**CS 571 – Artificial Intelligence**

**Movies and Series Knowledge-Based Expert System**

**Project Report**

**Brindavana Sachidanand**

**200350012**

[**sachi002@cougars.csusm.edu**](mailto:sachi002@cougars.csusm.edu)

**Project Completed**

**Check List**

* **YES Did you follow all the requirements in implementing the system?**
* **YES Did you create one Word file of your report with the cover page and section headers as specified?**
* **YES Did you answer all questions per section?**

**Section 1 – Purpose of the Expert System**

1. **Introduction and Domain**: My Project is a Movie and Series Knowledge-Based Expert System that recommends Movies and Series to watch based on the choices that the user provides. The Expert System is for **Selection** i.e., Selection of movies and series based on user choices.

2. **Goal**: The goal of the program is to list out the movies or series that the user can watch based on the choice of input provided by the user. We start off by asking the user if he wants to watch a movie or a series.

If he chooses movies as the option, then the user has to enter his Age, Mood, Genre interested, Length of the movie he is interested in and his Subscription.

If he chooses series as the option, then the user has to enter his Age, Mood and Genre interested to watch and Subscription.

Based on the values entered by the user, a particular rule is triggered and gives a required output. The possible values of output for movies could be

You can watch A Star is Born, La La Land

You can watch Lion King, Despicable Me

You can watch Conjuring, The Grudge

Etc.

The possible values of output for series could be

You can watch Money Heist, Sherlock

You can watch Friends, Big Bang Theory

You can watch Upload, The Mandalorian

Etc.

**User Input and Output**

Let us consider an example.

Let us say that the user chose to watch a movie, then the user has to enter the following as input.

? – top.

Do you want to watch a movie or series? movie.

Age? 20.

Subscription? prime.

Genre? romance.

Mood? happy.

Length? 2.

Based on the above input, the output is as follows.

You can watch A Star is Born, La La Land

Since the user input is movie, age is > 18, subscription is prime, genre is romance, mood is happy and length is <= 2, the following rule gets chosen and gives the required output

choose\_movie(Age, Genre, Mood, Length, Subscription, 'A Star is Born, La La Land') :-

adult(Age), romance(Genre), happy(Mood), ltwo(Length), prime(Subscription).

3. **BWC, DFS is suitable for this problem domain rather than FWC**

**Forward chaining** (FWC) as the name suggests, starts from the known facts and moves **forward** by applying inference rules to extract more data, and it continues until it reaches to the goal, whereas **backward chaining** (BWC) starts from the goal, move **backward** by using inference rules to determine the facts that satisfy the goal.

In BWC, we start with something that we have to find, and we explore a path to the goal state before exploring other paths. We keep going until we either find a solution or hit a dead end (no solution) and DFS gives a guaranteed solution. BWC is more efficient and faster than FWC as it tests just fewer rules compared to FWC. BWC is more suited for our problem domain because we start with a main and then go to sub goals and ultimately get the result.

FWC isn’t suitable because it tests all the rules and we may end up getting infinite number of conclusions as FWC’s aim is to get any number of conclusions possible but BWC only searches for the required goal and selects only those rules that are required to reach the goal state.

**Section 2 – Implementation**

1. Facts: I have not used any facts in my program.

2. Rules: I have 3 kinds of rules: **User Input Rules**, **Main Choose Rules** and **Sub Goal Rules**

i. User Input Rules: These rules are used to accept input from the user. The first rule is the **top** rule. When the user enters top, he is asked to choose whether he wants to watch a movie or series. All these are handled by the user input rules. If user wants to watch a movie, the **topm** rule is triggered which further triggers the rule choose\_movie once user inputs his choices. If user wants to watch a series, then the **tops** rule is triggered which further triggers the choose\_series rule once user inputs his choices. choose\_movie and choose\_series are Main Choose Rules

ii. Main Choose Rules: **choose\_movie** rule is chosen for a movie and **choose\_series** rule is chosen for a series. These are main rules are for deciding which movies or series the user can watch based on the input the entered. There are many genres and we have rules for every genre entered and further based on Age, Mood, Length and Subscription. If the input entered by user doesn’t match the rule, then we have a default output, each for movies and series.

Example of Main Chose Rules:

**choose\_movie(Age, Genre, Mood, Length, Subscription, 'Lion King, Despicable Me') :- kid(Age), animation(Genre), happy(Mood), ltwo(Length), disney(Subscription).**

**choose\_series(Age, Genre, Mood, Subscription, 'Outlander, Black Mirror') :-**

**adult(Age), scifi(Genre), happy(Mood), netflix(Subscription).**

iii. Sub Goal Rules: There are some sub-goal rules that get triggered because of the main rules. The sub-goal rules are for assigning the Genre to a rule, for deciding whether kid or adult based on age, for deciding which genre movies can be watched by kids and adults, for assigning the moods happy and sad, for deciding whether the length of the movie is greater than/lesser than 2 hours and for assigning the Subscription.

If Age > 18, then it’s an adult, else it’s a kid

**adult(Age) :- Age > 18.**

**kid(Age) :- Age =<18.**

Mood could be happy or sad

**happy(Mood) :- Mood = happy.**

**sad(Mood) :- Mood = sad.**

There are many genres like thriller, comedy, action, horror, scifi, animation, romance, fantasy, horror\_comedy.

**thriller(Genre) :- Genre = thriller.**

**comedy(Genre) :- Genre = comedy.**

**action(Genre) :- Genre = action.**

**horror(Genre) :- Genre = horror.**

**scifi(Genre) :- Genre = scifi.**

**animation(Genre) :- Genre = animation.**

**romance(Genre) :- Genre = romance.**

**fantasy(Genre) :- Genre = fantasy.**

**horror\_comedy(Genre) :- Genre = horror\_comedy.**

There are two special sub goal rules that choose Genre and Age as something specific and these rules could be used in our main choose rules

**adult\_horror(Age, Genre) :- adult(Age), Genre = horror.**

**kid\_scifi(Age, Genre) :- kid(Age), Genre = scifi.**

Length of the movie could be > 2 or < = 2

**gtwo(Length) :- Length > 2.**

**ltwo(Length) :- Length =< 2.**

Subscription could be Netflix, Prime or Disney

**netflix(Subscription) :- Subscription = netflix.**

**prime(Subscription) :- Subscription = prime.**

**disney(Subscription) :- Subscription = disney.**

Based on the above inputs, the program provides respective suitable output.

**Flow of the program**

🡺 User enters top.

🡺 It goes to the top rule and asks user whether he wants to watch a movie or a series

top :- choice, read\_input.

choice :- write('Do you want to watch a movie or series? ').

read\_input :- read(Option), process\_input(Option).

process\_input(Option) :- (Option == movie), topm.

process\_input(Option) :- (Option == series), tops.

🡺 topm chooses movies

🡺 tops chooses series

🡺 topm triggers the main rule choose\_movie

🡺 tops triggers the main rule choose\_series

Let us take an example when user chooses to watch a series.

So, he enters series.

The rule choose\_series gets triggered

Let us he enters the following options

Age? 12.

Genre? action.

Mood? happy.

Subscription? netflix.

Then the following rule gets triggered based on the sub goal rules.

choose\_series(Age, Genre, Mood, Subscription, 'Flash, NCIS') :-

kid(Age), action(Genre), happy(Mood), netflix(Subscription).

kid(Age) sub goal rule is triggered as Age is < 18, action(Genre) sub goal rule is triggered as Genre is action, happy(Mood) is triggered as mood is happy, netflix(Subscription) rule is triggered as Subscription is netflix. All these are sub goals. Based on these, the output is displayed as “You can watch Flash, NCIS”

**Section 3 – Testing and Results**

The program runs thoroughly and matches my expectations.

Following are the screenshots of output obtained.

**Test 1**: When none of the rules match the input given by the user, the default is chosen (Lord of the Rings is the default for Movies and Two and a Half Men is the default for Series)

**Movies – Since none of the conditions matched, Lord of the Rings is recommended**

There is no horror movie which belongs to the happy category, thus the expert system gives the output as default.

A screenshot of a video game

Description automatically generated

Similarly, we can have other cases where we get default output

A screenshot of a computer screen

Description automatically generated

**Series – Since none of the conditions matched, Two and a Half Men is recommended**

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Test 2**: Arrives at answer quickly

Whenever the user enters his choices, the expert system will suggest a movie/series

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**Test 3 and Test 4**: Arrives at an answer after some backtracking

We use the trace option on gprolog to do backtracking of a rule and check whether we got the correct recommendation or not

Test 3: **choose\_movie(Age, Genre, Mood, Length, Subscription, 'The Tourist, Inception') :-**

**adult(Age), thriller(Genre), happy(Mood), gtwo(Length), netflix(Subscription).**

A screenshot of a cell phone

Description automatically generated

**choose\_movie(Age, Genre, Mood, Length, Subscription, 'Maleficent: Mistress of Evil, Shazam') :-**

**kid(Age), fantasy(Genre), happy(Mood), gtwo(Length), netflix(Subscription).**

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screen shot of a computer

Description automatically generated

A picture containing sitting, table, black, holding

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Test 4: **choose\_series(Age, Genre, Mood, Subscription, 'Stranger Things, Supergirl') :-**

**kid\_scifi(Age, Genre), happy(Mood), netflix(Subscription).**

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**choose\_series(Age, Genre, Mood, Subscription, 'The Vampire Diaries, The Purge') :-**

**adult\_horror(Age, Genre), sad(Mood), prime(Subscription).**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**Section 4 – Confidence Analysis**

MYCIN’s method to calculate confidence of the test results of Test 3 and Test 4

Test 3:

**choose\_movie(Age, Genre, Mood, Length, Subscription, 'The Tourist, Inception') :-**

**adult(Age), thriller(Genre), happy(Mood), gtwo(Length), netflix(Subscription).**

Calculation

= Min CF(adult(Age), thriller(Genre), happy(Mood), gtwo(Length), netflix(Subscription)) \* CF(choose\_movie)

= Min CF (0.9, 1.0, 1.0, 0.9, 1) \* 0.6 = 0.9 \* 0.6

Therefore, Confidence = **0.54**

**choose\_movie(Age, Genre, Mood, Length, Subscription, 'Maleficent: Mistress of Evil, Shazam') :-**

**kid(Age), fantasy(Genre), happy(Mood), gtwo(Length), netflix(Subscription).**

Calculation

= Min CF(kid(Age), fantasy(Genre), happy(Mood), gtwo(Length), netflix(Subscription)) \* CF(choose\_movie)

= Min CF (0.9, 1.0, 1.0, 0.9, 1) \* 0.6 = 0.9 \* 0.6

Therefore, Confidence = **0.54**

Test 4:

**choose\_series(Age, Genre, Mood, Subscription, 'Stranger Things, Supergirl') :-**

**kid\_scifi(Age, Genre), happy(Mood), netflix(Subscription).**

Calculation

Min CF (kid\_scifi(Age, Genre), happy(Mood), netflix(Subscription)) \* CF (choose\_series)

Min CF (0.7, 1.0, 1.0, 1.0) \* 0.6 = 0.7 \* 0.6

Therefore, Confidence = **0.42**

**choose\_series(Age, Genre, Mood, Subscription, 'The Vampire Diaries, The Purge') :-**

**adult\_horror(Age, Genre), sad(Mood), prime(Subscription).**

Calculation

Min CF (adult\_horror(Age, Genre), sad(Mood), prime(Subscription)) \* CF (choose\_series)

Min CF (0.7, 1.0, 1.0, 1.0) \* 0.6 = 0.7 \* 0.6

Therefore, Confidence = **0.42**

**Section 5 – Adding Machine Learning**

1. **Incorporating machine learning into rule-based expert systems in general**

Expert based systems can use the already existing rules and learn from them in order to recommend when a new set of input is given. The expert system could perform better by learning from the already existing rules and it could reach a state where it could develop its own rules whenever a new input is given.

1. **Using Machine Learning in my Expert System**

We have rules for just some conditions in the program and the system gives default output if there are no rules that match the given conditions/input given by the user. In this case, we could use Machine Learning such that the system learns from the already existing rules and further learns to give the correct output when a new set of conditions occur instead of giving a default output. The system could look at the input and recommend something closer if not the exact.

We need to have huge corpus of data containing many movies and series indexed by the Genre, Age, Subscription, Length and Mood and the system could learn to give output when new set of values are entered by the user. Having this huge corpus of data is also important because we do not have to write rules for all the possible conditions and having just few rules would be sufficient. The system can learn from the existing rules and recommend movies and series.