International Journal of Engineering and Management Research e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-13, Issue-3 (June 2023)
https://ijemr.vandanapublications.com https://doi.org/10.31033/ijemr.13.3.23

A Personalized Tour Recommender in Python using Decision Tree

Nayma Khan[1] and Mohammad Haroon[2]
[1]Student, Department of Computer Science & Engineering, Integral University, Lucknow, Uttar Pradesh, INDIA
[2]Professor, Department of Computer Science & Engineering, Integral University, Lucknow, Uttar Pradesh, INDIA

[1]Corresponding Author: naymakhan21@gmail.com

Received: 21-05-2023 **Revised:** 06-06-2023 **Accepted:** 22-06-2023

ABSTRACT

A tourist recommendation system has been implemented based on python, Django framework and MySQL database. Firstly, crawler technology is used to crawl the ratings(reviews of customer experiences) information of TripAdvisor, and then the crawled ratings data is stored in MySQL. In the system, users can view the location, can view the opinion analytics(review packages), and collect the location information. According to the user's collection of location information, the decision tree is used to recommend locations that may be of interest to users. If a new user enter his requirements then decision tree will predict best location based on his given input. Decision tree don't need new users past experience data. Through the design of these functional modules, the whole tourist recommendation system is realized.

Keywords-- Recommendation System, K-Nearest Neighbor (KNN) Algorithm, Convolution Neural Network (CNN) Algorithm, Decision Tree Algorithm

I. INTRODUCTION

A Tourist Recommendation System is significant in our social life since it enhances enjoyment while also providing a tailored user experience. Using such recommendation algorithms, users may receive recommendations for a group of locations based on their interests or the popularity of the locations. Despite the fact that there are hundreds of tourist recommendation systems, the majority of them either cannot recommend a location to existing users or cannot propose a location to a new user at all. In this paper, we compared the different algorithms to step up a tourist recommendation system. This recommendation engine scours tourist databases for all relevant variables, such as popularity and attractiveness, needed to make a recommendation[1,2,3]. Large correlations between a number of different categories of items can often be applied to generate more accurate recommendations. The proposed solution is efficient and effective, according to experiments using actual data. This Tourist Recommendation system will use a comparison among different algorithms that can be used to form a tourist recommendation system, to look for places similar to the customer's taste or genre, and would also combine recommendations that are highly rated by different customers' experiences so that the customer is not restricted to a particular category or genre but possesses the capacity to discover new and varied types[5,6].

II. LITERATURE REVIEW

The major goal of the tourist recommendation system is to reduce the issue of cold start and, to a certain extent, increase the forecast accuracy of the recommendation algorithm, according to our discussion and review of all trends and methodologies. Many techniques have been investigated, and the findings are as follows:

- Collaborative filtering is one of the widely used techniques in recommendation Systems. The major element that impacts how accurately collaborative filtering can predict outcomes is data sparsity[7,8]. Collaborative filtering system also suffers from the 'new user' problem. The addition of a new rating need to immediately modify all predictions.
- Also, author in [9] discusses the data collection process for a tourism recommendation system, including user information, interaction records, tourist attraction information, and contextual information.
- Author in [10]proposed TRS is hybrid since it integrates the three recommender methods (CF, CB, and DF) using two hybridization procedures, the weighted and switching approaches. The proposed approach benefits from the advantages of each recommender method and overcomes its drawbacks[11].
- Design and implementation of a real-time trip suggestion system that satisfies various requirements and doesn't need prior information[12].
- Author in [13] mentions a project which aims to use big data and AI to provide intelligent tools to

target and recommend the most suitable tourist offer, track and analyze opinions and forecast tourist demand.

Scope and Objective

While planning for trips it was difficult to choose the place, not every time the choose place is right to visit. Ever place has its own beauty and time to visit. And because of less knowledge about the place is becomes difficult to select spot. With the help of this system people can select right spot to visit by learning about the places to visit. This system will help for getting more information on the basis of the people's review who visited the places[14,15,16].

III. PROPOSED FRAMEWORK

A mathematical model for a tourist recommendation system can be designed based on several factors and techniques. Here's a high-level overview of a possible mathematical model:

The proposed model of tourist recommendation system in deep neural network consists of a number of hidden layers with a required number of neurons. A vectored representation of neurons in a hidden layer is given by:

$$A^{L} = f\left(w_{A+1}^{(L)T[L-1]} + B^{L}\right)$$

where $a^{[l]}$ is a vector and each element represents a nron in layer l, $W^{[l]}$ is the weight matrix in layer l. such that $W^{[l]} \in R^{i \times j}$, where i is the number of nodes in the hidden layer l. and j is the number of nodes in the previous layer (including the bias term $b^{[l]}$). Each neuron in the network is a nonlinear combination of inputs $a^{[l-1]}$. ted by the parameters w_i is the activation function. The proposed model implements two activation functions for the hidden layers and the output layer.

Tectified linear unit (ReLU) has been used as activation function for the hidden layers:

$$f(x) = \max(0, x)$$

The sigmoid function has been used as activation function for the output layer:

$$f(x) = \frac{1}{1 + \exp -x}$$

The output of the last feed-forward layer is passed through a sigmoid activation function to scale each of its element values in the range [0,1]. The model is trained using binary cross-entropy as the loss function, which is defined as

$$L_{(1-y_{ij})\log(1-\sigma(2ij))} \frac{-1}{T}\sum_{L=1}^{T}\sum_{j=1}^{U}(y_{ij}\cdot\log(\sigma(z_i)'))t$$

R2Tour is an AI-based recommendation system for personalized tour suggestions. It makes use of robust machine learning models like K-NN, SVM, voting, random forest, XGBoost, and LightGBM. The system examines an extensive dataset on Jeju tourism that contains information on local attractions, tourist profiles, and real-time context. Performance metrics for recommendations are measured using evaluation measures like accuracy and F1-score. With an Intel Core i9 processor and NVIDIA TITAN RTX graphics card, R2Tour runs on high-performance hardware[17.18,19,20].

To verify the recommendation performance of the model, we utilized Equation (**1**), which measures the accuracy and the F1-score, which are used when the data are imbalanced. The F1-score evaluation index uses the micro-F1 corresponding to Equation (**2**) and macro-F1 corresponding to Equation (**3**) to prove its effectiveness in a data imbalance problem[21,22,23].

$$ACURACY = \frac{T_P + TN}{TN + TP + FN} \quad (1)$$

$$MICROF1\ SCORE = \frac{2T_P}{2TP + FP + FN} \quad (2)$$

$$MICROF1\ SCORE = \sum_{i=0}^{n} F1\ SCORE \quad (3)$$

Data Representation

Let L be the set of locations: L = {l1, l2, ...,ln}, where n is the total number of locations. Represent each location l_i as a vector of features: l_i = (f1, f2, ..., fm), where m is the number of features for each location[24,25].

User Preferences

Let U be the set of users: U = {u1, u2, ...,uk}, where k is the total number of users. Model user preferences for user $u_i$as a vector: P_i = (p1, p2, ..., pm), where p_j represents the preference of user $u_i$for feature f_j. Assign weights to each preference: W = (w1, w2, ...,wm), where $w_j$represents the weight of feature f_j.

Similarity Metrics

Define a similarity function sim(u_i, l_j) to measure the similarity between user u_i and location $l_j$based on their preferences and features[26].

Recommendation Algorithm

Design an algorithm to generate recommendations based on user preferences and location attributes. For each user u_i, calculate the similarity scores with all locations in L using the similarity function sim(u_i , l_j). Select the top-k locations with the highest similarity scores as recommendations for user $u_{i[27]}$.

Feedback and Personalization

Incorporate user feedback to refine the recommendations over time. Update user preferences P_i based on user interactions with recommended locations. Adjust the weights W of the features in the user profile to reflect evolving preferences[28].

Optimization

Consider optimization techniques, such as matrix factorization, dimensionality reduction, or clustering, to enhance the efficiency of the recommendation system.

Evaluation

Define evaluation metrics, such as precision, recall, F1-score, or mean average precision, to measure the effectiveness of the recommendation system. Use historical data or user studies to assess the performance of the system and fine-tune the model accordingly.

3.1 Recommendation through KNN

The classification algorithm KNN (K-Nearest Neighbours) is straightforward and understandable. It is a non-parametric method that bases its predictions on how similar the input data point and its neighbouring data points are.

The KNN algorithm can be defined as follows:

1. Load the training dataset.
2. Choose the value of K (the number of nearest neighbors to consider).
3. For each new data point to classify: a. Calculate the distance between the new data point and all the data points in the training set. b. Select the K nearest neighbors based on the calculated distances. c. Assign the class label to the new data point based on the majority class among its K nearest neighbors.
4. Return the predicted class label for each new data point.

The KNN algorithm classifies new data points based on the class labels of their nearest neighbors. It assumes that data points with similar features tend to belong to the same class. The value of K determines the influence of the neighbors on the classification decision[29,30].

The most commonly used distance metric in KNN is Euclidean distance, which calculates the straight-line distance between two points in a multidimensional space. The Euclidean distance between two points (p1, p2, ...,pn) and (q1, q2, ..., qn) is given by the formula :

dist = sqrt ((p1 - q1)^2 + (p2 - q2)^2 + ... +(pn - qn)^2)

Customer ID	Rating
1	4.5
2	3.2
3	4.8
4	2.7
5	4.2

Table I: Rating based on KNN

Table I, representing the relationship between customers and their ratings based on KNN algorithm using dataset.

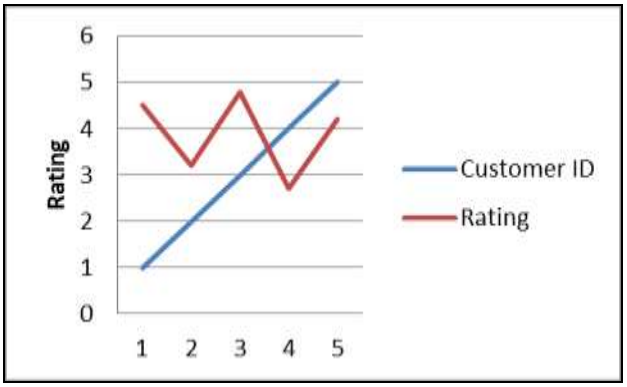

Figure I: Graph showing KNN ratings

The ratings are based on a dataset obtained from recommendation system where customers have provided their ratings for various locations or attractions is shown in Fig I.

3.2 Recommendation through CNN

CNN (Convolutional Neural Network) is a deep learning algorithm commonly used for processing and analyzing visual data such as images or videos. It is designed to automatically learn and extract relevant features from input data through a series of convolutional layers.

The CNN Algorithm can be defined as follows:

Step 1: Choose a Dataset. Choosing a dataset for the image classification task is the first step.

Step 2: Prepare the Dataset for Training.

Step 3: Create Training Data and Assign Labels.

Step 4: Define and Train the CNN Model.

Step 5: Test the Model's Accuracy.

Y[i, j] = sum(W[c, m, n] * X[i, c, m+p, n+q])

Where Y is the output feature map, W is the filter/kernel, X is the input data, and i, j, c, m, n, p, q represent indices.

The operation is applied layer by layer to process the input data and extract meaningful features for classification, detection, or other tasks.

Customer ID	Rating
1	3.8
2	3.5
3	4.2
4	2.9
5	4.5

Table II: Rating based on CNN

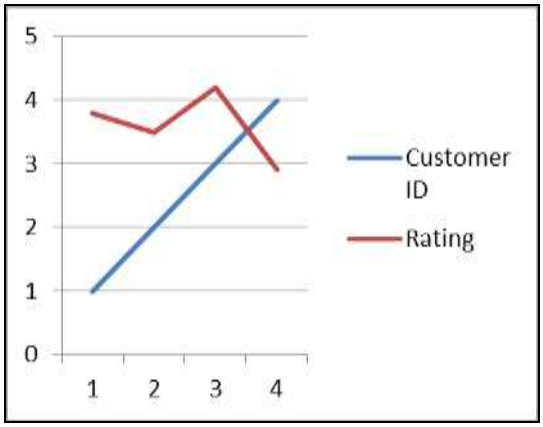

Figure II: Graph showing CNN Ratings

3.3 Recommendation through Decision Tree

A decision tree is a supervised machine learning algorithm that uses a tree-like structure to make decisions or predictions. It learns from labeled training data and creates a flowchart-like model where each internal node represents a decision based on a feature, each branch corresponds to a possible outcome of the decision, and each leaf node represents a predicted class or value.

Using the algorithm below, it is possible to understand the entire procedure:

Step 1: Begin the tree with the root node, says S, which has the entire dataset.

Step 2: Use the Attribute Selection Measure (ASM) to identify the dataset's best attribute.

Step 3: Split the S into groups that include potential values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Making new decision trees recursively using the subsets of the dataset generated in step 3. Keep going through this procedure until you reach a point where you can no longer categorise the nodes, at when you will refer to the last node as a leaf node.

Decision trees typically use impurity measures to evaluate the homogeneity or impurity of the data at each node. Common impurity measures include:

- Gini Impurity: Gini(p) = 1 - Σ(pi)^2, where pi represents the proportion of instances of class i in a node.
- Entropy: Entropy(p) = -Σ(pi * log2(pi)), where pi represents the probability of class i in a node.
- Classification Error: Classification_Error(p) = 1 - max(pi), where pi represents the proportion of instances of class i in a node.

Customer ID	Rating
1	4.5
2	3.7
3	4.9
4	3.0
5	4.6

Table III: Rating based on Decision Tree

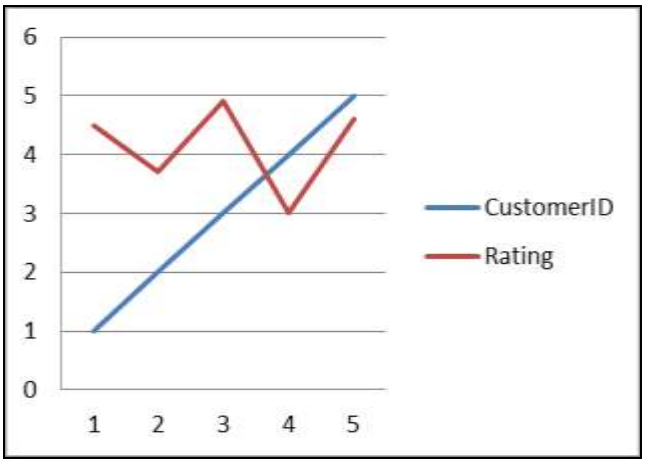

Figure III: Graph showing DT Ratings

3.4 Comparative Analysis

CustomerID	KNN Rating	CNN Rating	DT Rating
1	4.5	3.8	4.5
2	3.2	3.5	3.7
3	4.8	4.2	4.9
4	2.7	2.9	3
5	4.2	4.5	4.6

Table IV: Ratings of different algorithms

In the above table, each row represents a customer, and the columns indicate the customer's ID along with their corresponding ratings for the KNN algorithm, CNN algorithm, and decision tree algorithm.

Table IV is used to compare the ratings generated by the KNN algorithm, CNN algorithm, and decision tree algorithm in a tourist recommendation system. It allows for an evaluation of the performance and effectiveness of

these algorithms based on their respective ratings for different customers.

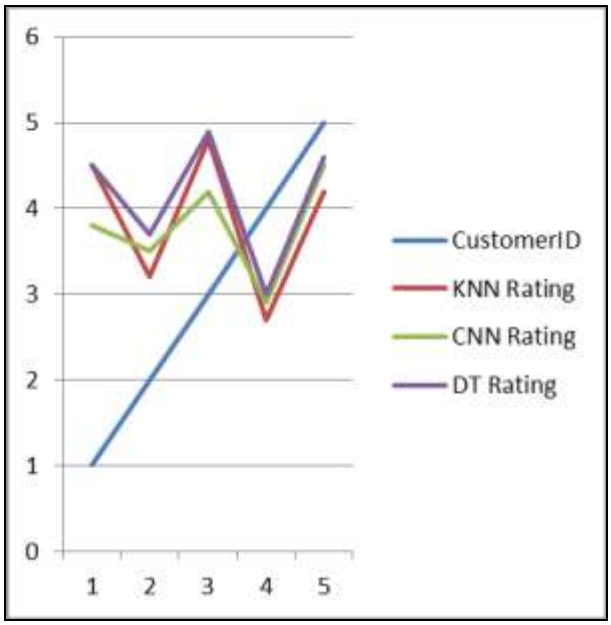

Figure IV: Graph showing different algorithms ratings

As shown in Fig IV, the graph generated from the ratings data can provide insights into patterns, trends, and variations in the algorithm's performance. Analyzing the graph can lead to conclusions about the relative effectiveness and performance of the KNN, CNN, and decision tree algorithms in the recommendation system.

IV. RESULT AND DISCUSSION

In this section, we describe the implementation of our tourism recommender system:

We extracted data set from TripAdvisor website as the experimental data. The content was used to predict the POIs for the new user on various ratings and reviews which was constructed from the information present in the tripadvisor website. The tourists from different travel type have been considered. This traveler type includes families, couple, solo, business and friends. In this paper mostly visited POIs are considered based on the reviews information. These POIs have been rated by the tourists like excellent, very good, average, poor, and terrible with rating 5,4,3,2,1 respectively. Those POIs whose reviews and ratings are greater than their mean are listed.

Information related to month wise visits is predicted. It is noticed that some months have more number of visits. From this it is relatively related to get the knowledge of more number of attractions during those months.

In order to evaluate the performance of our proposed system, we conducted a 10-fold cross- validation approach on the ratings data. The dataset was divided into 10 parts, with each part used as a testing set while the remaining parts served as the training set. The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were chosen as evaluation metrics.

We also utilized a normalized version of the MAE, known as NMAE, to express errors as percentages of the full scale. The NMAE considers the maximum and minimum values of the ratings in the dataset. Furthermore, we applied the RMSE metric, which is commonly used in recommender system evaluations and provides a more severe penalization for larger errors.

Algorithm	**MAE**	**RMSE**
KNN	0.18	0.374
CNN	0.22	0.275
Decision Tree	0.14	0.141

Table V: Showing MAE and RMSE values

Comparing the results, we found that the Decision Tree algorithm achieved the lowest MAE and RMSE values, as shown in Table V, indicating higher accuracy in the recommendation results. The KNN algorithm and CNN algorithm showed relatively higher MAE and RMSE values, suggesting lower accuracy in comparison.

V. CONCLUSION

Based on the results, the Decision Tree algorithm achieved the lowest MAE and RMSE values, indicating higher accuracy in predicting ratings compared to the other algorithms. The KNN algorithm and CNN algorithm showed relatively higher MAE and RMSE values. These findings suggest that the Decision Tree algorithm has potential for effectively predicting ratings in the context of a tourist recommendation system. Also, a comparative analysis of ratings generated by different algorithms is shown in this paper. The system provides users with personalized recommendations and allows for evaluation of algorithm performance based on the collected ratings.

REFERENCES

[1] Torres-Ruiz, M., Mata, F., Zagal, R., Guzmán, G., Quintero, R. & Moreno-Ibarra, M. (2020). A recommender system to generate museum itineraries applying augmented reality and social-

sensor mining techniques. *Virtual Reality, 24*, 175-189.

[2] Biehler, R. & Fleischer, Y. (2021). Introducing students to machine learning with decision trees using CODAP and Jupyter Notebooks. *Teaching Statistics, 43*, S133-S142.

[3] Abdi1, M. H. (2018). Matrix factorization techniques for context-aware collaborative filtering recommender systems: A survey. *Kenya : Canadian Center of Science and Education*.

[4] Srivastava, S., Haroon, M. & Bajaj, A. (2013, September). Web document information extraction using class attribute approach. In: *4th International Conference on Computer and Communication Technology (ICCCT)*, pp. 17-22. IEEE.

[5] Bansari Patel, P. D. (2017). Methods of recommender system: A review. *International Conference on Innovations in information Embedded and Communication Systems (ICIIECS), India*.

[6] Khan, R., Haroon, M. & Husain, M. S. (2015, April). Different technique of load balancing in distributed system: A review paper. In: *Global Conference on Communication Technologies (GCCT)*, pp. 371-375. IEEE.

[7] Khan, W. & Haroon, M. (2022). An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks. *International Journal of Cognitive Computing in Engineering, 3*, 153-160.

[8] Duan, Z. (2020). Personalized tourism route recommendation based on user's active interests. *21st IEEE International Conference on Mobile Data Management (MDM)*.

[9] FengshengZeng, Y. Z. (2020). Tourism recommendation system based on China: *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC*.

[10] Haymontee Khan, N. M. (2017). Tourist spot recommendation system using fuzzy inference system. *13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Bangledesh*.

[11] Huihui Hu, X. Z. (2017). *Recommendation of tourist attractions based on slope one algorithm*. China: 978-1-5386-3022-8/17. IEEE.

[12] Hybrid Recommender System for Tourism Based on Big Data and AI: A Conceptual Framework. (2021). Morocco: *Big data mining and analytics*.

[13] JyotirmoyGope, S. K. (2017). A survey on solving cold start problem in Recommender Systems. *International Conference on Computing, Communication and Automation (ICCCA2017), India*.

[14] Tripathi, M. M., Haroon, M., Khan, Z. & Husain, M. S. (2022). *Security in digital healthcare system. pervasive healthcare: A compendium of critical factors for success*, 217-231.

[15] Scholar, P. G. (2021). *Satiating A user-delineated time constraints while scheduling workflow in cloud environments*.

[16] Husain, M. S. & Haroon, D. M. (2020). An enriched information security framework from various attacks in the IoT. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*.

[17] Khan, W. (2021). An exhaustive review on state-of-the-art techniques for anomaly detection on attributed networks. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*(10), 6707-6722.

[18] Lu Peng, A. M. (2021). Spark-based distance weighted recommendation. *IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), China*.

[19] Husain, M. S. (2020). A review of information security from consumer's perspective especially in online transactions. *International Journal of Engineering and Management Research, 10*.

[20] Khan, N. & Haroon, M. (2022). *Comparative study of various crowd detection and classification methods for safety control system*. Available at: SSRN 4146666.

[21] M ViswaMurali, V. T. (2019). A collaborative filtering based recommender system for suggesting new trends in any domain of research. *5th International Conference on Advanced Computing & Communication Systems (ICACCS), India*.

[22] Haroon, M., Tripathi, M. M. & Ahmad, F. (2020). Application of machine learning in forensic science.in critical concepts, standards, and techniques in cyber forensics. *IGI Global*, pp. 228-239.

[23] Shakeel, N., Haroon, M. & Ahmad, F. (2021). A study of wsn and analysis of packet drop during transmission. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*.

[24] Banik, R. (2018). *Hands-on recommendation systems with Python: start building powerful and personalized, recommendation engines with Python*. Packt Publishing Ltd.

[25] Sowmya, S. & Bab, K. R. (2020). A decision tree based recommended system for tourism. *Journal*

of Composition Theory, Computer Science, 13(12), 27-42.

[26] Nitu, P., Coelho, J. & Madiraju, P. (2021). Improvising personalized travel recommendation system with recency effects. *Big Data Mining and Analytics, 4*(3), 139-154.

[27] Taneja, S. B., Douglas, G. P., Cooper, G. F., Michaels, M. G., Druzdzel, M. J. & Visweswaran, S. (2021). Bayesian network models with decision tree analysis for management of childhood malaria in Malawi. *BMC Medical Informatics and Decision Making, 21*(1), 1-13.

[28] Orama, J. A., Borràs, J. & Moreno, A. (2021). Combining cluster-based profiling based on social media features and association rule mining for personalised recommendations of touristic activities. *Applied Sciences, 11*(14), 6512.

[29] Christodoulou, E., Gregoriades, A., Pampaka, M. & Herodotou, H. (2020). Combination of topic modelling and decision tree classification for tourist destination marketing. In: *Advanced Information Systems Engineering Workshops: CAiSE 2020 International Workshops, Grenoble, France*, pp. 95-108. Springer International Publishing.

[30] Zhu, E., Ju, Y., Chen, Z., Liu, F. & Fang, X. (2020). DTOF-ANN: An artificial neural network phishing detection model based on decision tree and optimal features. *Applied Soft Computing, 95*, 106505.