sci, 216 (2021).
3. Raschka, S., Patterson, J., Nolet, C.: Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence, Information 11, 193 (2020).
4. Ray, S.: A quick review of machine learning algorithms,2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 35-39 (2019).
5. Jackins, V., Vimal, S., Kaliappan, M., & Lee, M. W.: AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes, The Journal of Supercomputing, 77 (2020).
6. Tyagi, S., Sengupta, S.: Role of AI in Gaming and Simulation. In: Pandian, A., Palanisamy, R., Ntalianis, K. (eds) Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI - 2019). ICCBI 2019. Lecture Notes on Data Engineering and Communications Technologies, vol 49. Springer, Cham (2020).
7. Joshi, A.: Virtual Reality Experiences, Kaggle, 18 May 2023, www.kaggle.com/datasets/aakashjoshi123/virtual-reality-experiences?resource=download (2023).
8. Alfeilat, H. a. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Salman, H. E., & Prasath, V. B. S.: Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. Mary Ann Liebert, Inc., 7(4), 221–248 (2019).
9. Pisner, D., & Schnyer, D. M.: Support vector machine, In Elsevier eBooks (pp. 101–121). (2020).
10. Li, L. et al.: XGBoost Model and Its Application to Personal Credit Evaluation, IEEE Intelligent Systems, vol. 35, no. 3, pp. 52-61(2020).
11. Feng, J. Y.: An Implementation of Naive Bayes Classifier, 2018 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 301-306 (2018).
12. Rigatti, S. J.: Random Forest, Journal of Insurance Medicine, 47(1), 31–39 (2017).

13. Yu, Q., et al.: Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training. Biomedical Signal Processing and Control, 72: 103323 (2022).
14. Sahaai, M. B.: Brain tumor detection using DNN algorithm. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(11): 3338-3345 (2021).
15. Bau, D., Zhu, J., Strobelt, H., Lapedriza, A., Zhou, B., & Torralba, A.: Understanding the role of individual units in a deep neural network, Proceedings of the National Academy of Sciences of the United States of America, 117(48) (2020).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.