# Comparative Analysis of Offline Recommendation Systems with Machine Learning Algorithms

**Conference Paper** · June 2021



**2 authors:**

Enes Celik
Kirklareli University



Sevinc Ilhan Omurca
Kocaeli University

# Comparative Analysis of Offline Recommendation Systems with Machine Learning Algorithms

Enes Çelik[*,1], Sevinç İlhan Omurca[2]

[*]: enes.celik@klu.edu.tr, ORCID: 0000-0002-3282-865X
[1]: *Computer Science, Kırklareli University, Kırklareli, Turkey*
[2]: *Computer Engineering, Kocaeli University, Kocaeli, Turkey*

**Abstract** In the electronic commerce and advertising industry, the importance of advice and recommendation systems has been increasing in recent years. Product ratings in consumer behavior and customer implicit knowledge are used to create recommendations in recommendation systems. These recommendations have many advantages both in terms of profit growth for the company and in finding what they are looking for easily from the customer's point of view. In this study, the customer's implicit knowledge was tested with various machine learning algorithms in offline recommendation systems as feedback. These machine learning algorithms were analyzed according to the evaluation criteria with Support Vector Machine, Gradient Boosting, AdaBoost, Random Forest, XGBoost, CatBoost and LightGBM algorithms. When creating a scalable recommendation system model, a model based on Boosting algorithms was proposed with a number of attribute extraction methods. Boosting algorithms are focused because of the sparse data in recommendation systems. In this study, a ready-made dataset from the University of California Irvine Machine Learning Repository was used to characterize and predict user behavior related to recommendation systems with personalized mobile recommendation systems that are sensitive to in-vehicle content. The ready-made data set has a lot of categorical attributes and although there are quite infrequent records, successful results have been achieved. In the proposed offline recommendation system, the best model was found to have a LightGMB algorithm with a 75% accuracy rate and a time complexity of 0.26 seconds. Open source Python libraries and the Google Colab platform were used in experimental studies. A number of recommendations for future research are given.

***Keywords:*** *Recommendation System, Machine Learning, Boosting*

## I. Introduction

A recommendation system is an information discovery system that helps find user preference for a specific item. Recommendation systems are special methods that give ideas about what is used by different users. Many decisions can be made considering recommendations such as which product to buy, what type of music to listen to, or what to read in their online news. It helps not only the user but also the firm to achieve the sales target. What the system recommends is called "Item", and users are called "User". A suggestion system typically focuses on a specific item format, and its recommendation is applied to give useful recommendations for a specific form [1].

As recommendations are personalized, different users can get different suggestions. Online magazines and newspapers offer non-personalized suggestions that are very easy to produce. Magazines and newspapers will not be useful in recommending popular articles to us. However, sequential lists of articles can be produced to make personalized recommendations. User preferences are considered when creating sorts to extract the most appropriate products or services. To calculate the most similar products or services, data is implicitly or explicitly collected by understanding the user's preferences [2].

The main purpose of a commercial recommendation system is to increase sales or to sell products that cannot be sold by recommending them. Recommendations are provided taking into account that products and services meet the customer's requirements. The company's purpose in using the recommendation system is to increase the number of users who prefer products or services on the web page. The recommendation system helps a user find items that can be difficult to find without a specific reference. It can be difficult to make such offers without a recommendation system because the service provider cannot risk suggesting videos that are not suitable for the user's taste. Thus, the recommendation system can also suggest unpopular items [3].

The recommendation system helps improve a person's experience with the app or website. It helps to provide better human-computer interaction, as well as providing important and relevant recommendations. Effective recommendations, with the help of an accurate and interactive user interface, increase the use of the system and the chances of accepting recommendations. Customer always prefers to use a website that recognizes their former users and treats them as respected customers. It is quite valuable considering the data obtained from the user in previous interactions, such as product ratings. Therefore, the more the customer uses a particular site, the better the recommendation model of the site and can be more personalized according to the user's preferences. The recommendation system acts as an actively evolving system according to the user's preferences by collecting the predictions or open ratings made by the system in some way [4].

In recent years, the importance of recommendation systems has been increasing in the virtual trade and advertising sector. In addition to product ratings in consumer behavior, implicit knowledge of the customer is used in recommendation systems to create recommendations. In this study, the implicit knowledge of the customer was analyses with various machine learning models in offline recommendation systems as feedback.

## II. Literature Review

In a study by Ramzan by et al, textual hotel reviews, numerical ratings, votes, and the number of video views make it difficult to get hotel recommendations. To produce the right recommendations, a smart approach has been proposed that also deals with large-scale heterogeneous data to meet the needs of potential customers. The collaborative filtering approach is one of the most popular techniques of the advice system for creating recommendations. A new Collaborative Filtering approach has been proposed that uses idea-based sensitivity analysis to achieve the hotel feature matrix. A recommendation system has been developed that combines word analysis, syntax analysis, and semantic scientific analysis to understand the sensitivity to hotel features and the profile of the guest type [5].

In a study by Thangavel by et al, estimation that students' placement in higher education is more difficult when the complexity of educational institutions increases. Educational institutions have sought more efficient technologies that help with better management and support decision-making procedures or help them identify recommendation systems. With machine learning techniques, new information can be extracted from historical data in the databases of the educational institution. This data is used to train and test the model for rule definition. This article provides a recommendation system that requires students to have one of the five placement statuses, Dream Company, Core Company, Mass Recruiters, Not Eligible, and Not Interested in Placements [6].

In a study by Bertens by et al, critical to games where online players can choose or purchase plenty of items during the game to advance and fully enjoy their experience. To try to maximize such purchases, a recommendation system has been developed to provide players with items that may be interesting. Such systems can better achieve their goals by using machine learning algorithms that can predict the rating of an item or product by a particular user. In this study, two of these algorithms were evaluated and compared with Extremely Randomized Trees and Deep Neural Network; both have given good results for video game recommendation engines [7].

In a study by Aher by et al, are used clustering and attribution algorithms in the course recommendation system, which recommended another course to another student based on the choice of other students for specific courses collected from the electronic course platform Moodle. A recommendation system was developed that uses the clustering technique K-Means and the association rules Apriori Algorithm [8].

In a study by Shahbazi by et al, take advantage of collaborative filtering approaches, implicit and explicit features in existing recommendation systems, and report a good classification or forecast result. A collaborative filtering-based algorithm is provided to handle large user data with purchase orders and recurrent purchased products. The proposed algorithm was developed by combining the Extreme Gradient Boosting machine learning architecture and the word2vec method to discover products purchased based on users' click patterns [9].

In a study by Fanca by et al, recommendations relate to different decision-making mechanisms, different techniques such as which product to buy, which movie to watch, or which holiday to book. The proposed method was developed by combining multiple recommendation systems that can make item suggestions based on user, element, and user-element interaction data using different machine learning algorithms [10].

In a study by Khanal by et al, students help through ever-increasing online learning materials presentation, personalized e-learning, and recommendation systems. Challenges continue in the form of data scarcity, cold start-up, scalability, and accuracy. This study developed an effective recommendation system for recommendation systems in the context of e-learning with Content-Based, Collaborative Filtering, Knowledge-Based, and Hybrid Machine Learning Systems [11].

In a study by Nawrocka by et al, are focused on contiguity of users or items or content-based filtering algorithms. These algorithms are calculated according to similarities, disadvantages, and advantages, measures to evaluate the algorithm. Successful results were given based on computer simulations to evaluate how collaborative and hybrid machine learning algorithms work [12].

In a study by Hernandez by et al, are developed an investment recommendation system based on the issuance of trading signals on the stock exchange. Calculation of technical analysis metrics performed with data extraction and machine learning algorithms for the prediction used also helped. The purpose of the platform is not only to be intuitive but also to provide personalized recommendations to users who are not experts in the exchange according to their history [13].

In a study by Lee by et al, are developed a new collaborative filtering algorithm based on Deep Neural Networks for the Netflix movie recommendation. The deep neural network used a normalized user rating vector and a normalized item rating vector. The proposed method performed better than traditional collaborative filtering algorithms [14].

## III. Material And Method

### *A. Random Forest*

Random Forest has been proposed as a variant of decision trees. In the Random Forest algorithm, each tree is defined as a combination of tree determinants, which are independently sampled and depend on the values of a random vector with the same distribution for all trees in the forest. The community algorithm helps us predict better target variables by reducing variance and error in classification problems. It does this by resampling method. In this method, the data set is divided into subsampling sets using the random repetitive sampling method, respectively. These installed subtrees are intended to reduce the error in the model [15].

### *B. Extreme Gradient Boosting*

Extreme Gradient Boosting (XGBoost) is a high-performance format of the Gradient Boosting algorithm optimized with various improvements. The most important features of the algorithm are its high predictive power, its ability to prevent overlearning, its ability to manage empty data, and its ability to do so quickly or as soon as possible. It is shown as the best decision tree-based algorithms [16].

### *C. LightGBM*

LightGBM is a histogram-based algorithm. Reduces calculation cost by making variables with continuous values truss. Thanks to this method, both the training time is shortened and the use of resources decreases. Two level-oriented or leaf-oriented strategies can be used in decision tree learning. In the level-oriented strategy, the balance of the tree is maintained while the tree grows. In the leaf-oriented strategy, the process of dividing from the leaves, which reduces loss, continues. LightGBM is separated from other boosting algorithms thanks to this feature. With leaf-oriented strategies, the model has a lower error rate and learns faster. LightGBM reduces data size to calculate information gain by neglecting attributes that may be considered less important. Combines or merges variables to reduce attribute size. With these two functions, LightGBM increases the efficiency of the training process [17].

### *D. CatBoost*

CatBoosting excels at processing categorical features that prevent overfitting and unbiased gradient calculation. High learning speed, ability to work with digital-categorical-text data, GPU support, and visualization options are the most reserved features of other community algorithms. It can cope with empty data, encoding categorical data [18].

### *E. AdaBoost*

AdaBoost is similar to Random Forest in that it determines the estimates made by each decision tree in the forest to decide on the final classification. In general, the depth of decision trees in AdaBoost is 1. In addition, the predictions made by each decision tree may have different effects on the final prediction made by the model. Adaboost is a sequential creation process based on minimizing errors from previous models while increasing the impact of high-performance models [19].

### *F. Gradient Boosting*

The Gradient Boost algorithm is a machine learning technique that creates prediction models similar to decision trees for classification problems. Gradient increment is a method by which new models are created that calculates the error in the previous model, and then add leftovers to make the final prediction. The Gradient Boost algorithm does not create nodes to improve each tree, such as AdaBoost. Instead, it begins with the leaf. This leaf represents an initial estimate for all weights. The first estimate here is the average value. Then Gradient Boosting creates the tree [20].

### *G. Support Vector Machine*

Support Vector Machine is a controlled machine learning algorithm that can be used for classification problems. In this algorithm, each data item is plotted as a point in the n-dimensional space (n: number of attributes) along with the value of each property, which is the value of a particular coordinate. Next, classification is carried out by finding the hyperplane, which distinguishes quite well from the two classes. Support Vector Machine is a boundary that best separates two classes from hyper-planes [21].

### *H. Dataset*

A ready-made dataset from UCI Machine Learning Repository was used to characterize and predict user behaviour related to personalized recommendation systems sensitive to in-car content. Amazon Mechanical Turk was used to collect data about users interacting with in-car mobile advice systems. This dataset was used to understand the behaviour of consumers and to predict their reaction to different coupons recommended in different contexts.

In the event of a purchase, information is collected when the user is shown a coupon. In this context, if the user accepts the offers (coupons) offered to them, the class label is marked as 1. Here, the user's acceptance of the coupon offer generates implicit feedback from the point of view of the recommendation systems. In which cases, recommendation systems can be improved by estimating which user accepts coupons.

The estimation problem is to predict whether a customer will accept a coupon for a particular location, taking into account their demographic and contextual characteristics. The venue here is advertised and presented as a suggestion. Replies in which the user says "I will go to the place immediately" or "I will go to the place later before the coupon expires" are labeled Y = 1, and their responses saying "No, I do not want the coupon" is labeled "Y = 0".

Five types of coupons; bars, takeaway restaurants, coffee shops, cheap restaurants, expensive restaurants have been investigated; because in the future, these venues will be advert advertised or recommended to users. Users were asked to provide their demographic information and preferences in the first part. In the second part, each user was told 20 different driving scenarios with additional context information and coupon information and asked if they would use them. Of the 752 surveys (scenarios), 652 were accepted and 12,684 rows of data cases were obtained [22].

The attributes of this data set include:

1. User Attributes
• Gender: male, female
• Age: below 21, 21 to 25, 26 to 30, etc.
• Marital Status: single, married partner, unmarried partner, or widowed
• Number of children: 0, 1, or more than 1
• Education: high school, bachelor's degree, associate's degree, or graduate degree
• Occupation: architecture & engineering, business & financial, etc.
• Annual income: less than $12500, $12500 - $24999, $25000 - $37499, etc.
• Number of times that he/she goes to a bar: 0, less than 1, 1 to 3, 4 to 8, or greater than 8
• Number of times that he/she buys takeaway food: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
• Number of times that he/she goes to a coffee house: 0, less than 1, 1 to 3, to 8 or greater than 8
• Number of times that he/she eats at a restaurant with an average expense of less than $20 per person: 0, less than 1, 1 to 3, 4 to 8, or greater than 8
• Number of times that he/she goes to a bar: 0, less than 1, 1 to 3, 4 to 8, or greater than 8

2. Contextual Attributes
• Driving destination: home, work, or no urgent destination
• Location of user, coupon, and destination: we provide a map to show the geographical location of the user, destination, and venue, and we mark the distance between every two places with the time of driving. The user can see whether the venue is in the same direction as the destination.
• Weather: sunny, rainy, or snowy
• Temperature: 30Fo, 55Fo, or 80Fo
• Time: 10AM, 2PM, or 6PM
• Passenger: alone, partner, kid(s), or friend(s)

3. Coupon attributes
• Time before it expires: 2 hours or one day

## IV. Experimental Studies And Results

An open-source Python, analyses, and measurements were made using Numpy, Pandas, Scikit-Learn, Matplotlib libraries. During the dataset preprocessing phase, data types are encoded as in Table I, categorical, numeric, and floating-point.

TABLE I
DATA TYPES OF ATTRIBUTES

| Attribute | Data Type |
|---|---|
| destination | category |
| passenger | category |
| weather | category |
| temperature | int64 |
| time | int64 |
| coupon | category |
| expiration | category |
| gender | category |
| age | int64 |
| maritalStatus | category |
| has_children | int64 |
| education | category |
| occupation | category |
| income | float64 |
| car | category |
| bar | category |
| coffeeHouse | category |
| carryAway | category |
| restaurantLessThan20 | float64 |
| restaurant20To50 | float64 |
| toCoupon_GEQ5min | int64 |
| toCoupon_GEQ15min | int64 |
| toCoupon_GEQ25min | int64 |
| direction_same | int64 |
| direction_opp | int64 |
| Y | int64 |

5-fold cross validation was performed on the data set. Since it is an offline recommendation system, the evaluation criteria are shown according to the Confusion Matrix table according to accuracy, precision, recall, f1-score values [23]. Confusion Matrix equations are calculated according to the criteria in Table II.

TABLE II
CONFUSION MATRIX

| | | Predicted Class | |
|---|---|---|---|
| | | Positive | Negative |
| Actual Class | Positive | True Positive (TP) | False Negative (FN) |
| | Negative | False Positive (FP) | True Negative (TN) |

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F_1\ Score = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

The most common machine learning methods are used to classify the in-car mobile recommendation data set. Support Vector Machine, Random Forest, XGBoost, LightGBM, CatBoost, AdaBoost, and Gradient Boosting methods were used in the classification experiments. During the test phase, accuracy, precision, recall, and F1-Score are considered benchmark measures. During the classification phase, 80% of the data sets were used in the training process and 20% of the data sets were used in the testing process. To find the best method, you need to have a high accuracy rate. Experimental results are given in Table III.

TABLE III
CLASSIFICATION ACCURACIES OF DIFFERENT ALGORITHMS

| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| SVM | 0,56 | 0,56 | 0,57 | 0,57 |
| AdaBoost | 0,67 | 0,68 | 0,77 | 0,72 |
| CatBoost | 0,74 | 0,74 | 0,83 | 0,78 |
| **LightGBM** | **0,75** | **0,75** | **0,84** | **0,79** |
| RandomForest | 0,74 | 0,73 | 0,82 | 0,78 |
| XGBoost | 0,71 | 0,71 | 0,82 | 0,76 |
| GradientBoosting | 0,73 | 0,71 | 0,82 | 0,76 |

When the results shown in table 3 are examined, it is determined that LightGBM achieves more accurate results than other algorithms. LightGBM has achieved high performance compared to other algorithms with 75% accuracy, 75% precision, 84% recall, and 79% f1 score. In addition, boosting algorithms showed close results.

The processing times of the methods are also measured. Generally, boosting algorithms have shorter processing times, while the Support Vector Machine algorithm has taken longer than other algorithms. As with accuracy, the shortest time in terms of processing time belongs to the LightGBM algorithm. The processing times of the algorithms are given in the graph in Table IV.

TABLE IV
THE PROCESS TIME OF ALGORITHMS

| **Algorithms** | **Time (second)** |
|---|---|
| SVM | 4,26 |
| AdaBoost | 0,43 |
| CatBoost | 0,55 |
| **LightGBM** | **0,26** |
| RandomForest | 1,19 |
| XGBoost | 0,69 |
| GradientBoosting | 1,28 |

Computational time is recorded to estimate efficiency of algorithms for recommendation systems. The algorithms are evaluated according to both accuracy and time complexity. In Fig. 1. the comparative graph of both the accuracy and the processing time of the algorithms is given.

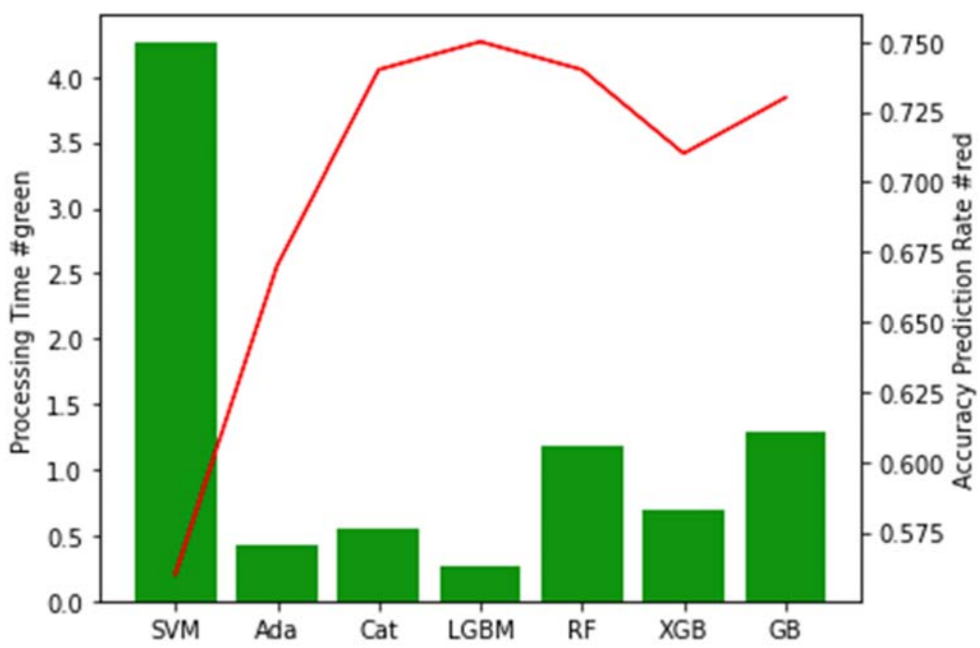

Fig. 1. Model Performances

## V. CONCLUSION

In this study, boosting algorithms in machine learning for offline recommendation systems were tested comparatively. Machine learning algorithms that are successful in categorical data are preferred for better classification accuracy. This experimental study has shown that recommendation systems are important for the advertising industry. Recommendation systems can be used to develop hybrid systems along with collaborative filtering and content-based filtering methods, as well as classification and clustering models in machine learning. It has been proven by this experimental study that it is important in its implicit knowledge besides explicit knowledge in recommendation systems. A higher performance rate was achieved according to the study [22] that prepared and analyses the data. In future studies, experiments can be made with different attribute selection or mitigation techniques. In addition, classification accuracy can be improved by performing tests with different machine learning algorithms by parameter optimization.

## REFERENCES


[1] Abbar, Sofiane, Mokrane Bouzeghoub, and Stéphane Lopez. "Context-aware recommender systems: A service-oriented approach." VLDB PersDB workshop. 2009.

[2] Adomavicius, Gediminas, et al. "Incorporating contextual information in recommender systems using a multidimensional approach." ACM Transactions on Information Systems (TOIS) 23.1 (2005): 103-145.

[3] Balabanović, Marko, and Yoav Shoham. "Fab: content-based, collaborative recommendation." Communications of the ACM 40.3 (1997): 66-72.

[4] Deshpande, Mukund, and George Karypis. "Item-based top-n recommendation algorithms." ACM Transactions on Information Systems (TOIS) 22.1 (2004): 143-177.

[5] Ramzan, B., Bajwa, I. S., Jamil, N., Amin, R. U., Ramzan, S., Mirza, F., & Sarwar, N. (2019). An intelligent data analysis for recommendation systems using machine learning. Scientific Programming, 2019.

[6] Thangavel, S. K., Bkaratki, P. D., & Sankar, A. (2017, January). Student placement analyzer: A recommendation system using machine learning. In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 1-5). IEEE.

[7] Bertens, P., Guitart, A., Chen, P. P., & Perianez, A. (2018, August). A machine-learning item recommendation system for video games. In 2018 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-4). IEEE.

[8] Aher, S. B., & Lobo, L. M. R. J. (2013). Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data. Knowledge-Based Systems, 51, 1-14.

[9] Shahbazi, Z., Hazra, D., Park, S., & Byun, Y. C. (2020). Toward Improving the Prediction Accuracy of Product Recommendation System Using Extreme Gradient Boosting and Encoding Approaches. Symmetry, 12(9), 1566.

[10] Fanca, A., Puscasiu, A., Gota, D. I., & Valean, H. (2020, October). Recommendation Systems with Machine Learning. In 2020 21th International Carpathian Control Conference (ICCC) (pp. 1-6). IEEE.

[11] Khanal, S. S., Prasad, P. W. C., Alsadoon, A., & Maag, A. (2019). A systematic review: machine learning based recommendation systems for e-learning. Education and Information Technologies, 1-30.

[12] Nawrocka, A., Kot, A., & Nawrocki, M. (2018, May). Application of machine learning in recommendation systems. In 2018 19th International Carpathian Control Conference (ICCC) (pp. 328-331). IEEE.

[13] Hernández-Nieves, E., del Canto, Á. B., Chamoso-Santos, P., de la Prieta-Pintado, F., & Corchado-Rodríguez, J. M. (2020, June). A Machine Learning Platform for Stock Investment Recommendation Systems. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 303-313). Springer, Cham.

[14] Lee, H., & Lee, J. (2019). Scalable deep learning-based recommendation systems. ICT Express, 5(2), 84-88.
[15] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
[16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794. ACM, 2016.
[17] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30, 3146-3154.
[18] Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363.
[19] Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).
[20] Friedman, J. H. (2002). Stochastic gradient boosting. Computational statistics & data analysis, 38(4), 367-378.
[21] Noble, W. S. (2006). What is a support vector machine? Nature biotechnology, 24(12), 1565-1567.
[22] Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., & MacNeille, P. (2017). A bayesian framework for learning rule sets for interpretable classification. The Journal of Machine Learning Research, 18(1), 2357-2393.
[23] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.