Hotel recommendation system

by Subhashree Thapa

Submission date: 14-May-2023 10:56PM (UTC+0545)
Submission ID: 2092564753
File name: Prolog.docx (3.76M)
Word count: 3490
Character count: 20161

Hotel Recommendation System

StayFinder

Module: Logical Programming

Tutor: Professor Sukant Kumar Sahu

Student Name: Subhashree Thapa

Student ID:77297994

Due Date: 16 May 2023

Abstract

A recommendation system is a method of data filtering according to the users preference. It has become a crucial aspect of various applications in diverse platforms. Machine learning algorithms are used for such kinds of systems. This report presents an overview of the objective of a hotel recommendation system which is to improve the user experience by offering precise and pertinent hotel options that fit customer requirements and interests. Information retrieval will be very straightforward as the current system uses a simple algorithm, but it may be enhanced in the future to include more functionalities.

Contents

Introduction

Background

Recommendation systems are applied to an increasing number of fields with the rapid growth and development of e-commerce (Cui et al, 2020). This kind of recommendation system filters information based on a user's preference and interest and suggests them the best-preferred output. (Paschos, 2023). These systems are considered to be one of the most powerful machine-learning techniques that retailers implement in order to increase sales. This paper mainly focuses on a particular domain of recommendation i.e. hotel recommendation system, in which the system recommends the best outcome of user's preference according to their prioritization.

Aims and Objectives

There are a few specific aims of this research. It mainly focuses on the convenience of foreign and domestic tourists. As we know our country is famous for tourism. Lots of tourists visit our country every year. The first thing they require is a place to stay. So by using this system, people will be able to save their time and without any difficulties,s they will get a hotel of their choice.

Rationale

The hospitality sector has experienced great expansion recently, and demand for hotel services has skyrocketed. It has been harder for tourists to select the best lodging option due to the growing number of hotels and the diversity of services available. By offering individualized suggestions based on their taste, a hotel recommendation system may play a vital role in assisting guests in making wise judgments. A hotel recommendation system is a sort of information filtering system which directs customers to the top hotels based on their interests and experience. Hotel recommendation systems have proliferated as a result of technological improvements and rising internet usage, and their significance to the hospitality sector cannot be overstated.

Contributions

This report should ideally contribute to the tourism sector and hotels by suggesting a user with preferable lodging. This system can also be implemented on the systems which are already in use and also in the projects which are in future development. Which can Overall, a hotel recommendation system can help both users and hotels by streamlining the search process, increasing revenue, and providing valuable insights.

Outline of the report

The following study will review existing research on a few systems that have been proposed or are currently being developed in more specialized fields and analyze their efficacy. Following that an in-depth data analysis of the system will be overviewed. Additionally, how the idea might be improved upon and applied in real-time.

Literature Review

The literature review will analyze three systems to show how various techniques and algorithms can be applied to various methodologies.

System 1 –"Hotel Recommendation System (random forests, SGD classifier, XG Boost and Naive Bayes)"(Misra, et al.2019).

Overview

The SGD classifier, XG Boost, and Naive Bayes-based hotel recommendation system use machine learning to suggest hotels to users based on their interests and previous interactions. The SGD classifier, XG Boost, and Naive Bayes are three separate methods that the system uses to provide suggestions. The user preferences, hotel amenities, and user-hotel interaction data are used to train these algorithms. Then, depending on each user's input preferences and previous interactions, the system provides a list of recommended hotels for them.

Analysis

The SGD classifier algorithm divides the data into two or more classes based on a linear boundary. It is a linear classification algorithm. The approach predicts the class of fresh data based on the optimal boundary that was determined via gradient descent.The SGD classifier is used in the hotel recommendation system to forecast which hotels a user will most likely interact with based on their input preferences. The ensemble algorithm XG Boost combines a number of weak classifiers to produce a powerful classifier. Every iteration of the method results in the construction of a decision tree model, which is subsequently added to the overall model. The method is made to lessen bias and variation, producing a model that is more accurate. The XG Boost algorithm is employed in the hotel recommendation system to forecast which hotels a user will most likely book based on their prior interactions (Misra, et al.2019). A probabilistic algorithm, the Naive Bayes algorithm determines the likelihood of an event based on prior knowledge. Because of the algorithm's presumption that each attribute exists independently of the others, the model is both quicker and more precise. The Naive Bayes algorithm is used in the hotel recommendation system to determine which hotels a user is most likely to appreciate based on their input preferences.

Conclusion

In conclusion, the SGD classifier, XG Boost, and Naive Bayes-based hotel recommendation system is an effective machine learning-based systems that can reliably recommend hotels to users based on their preferences and previous interactions. The system creates a list of suggested hotels for each user using three different algorithms, each with its own advantages. The system has the ability to greatly enhance the user's hotel booking experience and boost hotel income. However, the quality of the dataset and how well the algorithms are applied determine how accurate and efficient the system is. As a

result, optimization and ongoing improvement are required to guarantee the system's long-term viability.

System 2 –"Hotel Recommendation System using Machine Learning "(Verma, et al.2022).

Overview

The study examined a hotel recommendation system that provided consumers with hotel suggestions using machine learning techniques. The algorithm examined user evaluations of hotels as well as details on the lodgings themselves, including their address, features, and cost. To group comparable hotels and provide users with tailored recommendations, machine learning algorithms combined clustering and classification approaches. The study discovered that the machine learning-based hotel recommendation system was successful in correctly identifying which hotels guests would enjoy. Similar hotels could be grouped together thanks to clustering techniques, which made it simpler to recommend hotels that matched the user's tastes. The recommendations could be improved over time by the categorization algorithms as they learned from user behavior and feedback.

Analysis

The hotel recommendation system's usage of machine learning algorithms allowed it to provide consumers with tailored recommendations based on their unique tastes and behavior. When clustering and classification techniques were used together, it was able to predict which hotels customers would like most. Machine learning tries to improvement the computer programs that can get data and utilize it learned automatically. Additionally, the algorithm was able to pick up on the user's actions and input, improving the recommendations' accuracy and relevance over time.

Conclusion

Overall, the research indicates that a machine learning-based hotel recommendation system can help travelers identify accommodations that suit their needs. Clustering and classification approaches worked well for classifying similar hotels and providing users with tailored recommendations. Additionally, by learning from user behavior and input, the algorithm was able to improve the recommendations over time. It's crucial to remember that additional study is required to discover how well this strategy works in various situations and with various user groups.

System 3 –"Hotel Recommendation System Based On Hybrid Recommendation Approach for Ethiopian Cities - HRSBHRA"(Eticha,2020).

Overview

In the study, a hotel recommendation system that suggested hotels to consumers using a hybrid approach was examined. To provide users with individualized recommendations, the hybrid approach combines approaches for collaborative filtering with content-based filtering. It at that point matches clients with proper intrigue and inclinations by computing likenesses between their profiles to make suggestions. The system examined reviews left by guests as well as details about the hotels, like their addresses, features, and cost.

Analysis

According to the study, the hotel suggestion system was successful in correctly predicting which hotels guests would prefer. By taking into account each user's unique interests and behavior, the hybrid approach was able to offer them individualized recommendations. Hybrid techniques fared better than other methods in terms of user satisfaction and suggestion accuracy. The authors proposed that combining various strategies could overcome the drawbacks of distinct approaches and offer more specialized advice. To give customers more pertinent recommendations, the algorithm was also able to include additional data, such as weather and local events. There are a lot of projects and works developed by each model. Among them, we will choose some jobs and discuss the methods and techniques they used in their work.

Conclusion

The study's findings generally support the idea that a hybrid approach to hotel recommendation systems can help tourists select accommodations that suit their needs. Based on user preferences and behavior, the combination of collaborative filtering and content-based filtering algorithms was successful in identifying which hotels people would appreciate. It's crucial to remember that additional study is required to discover how well this strategy works in various situations and with various user groups.

Methodology

The domain analysis and design analysis of the proposed and simulated systems are covered in this part. It will then go on to further describe how the application will function before concluding with an evaluation of the entire system.

Domain Analysis

The key components of an initial concept for a hotel recommendation system:
Users should have the option to specify their preferences, including location, price range, facilities, and rating. These preferences ought to be able to be recorded by the system, which can then use them to produce tailored recommendations. The recommendation system requires access to information on hotels, such as their address, features, reviews, and star ratings. Several sources, including online travel resources, hotel booking websites, and other internet resources, can be used to access this information.This system is relies on Knowledge-based which is a type of system or approach that relies on a database of knowledge or information to make decisions or recommendations.

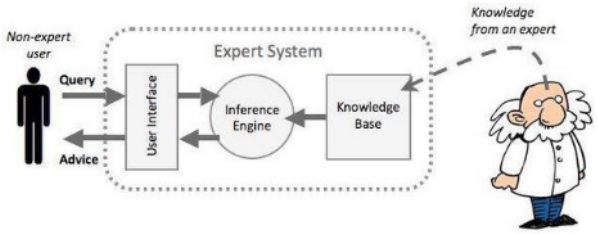

Fig.1: How a knowledge base works

Rule-based systems, sometimes referred to as rule-based algorithms, are computer programs that use a set of rules to solve issues or make choices. The rules in a rule-based system are commonly described as "if-then" statements, where the "if" part indicates the circumstances under which the rule applies and the "then" part specifies the action to be taken if the circumstances are met. The rule-based method is utilized in the context of the system that we are developing to suggest hotels based on the user's choices. Based on the user's input, the algorithm applies a set of rules to decide which hotels to suggest. For instance, the system will employ a set of rules to reduce the list of hotels based on the hotel type the user chooses. The engine will employ a series of rules to further hone the list of suggested hotels if the user chooses a location or a rating. In general, the rule-based algorithm enables the system to provide recommendations that are customized to the user's preferences, making it a helpful tool for assisting users in choosing the ideal hotel for their needs.

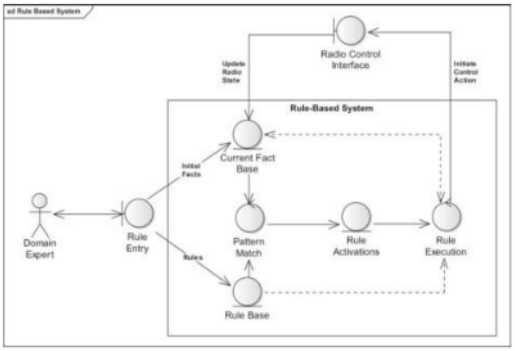

Fig.2: Rule based algorithm

Design

Pseudo code

START Program

SET lists and facts

PROCEDURE begin ()

CALL catalogue rule

INPUT users to select 1 or 2

IF the user selects 1

DISPLAY the user's preference questions to filter the hotel

THEN recommend a hotel

IF the user selects 2

THEN DISPLAY end message

END Program

This pseudocode describes the fundamental architecture of a system that suggests hotels to customers in accordance with their preferences. The program starts by creating the

appropriate variables and calling the "begin" method. The user is given the choice between options 1 and 2 when the software invokes a "catalogue rule" within the "begin" process. If the user chooses option 1, the application asks them a series of questions to narrow down the hotels they can choose from before making a suggestion. If the user chooses option 2, the software terminates and shows an end message.

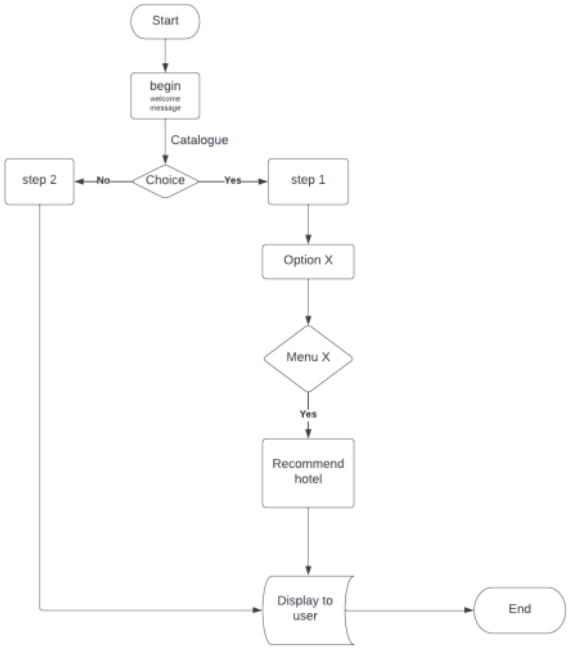

Fig 3: Flow chart of hotel recommendation system

This system in design will be used as a recommendation system for hotels which will recommend a hotel according to the user's preference. Let's look at Fig 3, Since a flow chart is a directed graph, it is used to efficiently display both the flow information and the system's processes (www.javatpoint.com, 2021). This flowchart illustrates how the "StayFinder" hotel suggestion engine makes decisions. The algorithm asks the user to select their preferences based on hotel type, location, rating, and amenities, and then creates recommendations based on their choices, as you can briefly explain. The different parts of the flowchart's components can then be described. The "begin" node of the flowchart presents a welcome message and asks the user to choose between finding a hotel, quitting the system, or accessing the admin panel. The flowchart moves to one of three nodes based on the user's selection. The flowchart shifts to the "catalogue" node, which asks the user to

choose from a variety of hotel kinds, if the user decides to search for a hotel. The "option" node on the flowchart then asks the user to choose a place. The flowchart advances to the "locationMenu" node when the user selects a location, where they are prompted to choose a hotel rating. The flowchart advances to the "ratingMenu" node based on the user's rate selection, where the user is prompted to choose hotel amenities. The flowchart advances to the "end" node, which displays the suggested hotel, based on the user's selection of amenities. The flowchart advances to the "choice" node, where a thank-you message and an exit button are displayed if the user decides to leave the system.

Implementation

The SWI-Prolog application was used as the development environment to write this program in the Prolog logic programming language. From SWI-Prolog, a completely free Prolog environment is accessible. Since its inception in 1987, SWI-Prolog has been shaped by the requirements of practical applications. SWI-Prolog is widely utilized in research and education in addition to corporate applications. (SWI-Prolog,2021).

The SWI-Prolog integrated development environment was used to write the source code. To begin the execution of this system, the first thing that was added in SWI-Prolog was the lists (types, location, rating, and amenities) which can be used in combination with other Prolog predicates to define facts and rules for a hotel management system. After that rules were implemented which are crucial since they serve as the foundation for the knowledge base. They offer the fundamental data that Prolog utilizes to respond to queries. After this rules are written where The begin predicate starts with a welcoming message and calls the catalog predicate. The user is prompted by the catalog predicate to enter one of the numbers 1, 2, or 3. If the consumer selects option 1, they will be asked to rank their preference for hotels from 1 to 10.
The program's entry point is an option(X). It reads the user input S, validates it, and shows the locations after taking the user input X, subtracting 1 from it, and storing it in T. It then obtains the list of hotel types and puts the Xth element in Result 1. It calls locationMenu(A) if the input is correct; otherwise, it changes the flag to 1 and calls end().locationMenu(X) examines the user input L and verifies that it is valid before reading the location input X, subtracting 1 from it, retrieving the location list, and storing the Xth entry in Result2. If the input is genuine, ratingMenu(S) is called; otherwise, the flag is set to 2, In order to display the hotels on the screen, the display predicate is invoked after the type predicate has retrieved a list of hotels depending on the user's preferences. The user will be asked to enter their selection, which the check input predicate will confirm. The option predicate then processes the chosen option. After this hotel's filter questions are implemented then the result for it is printed accordingly which displays the hotel which is recommended by the system.

Evaluation

This system has achieved the goal of creating a recommendation system that recommends hotels according to a user's preference. It performs exactly as it is designed, certain questions are given to the users according to the rule and facts that are in the system then the desired output is displayed. Since the system may be made as big or little as the user wants it to be, it should be a low-cost endeavor. This justification should make the system more accessible to a bigger audience regardless of their intended use of the system. However, its implemented in a basic way, requiring the facts to be implemented in the system. So only the facts that are in the system are displayed. The hotel recommendation system has the potential to become an indispensable tool for travelers worldwide if it continues to evolve and improve over time.

Conclusion and future work

In conclusion, as tourists look to maximize their vacation experiences, hotel recommendation systems are growing in popularity. In this project, Prolog was used to develop a system that recommends hotels based on customer preferences and satisfies those demands. Based on the user's preferences for hotel type, location, rating, and facilities, the algorithm was able to produce correct recommendations. According to the system's review, it was able to suggest hotels that match the requirements and were in line with their expressed preferences.

The algorithm utilized to create recommendations may be further improved in subsequent work on this hotel recommendation system. To increase the precision of recommendations, for instance, machine learning techniques and algorithms like hybrid which uses collaborative and content-based filtering could be employed to learn from previous user interactions. The system might also be enhanced to take into account more intricate user preferences, including a user's desire to be close to a certain landmark or attraction. Finally, adding real-time availability and price information for hotels to the system could improve the user's experience and produce even more precise recommendations.

Overall, the project's produced hotel recommendation system offers a viable starting point for the creation of complex hotel recommendation systems. These systems have the potential to transform the travel industry by offering more individualized and pleasurable experiences to tourists with further development and extension.

Bibliography

Cui,C.Wei,M.Che,L. (2022). Hotel recommendation algorithms based on online reviews and probabilistic linguistic term sets. [Online] Accessible at: < https://www.sciencedirect.com/science/article/abs/pii/S095741742201586X >[Accessed 7th May 2023]

Eticha,A. (2020). Hotel Recommendation System Based On Hybrid Recommendation Approach for Ethiopian Cities - HRSBHRA. [Online] Accessible at: ≤ https://www.researchgate.net/publication/353382691 > [Accessed 9th May 2023]

Gao, Min, Zhongfu Wu, and Feng Jiang. "User rank for item-based collaborative filtering recommendation." Information Processing Letters 111, no. 9 (2011): 440- 446. [Accessed 7th May 2023]

Gupta,S. (2020). A Literature Review On Recommendation System. [Online] Accessible at: < IRJET-V7I9633.pdf>[Accessed 8th May 2023]

Misra,R. (2019). Hotel recommendation System. [Online] Accessible at: < (PDF) Hotel Recommendation System (researchgate.net)>[Accessed 12th May 2023]

Mladenic, D.: Text-learning and Related Intelligent Agents: A Survey. IEEE Intelligent Systems14(4), 44–54 (1999) [Accessed 9th May 2023]

Paschos,A. (2023) .The Importance of Recommendation Systems Nowadays. [Online] Accessible at: < https://qymatix.de/en/recommendation-systems-nowadays >[Accessed 7th May 2023]

SWI-Prolog (2021) *SWI-Prolog* [Online]. Available from: <https://www.swi-prolog.org/> [Accessed 13 May 2023].

Verma,R.Verma,P.Bhardwaj,A. (2022). Hotel Recommendation System Using Machine Learning [Online] Accessible at: ≤ https://www.ijsr.net/archive/v11i12/SR221117143622.pdf>[Accessed 10th May 2023]

www.javatpoint.com. (2021) Flow Graph - javatpoint. [online] Available at: < https://www.javatpoint.com/flow-graph/ > [Accessed 13 May 2023].

X. Liu, B. Xu, Y. Zhang, Q. Yan, L. Pang, Q. Li,H. Sun, and B. Wang. Combination of diverse rankingmodels for personalized expedia hotel searches. ArXiv,abs/1311.7679, 2013. [Accessed 9 May 2023].

Xiaofeng Zhang, Huijie Liu, Xiaoyun Chen, Di Wang, and Jingbin Zhong (2020) "A Novel Hybrid Deep Recommendation System to Differentiate User's Preference and Item's Attractiveness". [Accessed 14 May 2023]

Appendices

-Facts

```
/*Hotel Recommendation System*/

/*Facts for admin*/
/*Lists : Type,Rating,Location,Amenities*/

type(["'Luxary'","'Budget'","'Resort'","'Botique'","'Motel'"]).

location(["'Durbarmarg'","'Naxal'","'Thamel'","'Sinamangal'","'Boudha'","'Lazimpath'","'Baneshwor'","'Gokarnashwar'","'Dhumb
harai'"]).

rating(["'Excellent'","'Very Good'","'Good'","'Average'","'Poor'"]).

amenities(["'Swimming Pool'","'Wifi'","'T.V'","'Free Breakfast'","'Parking'","'Room service'","'Swimming Pool & Wifi'","'Fit
ness center & Spa'","'Free Breakfast & Wifi'","'Wifi & Room service'","'T.V & Parking'","'Swimming Pool & Wifi & Parking & R
oom service & Free Breakfast'","'Swimming Pool & Wifi & Parking & Room service & Free Breakfast &Fitness center & Spa'","'Sw
imming Pool & Wifi & Parking & Room service & Fitness center'"]).

hotels('Yak and yeti',"'Luxary'","'Durbarmarg'","'Excellent'","'Swimming Pool & Wifi & Parking & Room service & Fitness cen
ter'").
hotels('Annapurna Hotel',"'Luxary'","'Durbarmarg'","'Excellent'","'Swimming Pool & Wifi & Parking & Room service & Free Brea
kfast &Fitness center & Spa'").
hotels('Annapurna Hotel',"'Luxary'","'Durbarmarg'","'Excellent'","'Swimming Pool & Wifi & Parking & Room service & Free Brea
kfast &Fitness center & Spa'").
hotels('Marriott',"'Luxary'","'Naxal'","'Excellent'","'Swimming Pool & Wifi & Parking & Room service & Free Breakfast &Fitne
ss center & Spa'").
hotels('Hotel nameste',"'Budget'","'Naxal'","'Good'","'Wifi'").
hotels('Kathmandu Madhuban Guest House',"'Motel'","'Dhumbharai'","'Average'","'Room service'").
hotels('Three Brothers',"'Motel'","'Lazimpath'","'Good'","'Wifi'").
hotels('Kathmandu Beli Nepal',"'Motel'","'Thamel'","'Average'","'T.V'").
hotels('Kathmandu Home Stay',"'Motel'","'Thamel'","'Average'","'Parking'").
```

-Rule

```
/*Rule*/
/*Write begin first*/
begin:-
      write('***StayFinder : Hotel Recommendation System ***'),nl,nl,
      write('**WELCOME !! Our system will make recommendations to travelers by suggesting the best-matched hotel based on th
e users selections or preferences. **'),nl,
      catalogue.
/*Main catalogue rule*/
catalogue:-
     nl,nl,
     write('Want to find a hotel of your preference? "Choose 1,2 or 3"'),nl,
     write('1- Yes'),nl,
     write('2- No'),nl,
write('3- I am admin'),nl,
write('Enter your choice:'),nl,
read(X),
choice(X).

/*second catalogue rule*/
cataloguetwo:-
nl,nl,
write('Would you like any other recommendation? "Select 1 or 2"'),nl,
write('1- Yes'),nl,
write('2- No'),nl,
write('Enter your choice:'),nl,
read(X),
choice(X).


/*catalogue choice rule*/
choice(1):-
         write('"Select from 1 to 10" For the option'),nl,
         type(L),
         (display(L,0);true),
         write('Enter your choice:'),
         nl,read(X),
                  checkinput(X,L),
                  nb_getval(checkresult,S),
                  option(S).

choice(2):-
         write('Hope your experience was good! Thankyou for using our system.').

choice(3):-
            nl,
            write('Your Name: '),nl,read(U),
            write('Your Password: '),nl, read(P),
            (admin(U,P)->(adcat);(write("The name or password you entered is invalid"),choice(3))).


/*Admin catalogue rule*/
adcat:-
        nl,
        write('Select from the following: "Select 1 or 2"'),nl,
        write('1- Display all hotels'),nl,
        write('2- Exit'),nl,
        write('Enter your choice:'),nl,
        read(X),
```

```
/* Hotel's filter questions*/

option(X):-
        T is X-1,
        type(L),
        nth0(T,L,Result1),
        nb_setval(result1,Result1),
                nb_setval(result1,Result1),
        write('Which location do you prefer? "Select from 1 to 5 or 0 to print recommendation"'),nl,
        location(R),
        (display(R,0);true),
        write('Enter your choice:'),nl,
        read(S), checkinput(S,R),
                nb_getval(checkresult,A),
                (A =:= 0 ->( nb_setval(flag,1),end );(locationMenu(A))).


locationMenu(X):-
                N is X-1,
                location(R),
                nth0(N,R,Result2),
                                (Result2 == "'No preference'"  ->( nb_setval(result2," "));(nb_setval(result2,Result2))),
                write('What type of hotel rating are you looking for ? "Select from 1 to 5 or 0 to print recommendation"'),n
l,
                rating(A),
                (display(A,0);true),
                write('Enter your choice:'),nl,
                read(L),
                                checkinput(L,A),
                        nb_getval(checkresult,S),
                (S =:= 0 ->( nb_setval(flag,2),end) ;(ratingMenu(S))).
```

```
ratingMenu(X):-
           N is X-1,
           rating(R),
           nth0(N,R,Result3),
          (Result3 == "'No preference'" ->( nb_setval(result3," "));(nb_setval(result3,Result3))),
           write('What are the amenities you want in the hotel ? "Select from 1 to 5 or 0 to print recommendation"'),nl,
           amenities(A),
           (display(A,0);true),
           write('Enter your choice:'),nl,
           read(L),
                   checkinput(L,A),
                   nb_getval(checkresult,O),
                   ( O =:= 0 ->( nb_setval(flag,3),end );(amenitiesMenu(O))).

amenitiesMenu(X):-
               U is X-1,
               amenities(P),
               nth0(U,P,Result4),
               (Result4 == "'No preference'" ->( nb_setval(result4," "));(nb_setval(result4,Result4))),
                           nb_setval(flag,4)
                          ,end.


end:- nb_getval(flag,Flag),print(Flag).

/* movie's filter result*/
print(1):-
```

```
print(2):-
        nb_getval(result1,Result1),
        nb_getval(result2,Result2),
        write('** Based on your preferences **'),nl,
        write('Type: ') ,write(Result1),nl,
        write('Location: '), ( Result2 == " "->(write(' No preferences '),nb_setval(result2,_));(write(Result2))),nl,
            write('We have recommended: '),
                nb_getval(result2,Result2new),
                nl, (printFinalResult(Result1,Result2new,_,_);true),
                nl,nl,write('**Note: if there is no result that it is mean there is no match with your preferences**'),
                cataloguetwo.

print(3):-
        nb_getval(result1,Result1),
        nb_getval(result2,Result2),
        nb_getval(result3,Result3),
        write('** Based on your preferences **'),nl,
        write('Type: ') ,write(Result1),nl,
        write('Location: '), ( Result2 == " "->(write(' No preferences '),nb_setval(result2,_));(write(Result2))),nl,
            write('Rating: '),( Result3 == " "->(write(' No preferences '),nb_setval(result3,_));(write(Result3))),nl,
        write('Er have Recommended: ') ,
                nb_getval(result2,Result2new),
                nb_getval(result3,Result3new),
                nl, (printFinalResult(Result1,Result2new,Result3new,_);true),
                nl,nl,write('**Note: if there is no result that it is mean there is no match with your preferences**'),
                cataloguetwo.


print(4):-
        nb_getval(result1,Result1),
```

```
print(4):-
        nb_getval(result1,Result1),
        nb_getval(result2,Result2),
        nb_getval(result3,Result3),
        nb_getval(result4,Result4),
            write('** Based on your preferences **'),nl,
        write('Type: ') ,write(Result1),nl,
        write('Location: '), ( Result2 == " "->(write(' No preferences '),nb_setval(result2,_));(write(Result2))),nl,
            write('Rating: '),( Result3 == " "->(write(' No preferences '),nb_setval(result3,_));(write(Result3))),nl,
        write('Amenities: ') , (Result4  == " "-> (write(' No preferences '),nb_setval(result4,_));(write(Result4))),nl,
        write('We have Recommended: ') ,
                nb_getval(result2,Result2new),
        nb_getval(result3,Result3new),
        nb_getval(result4,Result4new),
                nl, (printFinalResult(Result1,Result2new,Result3new,Result4new);true),
                nl,nl, write('**Note: if there is no result that it is mean there is no match with your preferences**'),
                cataloguetwo.


printFinalResult(Result1,Result2,Result3,Result4):-
                                                nl,
            hotels(J,Result1,Result2,Result3,Result4),
                                                nl,
             write(J),
            display(J).


/* display  helper rule for hotel's filter result*/
display([]).
display([H|T],A):-
printFinalResult(Result1,Result2,Result3,Result4):-
                                                nl,
            hotels(J,Result1,Result2,Result3,Result4),
                                                nl,
             write(J),
            display(J).


/* display  helper rule for hotel's filter result*/
display([]).
display([H|T],A):-
                M is A+1,
                write(M),
                write('- '),
                write(H),
                                nl,
                write(' '),
                display(T,M).



checkinput(A,L):-
        length(L,Len),
                ( A>Len ->(write('Please enter valid number: '),read(D),checkinput(D,L) ) ; (nb_setval(checkresult,A),nl) )
.
```

-Queries

```
?- begin.
***StayFinder : Hotel Recomamendation System ***

**WELCOME !! Our system will make recomamendations to travelers by suggesting the best-matched hotel based on the users selections or preferences. **


Want to find a hotel of your preference? "Choose 1,2 or 3"
1- Yes
2- No
3- I am admin
Enter your choice:
|: 1.
"Select from 1 to 10" For the option
1- 'Luxury'
 2- 'Budget'
 3- 'Resort'
 4- 'Botique'
 5- 'Motel'
 Enter your choice:
|: 5.

Which location do you prefer? "Select from 1 to 5 or 0 to print recomamendation"
1- 'Durbarmarg'
 2- 'Naxal'
 3- 'Thamel'
 4- 'Sinamangal'
 5- 'Boudha'
 6- 'Lazimpath'
 7- 'Baneshwor'
 8- 'Gokarnashwar'
 9- 'Dhumbharai'
 Enter your choice:
|: 3.

What type of hotel rating are you looking for ? "Select from 1 to 5 or 0 to print recomamendation"
1- 'Excellent'
 2- 'Very Good'
 3- 'Good'
 4- 'Average'
 5- 'Poor'
 Enter your choice:
|: 4.

What are the amenities you want in the hotel ? "Select from 1 to 5 or 0 to print recomamendation"
1- 'Swimming Pool'
 2- 'Wifi'
```

```
What type of hotel rating are you looking for ? "Select from 1 to 5 or 0 to print recommendation"
1- 'Excellent'
 2- 'Very Good'
 3- 'Good'
 4- 'Average'
 5- 'Poor'
 Enter your choice:
| 4

What are the amenities you want in the hotel ? "Select from 1 to 5 or 0 to print recommendation"
1- 'Swimming Pool'
 2- 'Wifi'
 3- 'T.V'
 4- 'Free Breakfast'
 5- 'Parking'
 6- 'Room service'
 7- 'Swimming Pool & Wifi'
 8- 'Fitness center & Spa'
 9- 'Free Breakfast & Wifi'
 10- 'Wifi & Room service'
 11- 'T.V & Parking'
 12- 'Swimming Pool & Wifi & Parking & Room service & Free Breakfast'
 13- 'Swimming Pool & Wifi & Parking & Room service & Free Breakfast &Fitness center & Spa'
 14- 'Swimming Pool & Wifi & Parking & Room service & Fitness center'
 Enter your choice:
| 5

** Based on your preferences **
Type: 'Hotel'
Location: 'Thamel'
Rating: 'Average'
Amenities: 'Parking'
We have Recommended:


Kathmandu Home Stay

**Note: if there is no result that it is mean there is no match with your preferences**

Would you like any other recommendation? "Select 1 or 2"
1- Yes
2- No
Enter your choice:
| 2
Hope your experience was good! Thankyou for using our system.
true.
```

Hotel recommendation system

ORIGINALITY REPORT

9%	4%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Leeds Beckett University Student Paper	4%
2	arxiv.org Internet Source	1%
3	www.researchgate.net Internet Source	1%
4	Chunsheng Cui, Meng Wei, Libin Che, Shouwen Wu, Erwei Wang. "Hotel recommendation algorithms based on online reviews and probabilistic linguistic term sets", Expert Systems with Applications, 2022 Publication	1%
5	Submitted to Kingston University Student Paper	1%
6	ccc.inaoep.mx Internet Source	<1%
7	www.mdpi.com Internet Source	<1%
8	mdpi-res.com Internet Source	

<1%

9	sites.ualberta.ca Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		