SN Computer Science (2024) 5:57
https://doi.org/10.1007/s42979-023-02375-y

ORIGINAL RESEARCH

Comparing Artificial Intelligence Classification Models to Improve an Image Comparison System with User Inputs

Sandra Jardim[2] · Jorge Valente[1] · Artur Almeida[1] · Carlos Mora[2]

Received: 20 March 2023 / Accepted: 29 September 2023

Abstract
Data science techniques have increased in popularity over the last decades due to its numerous applications when handling complex data, but also due to its high precision. In particular, Machine (ML) and Deep Learning (DL) systems have been explored in many unique applications, owing to their high precision, flexible customization, and strong adaptability. Our research focuses on a previously described image detection system and analyses the application of a user feedback system to improve the accuracy of the comparison formula. Due to the non-traditional requirements of our system, we intended to assess the performance of multiple AI techniques and find the most suitable model to analyze our data and implement possible improvements. The study focuses on a set of test data, using the test results collected for one particular image cluster. We researched some of the previous solutions on similar topics and compared multiple ML methods to find the most suitable model for our results. Artificial Neural networks and binary decision trees were among the better performing models tested. Reinforcement and Deep Learning methods could be the focus of future studies, once more varied data are collected, with bigger comparison weight diversity.

Keywords Artificial intelligence · Machine learning · Artificial neural networks · User feedback

Introduction

Artificial Intelligence (AI) has become an increasingly popular discipline of computer science. AI focuses on creating computer programs and algorithms capable of performing several data processing tasks [2]. Over the last few years, this area has increasingly attracted interest, both in the research community as well as in real-world applications. Amongst the several disciplines of AI, Machine learning uses techniques to automate the construction of analytical models, which are then used for processing a large range of data types [27]. It has been extensively applied to analyze large samples of data or to detect and establish patterns to predict information on new instances of related data [36]. Due to its rise in interest and recent developments, a big part of the population has had frequent interactions with modern AI techniques, often without realizing it [31]. Classical ML techniques have been successfully applied across different topics, research fields, and industries [27], one of which is recommendation systems. In fact, amongst the many applications of AI, decision-making systems can be extremely powerful tools for many data-driven scenarios [3].

While there are different ways to classify ML methods, we can consider the existence of two main classes: supervised and unsupervised learning methods. The main difference between the two is the presence of labels in the datasets:

– Supervised learning models are mainly used to determine predictive functions using labeled datasets for training. Each data object must include both the values of the independent variables as well as expected labels or output values. Using this data, this class of algorithms tries to identify the relationships occurring between the input and output values and generate a predictive model able to determine the result based only on the corresponding input data. Supervised learning methods are best suited for regression and data classification, being primarily used for a variety of algorithms like Linear Regression, Artificial Neural

✉ Sandra Jardim
sandra.jardim@ipt.pt

1 Techframe-Information Systems, SA, 2785-338 São Domingos de Rana, Portugal

2 Smart Cities Research Center, Polytechnic Institute of Tomar, 2300-313 Tomar, Portugal

Published online: 02 December 2023

Networks (ANN), Decision Trees (DT), Support Vector Machines (SVM), K-nearest Neighbors (KNN), Random Forest (RF), and others [36].

– Unsupervised learning models are typically used to solve problems in pattern recognition, based on unlabeled training datasets. This class of algorithms is able to classify the training data into a number of different categories, or classes, according to features based on dimensionality reduction and clustering techniques [19]. Some commonly employed algorithms include Principal Component Analysis (PCA) and K-Means [36]. Since the number of determined classes is unknown and the meaning of each one is unclear, unsupervised learning models are usually used for classification problems and for association mining.

Classification tasks usually involve matching a certain data object to a cluster of objects with similar characteristics. These can be represented as follows: Considering a set of input and output pairs $Z = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, we can construct a classifier function f that maps the input vector $x \in X$ onto distinct labels $y \in Y$. In binary classification tasks, like our study, the set of labels is simply $Y = \{-1, 1\}$.

Some of the advantages associated with ML, over conventional statistical methods like Logistic Regression, is that ML algorithms do not require the data to conform to the same statistical assumptions, such as independence of observations and the avoidance of multicollinearity of independent variables [32]. Models like Logistic Regression (LR) allow users to determine the relation of a binary (or dichotomous) outcome with one or more predictors, that may be either categorical or continuous. The independent variables are analyzed using the model, and a regression coefficient (usually denoted as "beta") and a "P" value are calculated for each of these. This P value is used to determine whether or not the particular variable contributes significantly to the occurrence of the outcome [26].

Support Vector Machines (SVM) are another frequently used ML model and a powerful classification tool. Some of its main advantages are its capacity to represent non-linear relationships and its ability to properly classify unseen data [12]. SVM operates by attempting to find the hyperplane that realizes the furthest margin of separation amongst the identified classes [7]. Some authors have identified possible advantages that SVM might present over Artificial Neural Networks (ANN), as they do not typically possess the same propensity for instability that Artificial Neural Networks might have, particularly with the effects of different random starting weights [4]. In general, a representation of the hyperplane solution used to classify a new sample x_i is:

$$f(x) = (w, x_i) + b, \tag{1}$$

where (w, x_i) is the dot-product between the weight vector w and the input sample, and b is a bias value. The value of each element of w can be viewed as a measure of the relative importance of each of the sample attributes for the classification of a sample.

Other recent applications have appeared, like the *CatBoost* library, a powerful ML model that uses binary decision trees as the base predictors, in a process known as a gradient boosting [13, 24]. Gradient boosting can be described as a process that uses an ensemble predictor to perform a gradient descent in a functional space [24]. Effectively, this model uses a group of decision trees, built consecutively during training. Each successive tree is built with reduced loss when compared to the previous trees. Such models have been studied for decades and are backed by convincing results that show how stronger predictors can be obtained by combining weaker models (base predictors) in a greedy manner [16]. This ML algorithm is a popular tool and has been shown to achieve promising results in a variety of practical tasks [13]. It was previously used as a learning method for complex problems, tackling heterogeneous features, noisy data, complex dependencies, recommendation systems, and weather forecasting, amongst many others [13, 29, 33, 34].

Artificial Neural Networks (ANN) are a class of neural models that consists of an input layer, an output layer and a set of hidden layers, performing two different stages. The first is called feed forward, where each hidden layer receives inputs from the previous nodes and a learning error is calculated based on the input layer and on the selected activation function [30]. Under the second stage, named feedback stage, the error is propagated backward to the input layer and the process suffers various iterations, until the correct result is achieved [22].

In addition to the previously mentioned classes of methods, Reinforcement Learning can also be regarded as another class of machine learning algorithms, which has been used in many different feedback-based systems. This class of algorithms refers to the generalization ability of a machine to correctly answer unlearned problems [36]. Reinforcement Learning consists of a set of techniques that use a mathematical framework to learn patterns and optimize control strategies directly from the data [25, 31], based on a reward function in a Markov Decision Process (MDP) [10, 28]. A MDP environment is typically defined through a balance of exploration with exploitation.

The behavior of the Markov Decision Process is usually determined by a reward function [31], whereby an agent is given a state and a reward from the environment, which is used to determine the appropriate action to take [17, 20, 28, 31]. Often, the success of Reinforcement Learning algorithms depends on a well-designed reward function [10]. This group of techniques has been used with tremendous theoretical and practical achievements in robotics control, gaming, autonomous driving, computer vision, and healthcare [20].

In this paper, we tested an assembly of several models, to find the best suited for our application and our set of data. The objective was to develop a system to automatically manage the vector weight distribution of several evaluation criteria, for an image retrieval system based on the user feedback collected from the selection of the proposed results. The system takes the feedback from previous weight vectors into account, before deciding to establish a new one [17]. Traditional feedback systems that use Reinforcement Learning strategies tend to have a previously known set of actions that the agent can select, that are available to all users. Our system differs in the fact that, since the data we had available was limited, we needed to constantly test new weight variations and adapt. In this sense, Reinforcement Learning algorithms do not appear to be the best choice yet: we simply do not have yet a complete set of actions (vectors) for the agents to select. Indeed, we do not intend to find the best vector amongst our test set but instead to build knowledge by testing different models until we find the one that is most well suited, which can give valuable information about our data and possible weight distributions for the vector. With this in mind, we aimed to compare the performance of traditional statistical methods, as well as more recent Machine Learning models, to see which one can exhibit the most accuracy. Amongst the tested models, we hypothesize a better performance for either the Artificial Neural Networks or the CatBoost model, since both appear to be flexible and capable of learning more complex dependencies of the data features.

Related Work

Research involving feedback systems has successfully applied an extensive body of methods and techniques [1, 3, 8, 21, 35]. Recently, Cavalcanti et al. reviewed existing literature from 2009 to 2018 on automatic feedback systems for online learning environments. In their findings, the authors determined, in 50.79% of the examined articles, that automatic feedback systems have a positive effect on student's activities performance. Their conclusions highlighted that the most suitable application for those automatic feedback systems was to help students on a specific content/discipline [6]. In a study published in 2010, Daybelge and Cicekli suggested a novel system to rank translation results obtained by an example-based Machine Translation (EBMT) system. This system was capable of learning context-dependent co-occurrence rules, mainly by applying the user feedback obtained from evaluating the generated translation results [8].

Recent AI developments have led to improvements and a greater understanding of different recommendation systems, including their ethical implications [5]. In a paper released in 2023, Afzaal et al. [1] suggested an AI-based system to better enable self-regulated learning in students. In 2022, Pal [21] presented a research where they developed a new implicit feedback system for recommendation, suggesting the application of a Lifelong Learning Model through a Multi-agent Lambda Architecture. This system was specifically developed as a way to continuously update its model on streaming datasets and improve over time. The author aimed to develop an improved version of a system popularised by Amazon, two decades earlier. Their overall objective was to maximize the number of clicks from the users, by showing the most relevant recommendations [21]. In a separate study, Bhaskaran and Marappan developed and tested a new transduction support vector recommendation system for E-learning applications. This system allowed the learners to manage the teaching materials from any suitable course [3]. A few years earlier, Zhang et al. conducted a comprehensive review of the state-of-the-art research regarding Deep Learning-based recommendation systems [35]. In their review, the authors proposed a classification scheme for organizing and clustering the studied publications, highlighting multiple influential research prototypes. Additionally, the authors discussed the advantages and disadvantages of using DL techniques for recommendation tasks [35], while also emphasizing the number of promising techniques and models that emerged each year [35].

More relevant for our research, other studies [11, 12, 32] have previously attempted to compare the application of different learning models in the same setup. Piekutowska et al. applied both a linear and non-linear model to forecast the tuber yield of three different potato crops, using ANN and other ML models [23]. In a different study, Shaukat et al. comprehensively analyzed the published literature to determine the best performing ML classifiers, using popular datasets in sub-domains of cyber threats. The authors concluded that ML techniques have shown more potential to detect cyber threats than conventional methods [30]. Additionally, they highlighted that ML techniques were still facing challenges in the cybersecurity domain, mostly when referring to the unavailability of benchmark and updated datasets to train the respective models [30]. In the medical field, Song et al. reviewed some of the existing research works to determine whether ML models were superior at predicting acute kidney injury (AKI), when compared to Logistic Regression (LR). The authors reviewed 24 research papers, containing 84 prediction models and found that ML models can perform equally to that of LR, but specific ML models, such as Gradient Boosting, exhibited superior performance at predicting AKI to other ML models in the literature [32]. Indeed, models like the Gradient Boosting have been studied for over a decade [13, 24, 29, 33, 34].

Finally, the image comparison system that served as the base for this study was discussed in two previous studies, developed by Jardim et al. [14, 15]. The system focuses on

trademarked graphic images, and proposes a multi-stage algorithm that receives as input an RGB image and uses Deep Convolutional Networks to produce multiple outputs, corresponding to the extracted regions [14]. Image feature extraction was used to describe more commonly encountered objects and perform research with a high degree of abstraction [15]. This hybrid approach to Image Region Extraction focuses on automated region proposal and segmentation techniques such as K-Means Clustering and Watershedding which are applied to a highly variable dataset [14]. A schematic representing the behavior of the full system can be seen in Fig. 1.

Methods

The main challenge of this work is to develop a system that takes into account user feedback to improve image searches for future users. The dataset containing all the classified images was comprised of approximately 3,170,000 trademark images, from many country jurisdictions and with a wide spectrum of characteristics. Since our search objects are images, the system had to be built according to the information we had available, and considering the classification data previously obtained for each image, the existing clusters and other related data. When the original system classifies an image object, different algorithms are applied to determine the properties of the image (such as an analysis of the image regions or their edges). While the system was being developed, we chose to create a vector of 3 aggregated weights to be applied to each image cluster, each weight representing the importance of a specific evaluation algorithm. This vector allows different combinations of weights to be used for the image processing techniques. By relating a weight vector to each image cluster, we hope to obtain the distribution that best targets the unique characteristics of that cluster. This way, after the images in our dataset were classified, each cluster would theoretically have a vector that enables the best possible image comparison results. Once a user conducts an image search using our system, it has the option to validate the results that we propose, using the current vector of weights for the detected cluster. The challenge is to find the best way to apply the correction in the system so that user input can help to improve the accuracy of the classification algorithms. Since the system was intended to be easy to understand and not overbearing, we faced some significant challenges to ensure the system still worked as intended:

- There is not a fixed number of validations/interactions that the user establishes with the system. In order to maintain an enjoyable user experience, validations can be performed in any number of image search results. This means the system must be able to recognize the points at which the users stop interacting with it.
- The system can provide an extensive amount of information based on the position and quantity of changes made.

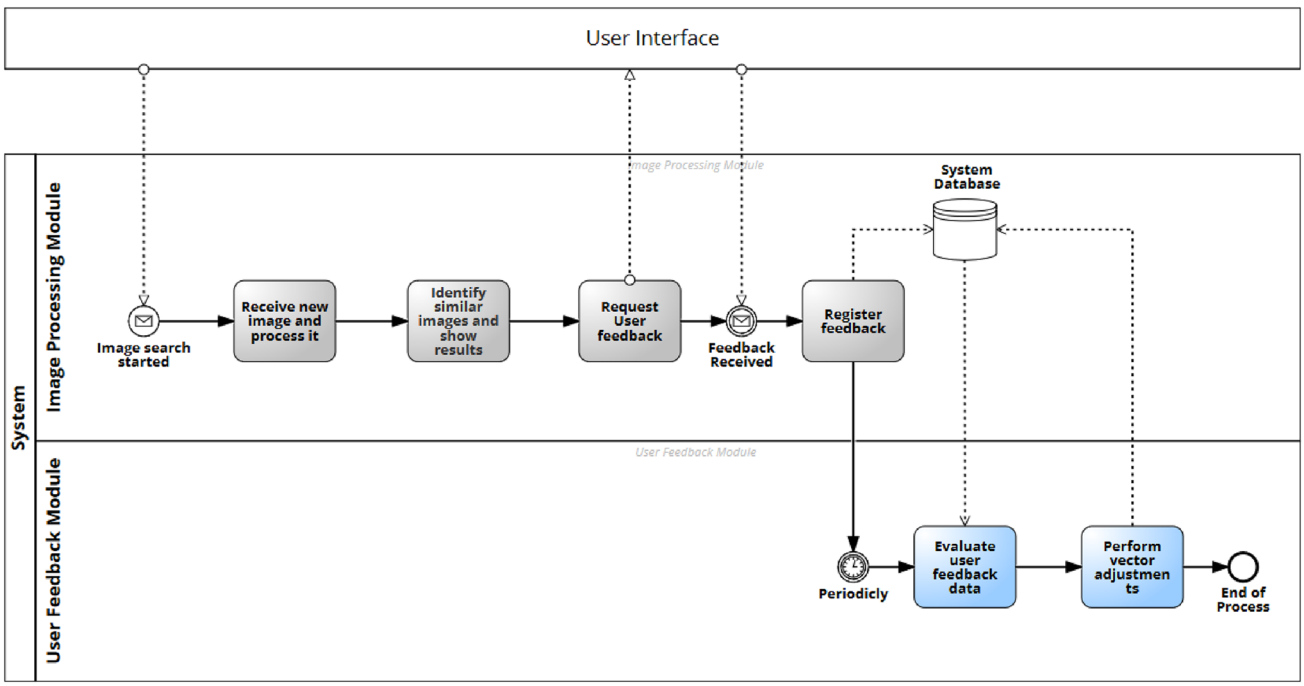

Fig. 1 BPMN diagram representing the general behavior from the two systems. The system detailed in the study refers to the blue boxes and will be detailed in the section

Few changes, or those close to the top results, indicate that the system was fairly accurate, while a large number of changes, or ones that are very distant from the top results, indicate the existence of a flaw in the weight values that need to be addressed.

– All the information should be collected in a way that's invisible to the user, in order to provide an experience as seamless as possible and to not overwhelm with excessive information.

All this information meant that we had to develop a decision algorithm, to identify and quantify the changes made by the user. This was implemented so we could correctly identify the worst performing vectors and adjust them accordingly. This sort of structure is not the traditional way that user feedback systems work. Changes had to be made due to the characteristics and necessities of the system and the information collected. We needed to build a dataset with data from real system usages, that would serve to train and test the ML models, using the user feedback data, and determine their performance. This dataset was comprised of multiple tests and applications in which a user would enter an image for comparison, look at part of the results, and provide feedback on the most similar images that were returned by the algorithm. Every cluster was set to the same default values for the weight vector. For testing purposes, if an image was found to have a similarity value above 90%, the system would automatically flag the image as a match. The user would then manually validate each result, by keeping or changing the flags for the images that they considered similar. The results include the images flagged as a match by the system, the user inputs (most similar images), the number of images that the user validated during the search, and the search date and time, amongst others.

In order to have data using different weight vectors, we developed an automatic weight variance algorithm. Its function was to analyze the recent results, determine the overall accuracy of the searches (based on user feedback), and create changes in the weight vectors if the accuracy was below a given threshold. To determine the accuracy, we looked at several metrics like the amount of changes made by the user, but also if the changes occurred far from the established threshold of 90% similarity (changes made on images with either low or very high similarity indicated an improper evaluation). This system served to dynamically create changes for each cluster, depending on the results provided by the user: clusters with lower accuracy had a bigger degree of changes to the weights and threshold (and vice-versa). A simplified model of the process can be observed in Fig. 2. Another challenge that we faced during the data-gathering phase was that we had to use images that were not classified yet, to avoid always getting perfect matches. This meant that the images used to gather the feedback data were scattered between multiple clusters, and their representation in the final data was not proportional. After users finalized a round of tests, the variance algorithm would insert changes to the cluster data, and the users would then run a new battery of tests to establish data for the new weight vector and related threshold. The final sampling data consisted of 95 images tested. Some clusters had more reach and results than others, so we opted to select the cluster (C-20) with the most available data from our dataset for this research since the comparison vectors are supposed to be iterated separately for each cluster. After verifying the cluster distributions, we

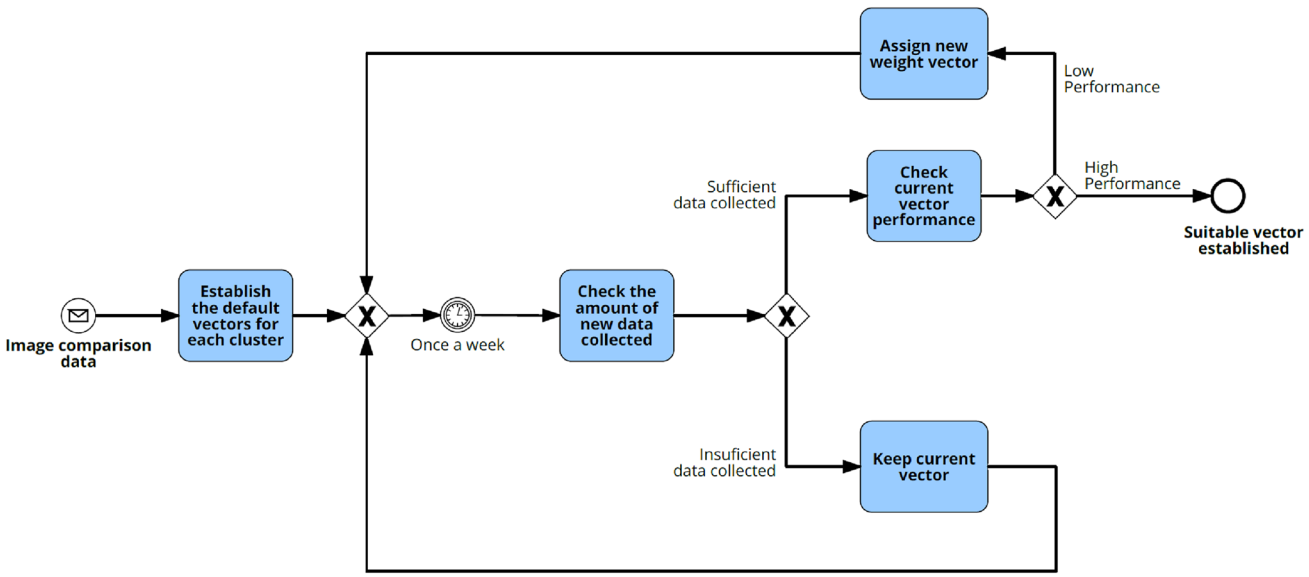

Fig. 2 Representation of the vector management algorithm and its general logic

conducted our tests on the most populated cluster in our sample (C-20).

The data was registered in an SQL database, using a website that provided the graphic interface for the user to interact. After the user inputs were collected, the collected data was saved in a CSV file to maintain its integrity during testing. The final dataset contained 3960 entries, strictly related to the cluster identified as C-20. For each model, these data were separated into a train and a test set, with 80% of the entries in the training set and 20% on the test set. This was done so the models could be tested for accuracy using a separate test set. Before each model was constructed, the features were scaled to a comparable metric, using the python library *sklearn*, using the Standard Scaler function.

While training the ML models, a k-Fold Cross-Validation was applied to all 3 ML models, to obtain a more accurate estimate of the model's accuracy. For The Logistic Regression, we opted to use the *sklearn* library, applying the *lbfgs* solver, which uses the Broyden Fletcher Goldfarb Shanno (BFGS) algorithm. This solver is the optimization algorithm used by default on the library since it can be applied to a wide range of different training data. Next, for the CatBoost model, changes in the hyper-parameters did not show significant improvements in the results. Ultimately, we opted to use a maximum of 5 trees (iterations), a depth of 4 for each tree, a learning rate value of 1 and, since the output was binary by design, we used the *Logloss* loss function. The sequential Artificial Neural Network was compiled using the *keras* library from *tensorflow* (version 2.11.0), with 2 hidden layers, the first with 4 neurons and the second with 12. Other configurations were tested, which produced similar or worse results. For training, we used the Adam optimizer as it is very performant and able to perform Stochastic Gradient Descent. Since our class is binary by nature, we used the "binary cross-entropy" loss function. For both hidden layers, composed by 4 and 12 neurons respectively, we used the rectified linear unit activation (*Relu*) function. Lastly, the output layer consisted of 1 neuron which, due to the binary nature of the result, used a sigmoid activation function. For the model training, different configurations were tested, ultimately opting for one using a batch size of 56 and 100 epochs, since it presented the best results amongst the parameters tested.

Results

After obtaining the data, we developed two separate datasets to conduct tests on AI models. The first dataset was the extensive list of image comparisons, containing the image clusters, the vector used, the similarity of the images, and a class variable indicating if the image was matched by the user or not. The type of data collected is more suitable for classification models. A second dataset, containing the aggregated performance data for the vectors tested in cluster C-20, was then isolated for analysis. The results obtained, reflecting the number of changes introduced by the users, can be observed in Fig. 3. These results indicate that 2954 entries, out of 3960, were considered by the users as correctly evaluated by the image comparison algorithm.

The results obtained when using the Logistic Regression method indicated an accuracy of 80.93%, when establishing a confusion matrix using the test set. To check if this value was representative of the data and not just of the particular test set, we applied a k-Fold Cross Validation which splits the test set into 10 subsets so we can estimate an average accuracy for all 10. With this method, we obtained an accuracy of 79.39% and a standard deviation of 1.25%. Using the Kernel SVM model, we were able to achieve an accuracy of 81.28% and a standard deviation of 1.43%, after applying the same k-Fold Cross-Validation to the test set, to determine the results more accurately. This method appeared to be slightly more accurate than the Logistic Regression to classify our dataset. When testing the Gradient Boosting model, using CatBoost, we were able to correctly predict the confusion matrix of the test set with an accuracy of 81.28% and a standard deviation of 1.43%, after applying the same k-Fold Cross-Validation as before. These results mimic the ones obtained using the Kernel SVM model. Lastly, when using the ANN, we managed to increase the accuracy of the model to 82.82%, tied with the CatBoost model for the best result for the study. This solution also did not show the same issue in classifying positive cases as the previous models did. The obtained results can be observed in Table 1.

In all of the models tested, the prediction errors are mostly associated with false positives, meaning that the models made most of their errors by predicting a match that

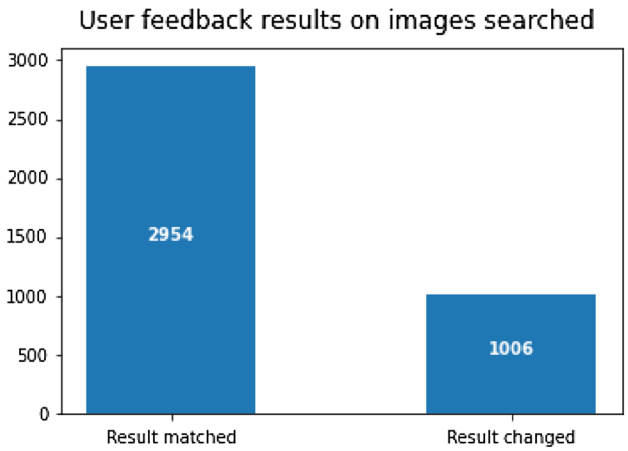

Fig. 3 User input results for the data sampled. The number of results where the outcome matched between the user and our algorithm vs. the number of changes the user introduced

did not materialize. As an example, the confusion matrix obtained for the ANN model consisted of 492 accurate matches, 92 false positives (inaccurate matches), 44 false negatives and 164 accurate negatives. With the best accuracy rate obtained of 82,82%, we conclude that the model did a capable job in classifying the existing data, with a low risk of having overfitted the model to our data.

Using the Catboost algorithm, we were also able to determine the importance of the different features used for our model. For the cluster C-20, when determining the most important weights in the vectors, the suggested values indicated a greater importance of the first weight; nearly 0% for the second value and only 8.02% for the last of the 3 weight. These results clearly indicate that, for our particular dataset, the image comparison algorithm is properly selecting the most important components of the weight vector. When considering the individual components of the weight vector, we seemed to have a higher importance for the image objects and no relevance for the image features when analyzing our samples for this cluster.

Discussion and Future Directions

Some limitations were identified, particularly when examining the sample quality for this study. Since this study was conducted for a system still under development, the quality and quantity of data available might not have been the most appropriate. Ideally, the models tested would benefit from a higher number of users conducting image searches, as well as a bigger sample size of the total searches executed. Indeed, a larger user base would not only reduce any possible bias when evaluating the quality of search results, but it would also allow for a larger amount of iterations on the weight vectors for each image cluster. This could speed up the process of obtaining good-performing weight vectors, for each image cluster. However, even accounting for the limited data, the main objective of this study was to determine a viable way to proceed and apply the user feedback, where we did achieve some success. Indeed, we managed to correctly classify different weight vectors with a good degree of accuracy and also observed some differences within the different ML models tested. The analysis of the feature importance also gave us important clues regarding the relevance of the features on our dataset, at least for this particular cluster.

Another difficulty encountered was the unusual characteristics of our user feedback system, which meant that not a lot of literature on similar projects was available. Traditional recommendation systems served as an appropriate starting point, where existing data has shown potential from Deep Learning models since these tend to better capture the intricate relationships that can exist in the data itself [35]. When comparing our results to the existing literature, recent data do seem to also support better performance when using neural networks in similar systems. In Afzaal et al. [1], the authors tested a support system for students based on AI recommendations and determined that Artificial Neural Networks and Random Forest were their best performing models, followed by Logistic Regression and K-nearest Neighbours. Although our accuracy differences were small between the different models tested, their results do seem consistent with ours. Lacic et al. [18] tested the use of Autoencoders as another potential use of DL models in an online job recommendation system. Their work indicated particularly good accuracy results, especially when applying autoencoders in a K-nearest Neighbors manner.

Recent trends have shown that this research field is thriving with innovation, with numerous deep recommendation systems being studied in the past few years [1, 9, 35]. However, some authors have also recently focused on some of the fairness and ethical problems that these systems can introduce [5, 9]. In fact, while deep learning systems have shown incredible potential, these are also known to have some limitations like their difficult interpretability and being somewhat data-hungry, as to fully support their richer parameterization [35]. These limitations were considered when we were selecting the ML models to be tested.

Conclusion

The objective of this study was to analyze the initial test data of our image retrieval system and determine some of the most suitable AI models to predict future results, with a view to developing a system to improve the performance of image comparisons, selecting the best parameters for each cluster. Due to the short amount of data available and the unusual

Table 1 Summary of the results for the different tested models

Model	Applications	Accuracy of predictions%	Standard deviation
Logistic regression	Classification	79.39	1.25%
Kernel support vector machine	Classification/ regression	81.28	1.43%
CatBoost	Classification/ regression	81.28	1.43%
Artificial neural networks (ANN)	Classification/ regression	82.82	N.A

requisites of our system, we were limited on the options available but were still able to gather important data about our system. As predicted, Artificial Neural Networks and modern ML models appear to show the best performance at predicting the result for new entries in our dataset.

The data from this study will be used to introduce improvements in our systems in the near future and, once further testing is developed and more data is gathered, we intend to re-run the tests and compare the most performing models. Additionally, once more weight vectors are tested for the different clusters, we aim to test the application of Deep Learning and Reinforcement Learning models, to determine the most suitable weight vector for each cluster. The usage of additional performance metrics would also be desirable for the different models. One of the main criticisms of Shaukat et al. was the necessity for standardized metrics when comparing the performance of models [30]. While the accuracy of the model seems to be a popular metric in research, additional descriptive metrics could be of use in future studies to detect specific performance issues.

Author Contributions Conceptualization, JV, CM and AA; formal analysis, SJ and JV; funding acquisition, CM; Investigation, SJ and JV; methodology, SJ, JV, CM and AA; project administration, CM; supervision, AA and CM; validation, SJ, JV, JA, CM and AA; writing-original draft, JV; writing-review and editing, SJ, JV and CM; All authors have read and agreed to the published version of the manuscript.

Funding Open access funding provided by FCT|FCCN (b-on). This manuscript is a result of the research project "DarwinGSE: Darwin Graphical Search Engine", with code CENTRO-01-0247-FEDER-045256, co-financed by Centro 2020, Portugal 2020 and European Union through European Regional Development Fund.

Code Availability Not applicable.

Data Availability Not applicable.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication. Not applicable.

References

1. Afzaal M, Zia A, Nouri J, Fors U. Informative feedback and explainable AI based recommendations to support students self regulation. Technol Knowl Learn. 2023. https://doi.org/10.1007/s10758-023-09650-0.
2. Barros DMS, Moura JCC, Freire CR, Taleb AC, Valentim RAM, Morais PSG. Machine learning applied to retinal image processing for glaucoma detection: review and perspective. Biomed Eng Online. 2020;19(20):1–21.
3. Bhaskaran S, Marappan R. Design and analysis of an efficient machine learning based hybrid recommendation system with enhanced density-based spatial clustering for digital e-learning applications. Complex Intell Syst. 2021. https://doi.org/10.1007/s40747-021-00509-4.
4. Bleckmann A, Meiler J. Epothilones: quantitative structure activity relations studied by support vector machines and artificial neural networks. QSAR Comb Sci. 2003;22:722–8.
5. Bonicalzi S, Caro MD, Giovanola B. Artificial intelligence and autonomy: on the ethical dimension of recommender systems. Topoi. 2023;42:819–32. https://doi.org/10.1007/s11245-023-09922-5.
6. Cavalcanti AP, Barbosa A, Carvalho R, Freitas F, Tsai Y-S, Gasevic D, Mello RF. Automatic feedback in online learning environments: a systematic literature review. Comput Educ: Artif Intell. 2021;2: 100027. https://doi.org/10.1016/j.caeai.2021.100027.
7. Cristianini N, Shawe-Taylor J. An introduction to support vector machines. Cambridge: Cambridge University Press; 2000.
8. Daybelge T, Cicekli I. A ranking method for example based machine translation results by learning from user feedback. Appl Intell. 2011;35:296–321. https://doi.org/10.1007/s10489-010-0222-7.
9. Deldjoo Y, Jannach D, Bellogin A, Difonzo A, Zanzonelli D. Fairness in recommender systems: research landscape and future directions. User Model User-Adap Inter. 2023. https://doi.org/10.1007/s11257-023-09364-z.
10. Hwang R, Lee H, Hwang HJ. Option compatible reward inverse reinforcement learning. Pattern Recogn Lett. 2022;154:83–9. https://doi.org/10.1016/j.patrec.2022.01.016.
11. Hodo E, Bellekens X, Hamilton A, Tachtatzis C, Atkinson R. Shallow and deep networks intrusion detection system: a taxonomy and survey. arXiv:1701.02145; 2017.
12. Howley T, Madden MG. The genetic kernel support vector machine: description and evaluation. Artif Intell Rev. 2005;24:379–95. https://doi.org/10.1007/s10462-005-9009-3.
13. Ibragimov B, Gusev G. Minimal variance sampling in stochastic gradient boosting. arXiv:1910.13204v1 [stat.ML] 2019.
14. Jardim S, António J, Mora C. Graphical image region extraction with K-means clustering and watershed. J Imaging. 2022;8:163. https://doi.org/10.1177/1550147718790753.
15. Jardim S, António J, Mora C, Almeida A. A novel trademark image retrieval system based on multi-feature extraction and deep networks. J Imaging. 2022;8:238. https://doi.org/10.3390/jimaging8090238.
16. Kearns M, Valiant L. Cryptographic limitations on learning Boolean formulae and finite automata. J ACM (JACM). 1994;41(1):67–95.
17. Ladosz P, Weng L, Kim M, Oh H. Exploration in deep reinforcement learning: a survey. Inf Fusion. 2022;85:1–22. https://doi.org/10.1016/j.inffus.2022.03.003.

18. Lacic E, Reiter-Haas M, Kowald D, Dareddy MR, Cho J, Lex E. Using autoencoders for session based job recommendations. User Model User-Adap Inter. 2020;30:617–58. https://doi.org/10.1007/s11257-020-09269-1.
19. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. arXiv preprint: 1509.02971, 2015.
20. Nguyen D-P, Tho M-CHBT, Dao T-T. Reinforcement learning coupled with finite element modeling for facial motion learning. Comput Methods Programs Biomed. 2022;221: 106904. https://doi.org/10.1016/j.cmpb.2022.106904.
21. Pal G. An efficient system using implicit feedback and lifelong learning approach to improve recommendation. J Supercomput. 2022;78:16394–424. https://doi.org/10.1016/j.cmpb.2022.106904.
22. Phan TD, Zincir-Heywood N. User identification via neural network based language models. Int J Netw Manag. 2019;29: e2049.
23. Piekutowska M, Niedbała G, Piskier T, Lenartowicz T, Pilarski K, Wojciechowski T, Pilarska AA, Czechowska-Kosacka A. The application of multiple linear regression and artificial neural network models for yield prediction of very early potato cultivars before harvest. Agronomy. 2021;11:885. https://doi.org/10.3390/agronomy11050885.
24. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: unbiased boosting with categorical features. arXiv:1706.09516v5 2019.
25. Qi C, Song C, Xiao F, Song S. Generalization ability of hybrid electric vehicle energy management strategy based on reinforcement learning method. Energy. 2022;250: 123826. https://doi.org/10.1016/j.energy.2022.123826.
26. Ranganathan P, Pramesh CS, Aggarwal R. Common pitfalls in statistical analysis: logistic regression. Perspect Clin Res. 2017;8:148–51. https://doi.org/10.4103/picr.PICR_87_17.
27. Raschka S, Patterson J, Nolet C. Machine learning in python: main developments and technology trends in data science, machine learning, and artificial intelligence. Information. 2020;11:193. https://doi.org/10.3390/info11040193.
28. Rito TG, Beregi S, Barton DAW. Reinforcement learning and approximate Bayesian computation for model selection and parameter calibration applied to a nonlinear dynamical system. Mech Syst Signal Process. 2022. https://doi.org/10.1016/j.ymssp.2022.109485.
29. Roe BP, Yang H-J, Zhu J, Liu Y, Stancu I, McGregor G. Boosted decision trees as an alternative to artificial neural networks for particle identification. Nucl Instrum Methods Phys Res, Sect A. 2005;543(2):577–84.
30. Shaukat K, Luo S, Varadharajan V, Hameed IA, Chen S, Liu D, Li J. Performance comparison and current challenges of using machine learning techniques in cybersecurity. Energies. 2020;13:2509. https://doi.org/10.3390/en13102509.
31. Singh V, Chen S-S, Singhania M, Nanavati B, Kar AK, Gupta A. How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries-A review and research agenda. Int J Inf Manag Data Insights. 2022;2: 100094. https://doi.org/10.1016/j.jjimei.2022.100094.
32. Song X, Liu X, Liu F, Wang C. Comparison of machine learning and logistic regression models in predicting acute kidney injury: a systematic review and meta-analysis. Int J Med Inf. 2021;151: 104484. https://doi.org/10.1016/j.ijmedinf.2021.104484.
33. Wu Q, Burges CJ, Svore KM, Gao J. Adapting boosting for information retrieval measures. Inf Retrieval. 2010;13(3):254–70.
34. Zhang Y, Haghani A. A gradient boosting method to improve travel time prediction. Transp Res Part C: Emerg Technol. 2015;58:308–24.
35. Zhang S, Yao L, Sun A, Tay Y. Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv. 2018;1(1):35.
36. Zhu M, Wang J, Yang X, Zhang Y, Zhang L, Ren Y, Wu B, Ye L. A review of the application of machine learning in water quality evaluation. Eco-Environ Health. 2022;1:107–16. https://doi.org/10.1016/j.eehl.2022.06.001.