World Journal of Innovation and Modern Technology, Vol. 8, Issue 4, (Apr) 2025
ISSN 2682-5910
DOI: 10.53469/wjimt.2025.08(04).15

Machine Learning Algorithm-Based Dating Predictions

Yuxuan Wang, Ke Wang*, Chenxi Li

School of Computer and Software, Jincheng College, Sichuan University, Chengdu 611731, Sichuan, China
**Correspondence author*

Abstract: ***In this paper, we studied the relationship between frequent-flier miles obtained every year, the percentage of time spent playing video games, the number of ice cream liters consumed per week and the final friendship situation in a group of dating data, learned KNN and decision tree algorithm, and established a two-person model, and finally analyzed and validated the results. Experimental results show that KNN algorithm and decision tree algorithm are both simple and convenient classification algorithms.***

Keywords: Artificial Intelligence; K-nearest Neighbor Algorithm; Decision Tree Algorithm; Machine Learning.

1. INTRODUCTION

With the vigorous development of science and technology, artificial intelligence has also undergone continuous technological innovation, and many excellent algorithms have emerged. K-nearest neighbor algorithm and decision tree algorithm have long been proposed in the field of machine learning. Although they are not as excellent as many algorithms after them, and even have many shortcomings, their simple and effective characteristics enable us to grasp their ideas and understand the ideas and connotations of the algorithms, which facilitates our further exploration and opens the door to research in the field of artificial intelligence. The experimental platform used in this article is Anaconda, which is an open-source Python distribution that is compatible with Python libraries such as pandas, numpy, and sklearn, making it easy to write code. Tang et al. [1] conducted a qualitative analysis of regional housing supply-demand imbalances in the US using big data approaches. In healthcare research, Wang et al. [2] developed a cell atlas of the immune microenvironment in gastrointestinal cancers, with particular focus on dendritic cells. Wang [3] proposed Bayesian optimization methods for adaptive network reconfiguration in urban delivery systems, while Li [4] applied machine learning techniques to enhance adverse event monitoring in Phase IV chronic disease drug trials. Several studies have investigated AI-driven solutions for digital platforms. Li and Wang [5] created a deep learning-enhanced adaptive interface to improve accessibility in e-government platforms, whereas Yuan [6] developed transformer-based techniques for processing medical texts in legal documents. In e-commerce, Song [7] demonstrated how AI integration can optimize operational efficiency through user-centric internal tools. Smart city technologies have also seen significant advancements. Chen [8] employed geospatial neural networks to enhance urban development through location intelligence. Meanwhile, Wang [9] examined legal aspects of enterprise naming rights and prior rights restrictions. Gong et al. [10] optimized enterprise risk decision support systems using ensemble machine learning methods. Finally, Bohang et al. [11] presented an image steganalysis approach combining active learning with hyperparameter optimization, demonstrating improved performance in scientific applications.

2. INTRODUCTION TO KNN ALGORITHM

KNN, Also known as K-nearest neighbor algorithm or proximity algorithm, it is one of the fundamental algorithms in data mining. The KNN algorithm itself is simple and effective, with only a forward communication process and no learning process. It is a lazy learning classification algorithm in machine learning.

The core idea of the KNN algorithm is to select the k nearest values to the desired predicted point n in a sample space. The class represented by the maximum probability of the occurrence of the k values in that class is the class of the unknown point n and has all the characteristics of that class. The KNN algorithm belongs to the classification and logistic regression algorithm. By adding the candidate's characteristic information to the classification set, classification judgment can be made.

The main advantages of KNN algorithm are:

(1) Due to its early introduction and years of development, KNN has a very mature theory and a simple and effective algorithm approach, which can be used in both regression and classification problems.

(2) Can be used for non-linear classification.

(3) The training time complexity is relatively low.

(4) Without assumptions about the data, the KNN algorithm has higher accuracy compared to other machine learning algorithms and is quite insensitive to outliers.

(5) The KNN algorithm itself only involves calculating the distance from the test sample to the training sample, and the calculation itself is relatively simple.

The KNN algorithm, as an early algorithm in the field of artificial intelligence, has some drawbacks, such as:

(1) The computation time, also known as time complexity, is affected by the sample size. When there are a large number of features, the computational workload can be significant.

(2) When the sample is imbalanced, the accuracy of predicting rare categories is very low.

(3) KNN is a lazy learning algorithm that basically does not learn, resulting in slower prediction speed compared to other algorithms such as logic learning.

(4) Compared to decision tree models, KNN models have weak interpretability.

In the process of implementing the KNN algorithm, several core points need to be noted, such as the value of k. If the value of k is small, it is equivalent to only calculating a few nearest neighbor sample types. Imagine that in the most extreme case, when k=1, the final result will only be affected by the nearest sample type, which will result in significant errors. This indicates that if the value of k is too low, it can lead to overfitting problems. On the contrary, if the value of k is large, such as in extreme cases where k=n (where n is the number of samples), the result will be the sample type with the most samples in n, resulting in the problem of underfitting. To avoid overfitting or underfitting, the value of k is generally between 3 and 10. A common practice is to set k as the square root of the number of cases in the training set [2].

3. PREDICTING FRIENDSHIP CANDIDATES BASED ON KNN MODEL

3.1 Data Processing

This article uses a set of dating data with a sample size of 1000 rows, which is large and convenient for training and testing. The candidates mainly include the following three characteristics: the number of frequent flyer miles obtained each year, the percentage of time spent playing video games, and the liters of ice cream consumed per week.In the experiment, Pandas library functions were used for data reading. Pandas is a tool based on Numpy that provides a large number of functions and methods for quickly processing data, enabling us to better handle data analysis work. In this experiment, we can use the pandas. read_csv() function to read sample data with a suffix of csv.

Because the range of values for several features of the data in this article varies greatly, and the KNN algorithm measures distance, this can lead to significant differences in the updated results, which is not desirable. We would like to consider the importance level of features to be the same and not favor any particular feature, so it is necessary to preprocess the data before using these features. There are two ways to preprocess the data: normalization and normalization.

Standardization requires a mean of 0 and a standard deviation of 1. The conversion formula is as follows:

$$Z = \frac{x-\mu}{\sigma} \quad (1)$$

Among them,

$$\mu \frac{(x_1+x_2+\cdots+x_n)}{n} \tag{2}$$

$$\sigma = \sqrt{\frac{(x_1-\mu)^2+(x_2-\mu)^2+\cdots+(x_i-\mu)^2}{n}} \tag{3}$$

Normalization can compress the values of all features into the range of 0 to 1 after processing, which can suppress the influence of outliers on the results. The normalization formula is as follows:

$$X_{norm} = \frac{X-X_{min}}{X_{max}-X_{min}} \tag{4}$$

3.2 Use KNN algorithm to establish a model and evaluate it

Before establishing the model, it is necessary to choose a vector distance measurement rule. In the above data processing process, the data has been standardized/normalized, and the importance of features is the same. Therefore, Euclidean distance is applied as the distance measurement rule. The smaller the Euclidean distance, the greater the similarity.

In two-dimensional space, assuming that the coordinates of point A are (q1, p1), the coordinates of point B are (q2, p2), and so on, and the coordinates of the Nth point are (qn, pn), the formula for the Euclidean distance d of this set of vectors is:

$$d = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \tag{5}$$

In this experiment, since there is only one sample dataset, it can be divided into a training set and a testing set based on certain conditions. The training set is used to train the model, while the testing set is used to test the effectiveness of the model trained by the training set. The features of the testing set and the training set cannot be different, and the sample size must be sufficient to ensure that statistically significant results can be generated. Therefore, in the specific implementation of the experiment, 1000 data points were separated in a ratio of 2:8, with the first part of the data used as the testing set and the other data used as the training set for calculation, in order to obtain the experimental results.

Experiments typically use RMSE (Root Mean Square Error) to evaluate models.

The formula for RMSE is:

$$RMSE = \sqrt{\frac{(actual_1-predicted)_1^2+(actual_2-predicted)_2^2+\cdots+(actual_n-predicted)_n^2}{n}} \tag{6}$$

Root mean square error, also known as standard error, reflects the degree of dispersion of a dataset. The smaller the standard deviation, the more accurate the data and the better the model performance.

3.3 Analysis of Experimental Results

After running, the weighted average RMSE is 0.29232644, indicating that the model performs well. As the sample size increases, the standard deviation becomes closer to the overall standard deviation, and the standard error decreases with the increase of measurement times. Continuing to increase the sample size will be beneficial for improving the accuracy of recognition. In other words, if the sample size can be further increased, the model will have the possibility of finding a suitable fit.

This test only involved simple classification applications. The KNN algorithm has its own application space in both classification and regression. If more complex problems need to be analyzed in the future, further optimization of the algorithm is needed in terms of classification efficiency and effectiveness.

4. INTRODUCTION TO DECISION TREE ALGORITHM

The decision tree algorithm is a basic classification and regression algorithm that can transform uncertainty into certainty. It is a graphical method that intuitively utilizes probability analysis. Due to the fact that the graphics drawn by decision branches resemble a tree, it is called a decision tree. The decision tree algorithm progresses step by step from the root node to the leaf nodes (decisions), and all data can fall to the leaf nodes, which can be used for both classification and regression [4].

Decision trees have the following advantages:

(1) Easy to understand and implement, it eliminates the need for users to have a lot of background knowledge during the learning process. This is also its ability to directly reflect the characteristics of the data, and as long as it is explained, it has the ability to understand the meaning expressed by the decision tree.

(2) Almost no data preprocessing is required. Compared to the KNN algorithm mentioned earlier, decision tree data does not require preprocessing such as standardization/normalization or creating virtual variables.

(3) It can handle classification problems and has a very good effect on regression problems. For example, ID3 and C4.5 are classification algorithms, while CART is a regression algorithm.

(4) The decision tree uses a white box model, which is easier to use logical discrimination to reflect this rule compared to black box model algorithms (such as artificial neural networks).

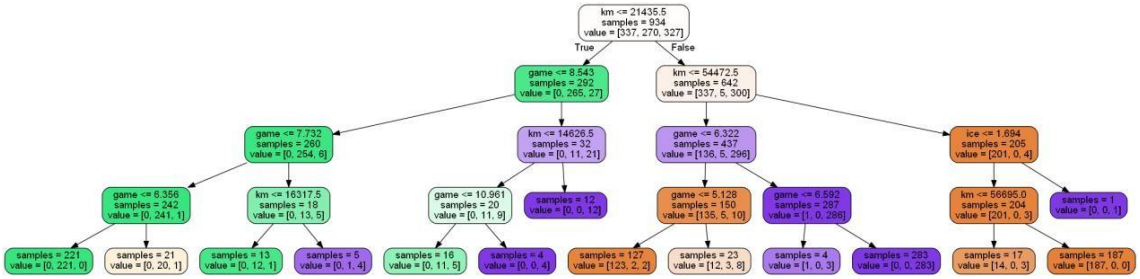

Figure 1: The final decision tree established

5. PREDICTING FRIENDSHIP CANDIDATES BASED ON DECISION TREE MODEL

5.1 Building a Decision Tree Model

In order to select features from the root node and construct a decision tree, a measurement criterion is needed to calculate the classification status after branch selection through different features. The decision tree is split using greedy thinking, usually selecting the optimal condition as the root node, and so on for the selection of other nodes.

This article uses the ID3 algorithm of decision trees, which is based on information theory and uses the information gain of entropy as the criterion for judgment. Entropy is a measure of uncertainty in a random variable, and the greater the uncertainty, the higher the entropy value obtained. The formula is:

$$H(X) = -\sum p_i \times log(p_i), i = 1,2,\cdots,n \tag{7}$$

Information gain refers to the change in data complexity of the root and splitting nodes before and after splitting

Decision trees cannot grow indefinitely, and there will always be times when they stop splitting. Imagine under extreme conditions, when the decision tree only has one data point left, the splitting will eventually stop, which undoubtedly leads to overfitting. So decision trees need pruning, generally there are two pruning methods: pre pruning and post pruning.

Pre pruning can limit depth, number of leaf nodes, number of leaf node samples, information gain, etc. While establishing a decision tree, pruning is performed. Post pruning is performed based on certain measurement standards, and only after the decision tree is established, will post pruning be performed.

This article uses the tree in the sklearn library. The DecisionTreeClassifier () function is used to build a decision tree and prune it using a limited depth method. The depth of the decision tree is limited to 4, and the code is: tr = tree.DecisionTreeClassifier(max_depth = 4,criterion = 'gini')

5.2 Analysis of Experimental Results

To demonstrate the training results, the original data was input again using the score function, resulting in an accuracy of 0.93617021276595747. From this, it can be concluded that the model performs well. Due to the use of ID3 decision tree in this experiment, there is a problem of information gain bias in selecting features with more values [6]. If C4.5 decision tree is used and information gain rate is used as the evaluation criterion, there is still room for improvement in model accuracy.

6. CONCLUSION

In summary, this article briefly implements the theoretical learning and practical application of k-nearest neighbor algorithm and decision tree algorithm based on Anaconda and Python languages. K-nearest neighbor algorithm and decision tree algorithm, as fundamental algorithms in the field of artificial intelligence, are the starting point for researching and learning more advanced algorithms. However, KNN algorithm and decision tree algorithm are still early algorithms in machine learning, and with the emergence of other algorithms, many of their inherent shortcomings have become more apparent. Although this article only roughly uses two algorithms through simple examples, there is still room for improvement in the algorithms in the examples. If we can continuously improve and perfect the two algorithms, we can have a deeper understanding of the ideas of machine learning and also benefit the learning of other algorithm ideas, thus further advancing research in the field of artificial intelligence.

REFERENCES

[1] Tang, Y., Zhao, S., & Yanjun, C. (2024). Regional Housing Supply and Demand Imbalance Qualitative Analysis in US based on Big Data.
[2] Wang, Y., Yang, T., Liang, H., & Deng, M. (2022). Cell atlas of the immune microenvironment in gastrointestinal cancers: Dendritic cells and beyond. Frontiers in Immunology, 13, 1007823.
[3] Wang, J. (2025). Bayesian Optimization for Adaptive Network Reconfiguration in Urban Delivery Systems.
[4] Li, T. (2025). Enhancing Adverse Event Monitoring and Management in Phase IV Chronic Disease Drug Trials: Applications of Machine Learning.
[5] LI, X., & Wang, Y. (2024). Deep learning-enhanced adaptive interface for improved accessibility in e-government platforms.
[6] Yuan, J. (2024, December). Efficient techniques for processing medical texts in legal documents using transformer architecture. In 2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC) (pp. 990-993). IEEE.
[7] Song, X. (2025). User-Centric Internal Tools in E-commerce: Enhancing Operational Efficiency Through AI Integration.
[8] Chen, J. (2025). Geospatial Neural Networks: Enhancing Smart City through Location Intelligence.
[9] Wang, H. (2024). The Restriction and Balance of Prior Rights on the Right of Enterprise Name.
[10] Gong, C., Lin, Y., Cao, J., & Wang, J. (2024, October). Research on Enterprise Risk Decision Support System Optimization based on Ensemble Machine Learning. In Proceeding of the 2024 5th International Conference on Computer Science and Management Technology (pp. 1003-1007).
[11] Bohang, L., Li, N., Yang, J. et al. Image steganalysis using active learning and hyperparameter optimization. Sci Rep 15, 7340 (2025). https://doi.org/10.1038/s41598-025-92082-w

Author Profile

Yuxuan Wang (1998-), male, Han, Chengdu, Sichuan Province, China, undergraduate degree, Jincheng College, Sichuan University, research direction: artificial intelligence.

Ke Wang (1985-), male, lecturer, master's student, Jincheng College, Sichuan University, research direction: cloud computing.

Chenxi Li **(**1998-7), male, Han, Guiyang City, Guizhou Province, bachelor's degree, Jincheng College, Sichuan University, research direction: big data technology development.