

# Movie Recommendation System using Palm 2



## Overview Report

Shashank Kallahalli Suresh

[skallaha@andrew.cmu.edu](mailto:skallaha@andrew.cmu.edu)

Hai Zheng

[haiz2@andrew.cmu.edu](mailto:haiz2@andrew.cmu.edu)

Jiayi Liang

[jiayilia@andrew.cmu.edu](mailto:jiayilia@andrew.cmu.edu)

Raj Shah

[rbshah@andrew.cmu.edu](mailto:rbshah@andrew.cmu.edu)

Fatima Fazal

[ffazal@andrew.cmu.edu](mailto:ffazal@andrew.cmu.edu)

## TABLE OF CONTENTS

<b>1. Model Building and Testing:</b>	<b>3</b>
<b>1.a. Use Case: Movie Recommendation System Diagram</b>	<b>3</b>
A. User Inputs for Recommendations	3
B. System Generates Movie Recommendations	3
C. Specific Example	4
D. Postconditions	4
<b>1.b. Python Script</b>	<b>5</b>
<b>1.c. ReadMe</b>	<b>8</b>
A. Introduction	8
B. Requirements	8
C. Setup	8
D. Usage	8
E. Workflow	8
F. Note	9
G. Improvements	9
H. Conclusion	9
<b>1.d. Testing</b>	<b>10</b>
Test Case 1: Positive output with Correct Recommendations	10
Test Case 2: No recommendations	10
Test Case 3: Repeated same input movie names	11
<b>2. Prompting and LLMs: Designs, Comparison, Feedback</b>	<b>12</b>
<b>2.a. Prompt Design</b>	<b>12</b>
<b>2.b. Palm 2</b>	<b>13</b>
A. Palm2 Performance Analysis and Feedback:	13
B. Tool Limitations & Feedback	13
<b>2.c. ChatGPT4</b>	<b>14</b>
Performance Analysis	14
Tool Limitations & Feedback	18
<b>2.d. Claude Comparison and Analysis</b>	<b>19</b>
Performance and Analysis	19
Observations	20
Improvements and Feedback	21
<b>3. Conclusion</b>	<b>22</b>
<b>4. Individual Contributions</b>	<b>23</b>

## 1. Model Building and Testing:

### 1.a. Use Case: Movie Recommendation System Diagram

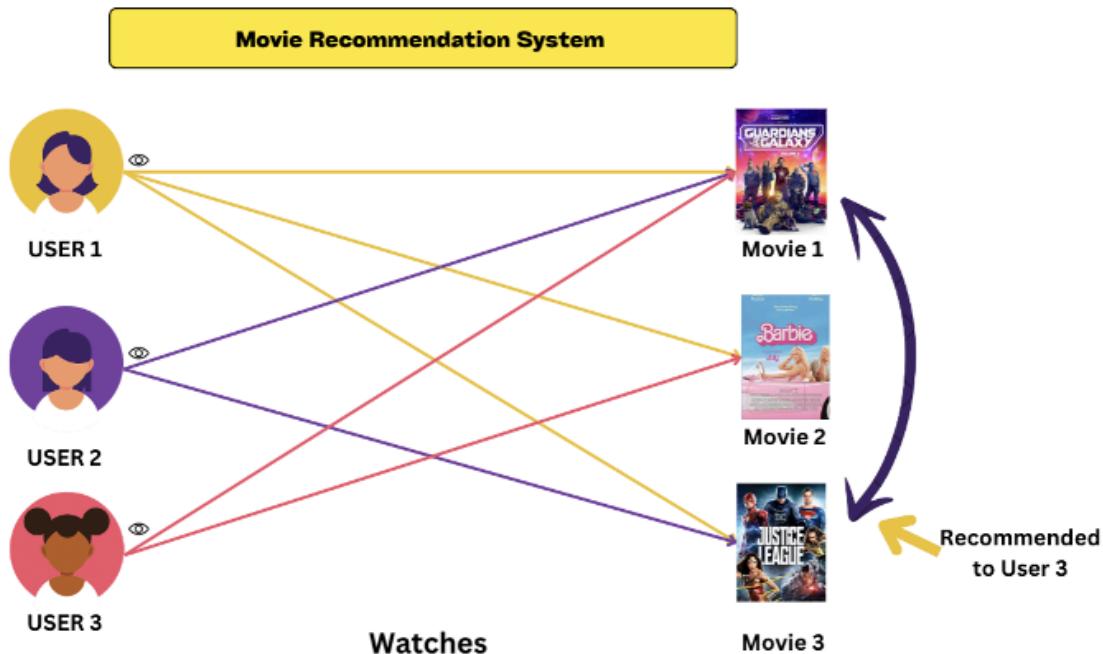


Figure: Use Case Diagram - Movie Recommendation using Palm 2

#### A. User Inputs for Recommendations

- Primary Actor: User
- Secondary Actor: Palm2 (Movie Recommendation System)
- Main Flow:
  - User logs into the Palm2 system.
  - Users provide their movie watching history or specific movie preferences.
  - The system acknowledges the user input.

## B. System Generates Movie Recommendations

- Primary Actor: Palm2 (Movie Recommendation System)
- Secondary Actor: User
- Preconditions: User has provided their watching history or specific movie preferences.
- User Flow:
  - Palm2 processes the provided user data.
  - It evaluates previously watched movies.
  - The system uses algorithms to identify movies similar in genre.
  - Palm2 presents a list of personalized movie recommendations to the user.

## C. Specific Example

- In the above Figure , If the User 3 previously watched "Movie 1," the system might recommend "Movie 3," noting that its genre or content closely aligns with that of "Movie 1."

## D. Postconditions

- User receives a list of movie recommendations.

## 1.b. Python Script

```
!pip install -q google-generativeai  
!pip install ipywidgets
```

```
import pprint  
import google.generativeai as palm  
  
from google.colab import files  
  
uploaded = files.upload()  
  
for fn in uploaded.keys():  
    print('User uploaded file "{name}" with length {length} bytes'.format(  
        name=fn, length=len(uploaded[fn])))
```

```
IMDBDataset.csv(text/csv) - 49683 bytes, last modified: 31/10/2023 - 100% done  
Saving IMDBDataset.csv to IMDBDataset.csv  
User uploaded file "IMDBDataset.csv" with length 49683 bytes
```

```
import os  
import google.generativeai as palm  
import ipywidgets as widgets  
import pandas as pd  
from IPython.display import display, clear_output  
  
PALM_API_KEY = os.getenv("PALM_API_KEY", "yourPalmAPIkey")  
  
  
movie_dataframe = pd.read_csv('IMDBDataset.csv')  
movie_dataset = movie_dataframe['movie_name'].tolist()  
  
class Recommend_movies:  
    def __init__(self) -> None:  
        self.model = palm
```

```

self.model.configure(api_key=PALM_API_KEY)

self.defaults = {
'model': 'models/text-bison-001',
'temperature': 0.7,
'candidate_count': 1,
'top_k': 40,
'top_p': 0.95,
'max_output_tokens': 1024,
'stop_sequences': [],
'safety_settings': [
{"category": "HARM_CATEGORY_DEROGATORY", "threshold": 1},
 {"category": "HARM_CATEGORY_TOXICITY", "threshold": 1},
 {"category": "HARM_CATEGORY_VIOLENCE", "threshold": 2},
 {"category": "HARM_CATEGORY_SEXUAL", "threshold": 2},
 {"category": "HARM_CATEGORY_MEDICAL", "threshold": 2},
 {"category": "HARM_CATEGORY_DANGEROUS", "threshold": 2}
],
}

def generate(self, movie_name, dataset):
results = []

prompt = f"""
input: Th Dark Knight
output: Batman Begins
The Prestige
Se7en
Fight Club
The Shawshank Redemption
input: {movie_name}
output:
"""


```

```

response = self.model.generate_text(**self.defaults, prompt=prompt)

recommendations = [line.strip() for line in response.result.split("output:")[-1].split("\n") if
line.strip()]

# Filter the movies from the dataset based on the recommendations
for movie in dataset:
    if movie in recommendations:
        results.append(movie)
return results

def on_button_click(b):
    movie_name = text.value
    if movie_name:
        clear_output(wait=True)
        display(text, button)
    results = r_m.generate(movie_name, movie_dataset)
    print("Recommended Movies:")
    for movie in results:
        print(movie)
    else:
        print("Sorry I cannot find the similar movie that you want. Try with a different movie name.")

r_m = Recommend_movies()
text = widgets.Text(
    value="",
    placeholder='Your movie goes here 🎬',
    description='Movie:',
    disabled=False
)
button = widgets.Button(description="Recommend")
button.on_click(on_button_click)

display(text, button)

```

## 1.c. ReadMe

### A. Introduction

This Movie Recommendation System takes in the name of a movie and recommends similar movies based on the dataset. The recommendation process is powered by the google.generativeai(presumably a mock-up API for the purpose of this example), which utilizes a generative model to provide movie suggestions.

### B. Requirements

- Python libraries:
  - os
  - google.generativelai (You need an API key for this)
  - ipywidgets
  - pandas
  - IPython
- Data: A CSV file named IMDBDataset.csv containing a list of movie names.

### C. Setup

- Set your PALM API key: You can either set it as an environment variable or directly in the code.

```
pythonCopy code: PALM_API_KEY = os.getenv("PALM_API_KEY", "YOUR_API_KEY_HERE")
```
- Load the movie dataset: The IMDBDataset.csv file is loaded into a pandas dataframe and then converted to a list of movie names.

### D. Usage

- Create an instance of the Recommend\_movies class.
- Use the generate method to get movie recommendations.
- The interface uses IPython widgets for a simple GUI where users can input a movie name and get recommendations.

### E. Workflow

- The Recommend\_movies class configures the generative AI model with the specified parameters.
- Upon providing a movie name and pressing the button, the model takes a sample prompt with movie names and their corresponding recommendations.
- Based on this prompt, the model tries to predict recommendations for the input movie.

- The generated recommendations are then cross-referenced with the dataset to ensure they are valid.
- The results are displayed using IPython.

## F. Note

- The safety\_settings in the Recommend\_movies class aims to prevent any inappropriate or harmful content from being generated.
- If the movie name is not found, the system will prompt the user to try a different name.

## G. Improvements

- A more extensive dataset will provide better recommendations.
- Fine-tuning the model specifically for movie recommendations can yield better results.
- Incorporate user feedback to continually improve the recommendation quality.

## H. Conclusion

This recommendation system provides a quick and easy way to find similar movies. With the potential to integrate more advanced features and datasets, this system can be further enhanced to suit various user needs.

## 1.d. Testing

### Test Case 1: Positive output with Correct Recommendations

1.1. When we queried the model with the movie name "Avengers: Endgame" which is in the IMDB dataset, the model correctly predicted "Avengers: Infinity War" as Recommended movie since both movies belong to the Action and Adventure genre.

The screenshot shows a user interface for movie recommendations. A search bar at the top contains the text "Movie: Avengers: Endgame". Below it is a button labeled "Recommend". Underneath the button, the text "Recommended Movies:" is followed by "Avengers: Infinity War". The entire interface is enclosed in a light gray border.

1.2. In this case, when we queried the model with the movie "Psycho" which belongs to the "Horror, Mystery, Thriller" genre, it suggested "Rear Window" from the "Mystery, Thriller" genre and "Vertigo" from the "Mystery, Romance, Thriller" genre.

The screenshot shows a user interface for movie recommendations. A search bar at the top contains the text "Movie: Psycho". Below it is a button labeled "Recommend". Underneath the button, the text "Recommended Movies:" is followed by "Rear Window" and "Vertigo". The entire interface is enclosed in a light gray border.

In most instances using our sample data, the model performed well in its predictions. However, due to the limited data we trained it on, there were a few unfavorable outcomes as shown below.

### Test Case 2: No recommendations

2.1. When we queried the model with the movie "Taxi Driver" which falls under the "Crime, Drama" genre, it didn't recommend other similar movies like "Scarface" or "The Godfather" even though they also belong to the "Crime, Drama" genre.

The screenshot shows a user interface for movie recommendations. A search bar at the top contains the text "Movie: Taxi Driver". Below it is a button labeled "Recommend". Underneath the button, the text "Recommended Movies:" is followed by a blank line. The entire interface is enclosed in a light gray border.

2.2. When we presented the model with "Dag II" from the "Action, Drama, War" genre, it failed to suggest movies like "Oldeuboi" despite it also being in the "Action, Drama" category.



2.3. When we input "Dil Bechara," categorized as "Comedy, Drama, Romance," the model failed to recommend akin movies like "The Apartment" and "City Lights," despite them sharing the same genre classification.



### Test Case 3: Repeated same input movie names

3.1. When we provided the model with the movie "The Green Mile," which belongs to the "Crime, Drama, Fantasy" genre, it suggested the same movie along with other titles.



## 2. Prompting and LLMs: Designs, Comparison, Feedback

As our project, "Enhanced Movie Recommender - Collective Intelligence" is built on top of a database and specific algorithm, we realized the same result can be achieved using palm 2 API. If we trained the Palm 2's chat box with sequences of inputs and outputs, we can potentially get a very accurate prediction model. With the goal of combining LLM with collective intelligence, we designed the following 'experiment.'

### 2.a. Prompt Design

1. Under the "Write your prompt examples" section, to test how well Palm2 is performing on movie recommendations to the users, our team has decided to generate 20 inputs and outputs samples to train the model of Plam2.

- Inputs: we have used a fixed format. (I like "\_\_\_" that is \_\_\_ genre. Please recommend a movie.)

a. In the first blank, we have inputted a movie name which is in the action genre.

b. We have decided to use the "action" genre for all the input samples for the purpose of variable control and design convenience.

- Outputs: We have searched for a comprehensive list of top 100 action movies on Google. We then have randomly selected a move from the list as an output on Palm2 each time.

I like "Die Hard" that is action genre. Please recommend me a movie. "Raiders of the Lost Ark"

2. Under the "Test your prompt" section, we have decided to test how well Palm2 is performing on recommending a movie based on the user input.

- Inputs: we have used a fixed format. (I like "\_\_\_". Please recommend a movie.)

a. This time, rather than including genre as a limiting condition, we have decided to only provide the movie name, for the purpose of testing whether Palm2 can automatically figure out the genre of the given movie and then recommend a new movie from the same genre.

- b. Since we have encountered an issue when running more than 10 testing cases, we have tested 9 cases in total: 3 in action, 3 in comedy, and 3 in romance. In this way, we can be confident that Palm2 not only does a great job on recommending action movies but also other genres.
- Outputs: For each input, we got a corresponding output, which was a movie name.

Test your prompt ⓘ		
	INPUT input:	OUTPUT output:
1	I like "Tango & Cash". Please recommend me a movie.	"Die Hard"
2	I like "The Last Boy Scout". Please recommend me a movie.	"Die Hard"
3	I like "The Matrix". Please recommend me a movie.	"Inception"
4	I like "Groundhog Day". Please recommend me a movie.	"Edge of Tomorrow"
5	I like "This Is Spinal Tap". Please recommend me a movie.	"Best in Show"
6	I like "Airplane!". Please recommend me a movie.	"The Naked Gun"
7	I like "The Notebook". Please recommend me a movie.	"Titanic"
8	I like "A Walk to Remember". Please recommend me a movie.	"The Notebook"
9	I like "Gone with the Wind". Please recommend me a movie.	"The Wizard of Oz"

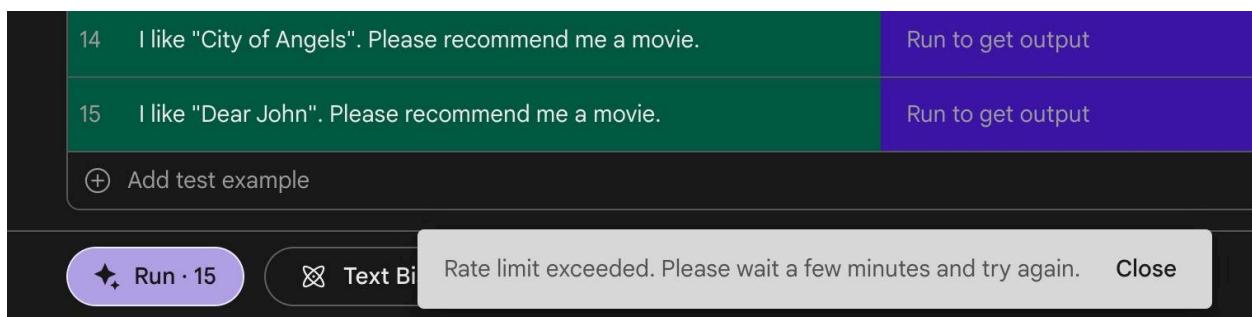
## 2.b. Palm 2

### A. Palm2 Performance Analysis and Feedback

Overall, Palm2 does an excellent job on identifying the correlation between the inputs and outputs from our training set. For each of the 9 testing cases, Plam2 perfectly provides a movie that is in the correct genre. When we input an action movie and ask for recommendations for another one, Palm2 gives us another action movie. When we input a comedy movie, it recommends another comedy movie. When we input a romance movie, it provides us with another romance movie. However, it is worth mentioning that, due to the limitations of the number of testing prompts in Palm2, the testing cases we have used only involve three genres, and we are not 100% confident that Palm2 would also do a perfect job on other genres. This concern can be addressed in the next step.

### B. Tool Limitations & Feedback

Originally we intended to train the model by uploading our database into Palm 2. Since Chat GPT 4.0 now supports our excel file upload size with 26,000 rows. Unfortunately, we realized that Palm 2's excel has a 500 rows limitation. This caused us to readjust our means of approach. Additionally, when we test 15 cases after the training, we encounter errors when we test beyond 10 cases.



Even though our team understands that the LLM can learn quickly in the “**Few-Shot-Learning**” approach, removing those constraints will help us to do a better in depth analysis to have a well rounded test.

## 2.c. ChatGPT4

### Performance Analysis

The model was trained with 20 sample inputs and outputs as shown in the screenshot below, and was then tested on 9 test cases for evaluating its performance, shown in the next screenshot.

**GPT-4**'s recommendation system consistently provided accurate and consistent recommendations. The model reliably makes recommendations for movies based on the genre

provided in the input, reflecting a thorough understanding of different movie genres. Moreover, the model shows the capacity to handle a variety of inputs, from action to romance and comedy, obviating its ability to adapt as needed.

\* GPT-4

**F** input: output:  
"I like ""Indiana Jones and the Temple of Doom"" that is action genre. Please recommend me a movie.  
" """Indiana Jones and the Last Crusade""  
"  
"I like ""Die Hard"" that is action genre. Please recommend me a movie.  
" """Raiders of the Lost Ark""  
"  
"I like ""Terminator 2: Judgment Day"" that is action genre. Please recommend me a movie.  
" """Die Hard""  
"  
"I like ""Predator"" that is action genre. Please recommend me a movie.  
" """The Road Warrior""  
"  
"I like ""RoboCop"" that is action genre. Please recommend me a movie.  
" """Aliens""  
"  
"I like ""Point Break"" that is action genre. Please recommend me a movie.  
" """Aliens""  
"  
"I like ""RoboCop"" that is action genre. Please recommend me a movie.  
" """Die Hard""  
"  
"I like ""The Rock"" that is action genre. Please recommend me a movie.  
" """Predator""  
"  
"I like ""Lethal Weapon"" that is action genre. Please recommend me a movie.  
" """RoboCop""  
"  
"I like ""Speed"" that is action genre. Please recommend me a movie.  
" """Indiana Jones and the Temple of Doom""  
"  
"I like ""Speed"" that is action genre. Please recommend me a movie.  
" """RoboCop""  
"  
"I like ""Speed"" that is action genre. Please recommend me a movie.  
" """Aliens""

Send a message ▶

ChatGPT can make mistakes. Verify important information.

F

"I like ""Tango & Cash"". Please recommend me a movie.

"



"Demolition Man"



F

"I like ""The Last Boy Scout"". Please recommend me a movie.

"



"Kiss Kiss Bang Bang"



F

"I like ""The Matrix"". Please recommend me a movie.

"



"Inception"



F

"I like ""Groundhog Day"". Please recommend me a movie.

"



"Edge of Tomorrow"



F

"I like ""This Is Spinal Tap"". Please recommend me a movie.

"



"Best in Show"

⟳ Regenerate



✉ Send a message



ChatGPT can make mistakes. Verify important information.

Complete list of sample inputs and outputs:

Your Preference	Recommendation Given
Indiana Jones and the Temple of Doom	Indiana Jones and the Last Crusade
Die Hard	Raiders of the Lost Ark
Terminator 2: Judgment Day	Die Hard
Predator	The Road Warrior
RoboCop (1st mention)	Aliens
Point Break	Speed
RoboCop (2nd mention)	Die Hard
The Rock	Predator
Lethal Weapon	RoboCop
Speed (1st mention)	Indiana Jones and the Temple of Doom
Speed (2nd mention)	RoboCop
Speed (3rd mention)	Aliens
First Blood (1st mention)	Total Recall
Total Recall	Commando
First Blood (2nd mention)	Aliens
Die Hard with a Vengeance	Con Air
First Blood (3rd mention)	True Lies
Face/Off	Con Air
Big Trouble in Little China	Con Air
Last Action Hero	First Blood

Complete list of test cases used and the models outputs:

Your Preference	Recommendation Given
Tango & Cash	Demolition Man
The Last Boy Scout	Kiss Kiss Bang Bang
The Matrix	Inception
Groundhog Day	Edge of Tomorrow
This Is Spinal Tap	Best in Show
Airplane!	The Naked Gun: From the Files of Police Squad!
The Notebook	P.S. I Love You
A Walk to Remember	The Fault in Our Stars
Gone with the Wind	Doctor Zhivago

## Tool Limitations & Feedback

- Since GPT-4's training data is not updated in real-time, it is not able to incorporate information on movies released after its most recent training cut-off in early 2022. Therefore, the model will not recommend new releases, current reviews, or trending movies.
- GPT-4's recommendations are only as unbiased as the data it was trained on. The recommendations will be biased if the training data contains any biases, such as an overrepresentation of some movie genres or an underrepresentation of others.
- GPT-4 lacks the ability to integrate with external databases in order to retrieve real-time data or user ratings from popular movie databases such as IMDb.

Overall, the model consistently understands user movie preferences and provides relevant recommendations, effectively leveraging its knowledge base to provide suggestions matching the input.

## 2.d. Claude Comparison and Analysis

### Performance and Analysis

Claude's movie recommendation system consistently provided highly accurate and relevant movie suggestions, further validated by its alignment with the IMDb most popular dataset. Remarkably, without explicit prompting, the system autonomously discerned patterns of collective intelligence within the training inputs.

Action film recommendations from an AI assistant ▾

input: I like "Indiana Jones and the Temple of Doom" that is action genre. Please recommend me a movie.

output: "Indiana Jones and the Last Crusade"

input: I like "Die Hard" that is action genre. Please recommend me a movie.

output: "Raiders of the Lost Ark"

input: I like "Terminator 2: Judgment Day" that is action genre. Please recommend me a movie.

output: "Die Hard"

input: I like "Predator" that is action genre. Please recommend me a movie.

output: "The Road Warrior"

input: I like "RoboCop" that is action genre. Please recommend me a movie.

output: "Aliens"

input: I like "Point Break" that is action genre. Please recommend me a movie.

output: "Aliens"

input: I like "RoboCop" that is action genre. Please recommend me a movie.

output: "Die Hard"

input: I like "The Rock" that is action genre. Please recommend me a movie.

output: "Predator"

input: I like "Lethal Weapon" that is action genre. Please recommend me a movie.

Message Claude... 📎 ➤

input: I like "Speed" that is action genre. Please recommend me a movie.

## Observations

This level of adaptability and intelligent data utilization showcased the system's capability to evolve and fine-tune its recommendations based on unprompted patterns within the training dataset.

The system quickly grasped the expected patterns of popularity and relevance over genre and era, demonstrating its agility in understanding and adapting to my requirements.

A screenshot of a conversational AI interface. At the top, a dark header bar displays the ID: 1-4d83-b456-b0eb933089a4. Below it, a light-colored card has the title "Action film recommendations from an AI assistant ▾". The card contains a summary message: "Here is a summary table of the movie recommendations I provided based on the liked movies:". A table follows, titled "Liked Movie" and "Recommended Movies". The table lists eight liked movies with their corresponding recommended titles. At the bottom of the card, a message reads: "This summarizes the different movie suggestions across genres like action, sci-fi, comedy, romance and drama that I recommended based on the liked movies provided. Let me know if you need me to clarify or expand on any of the recommendations!" Below the card, there are buttons for "Copy" and "Retry". At the very bottom, a text input field says "Message Claude..." and features a send button with a circular arrow icon.

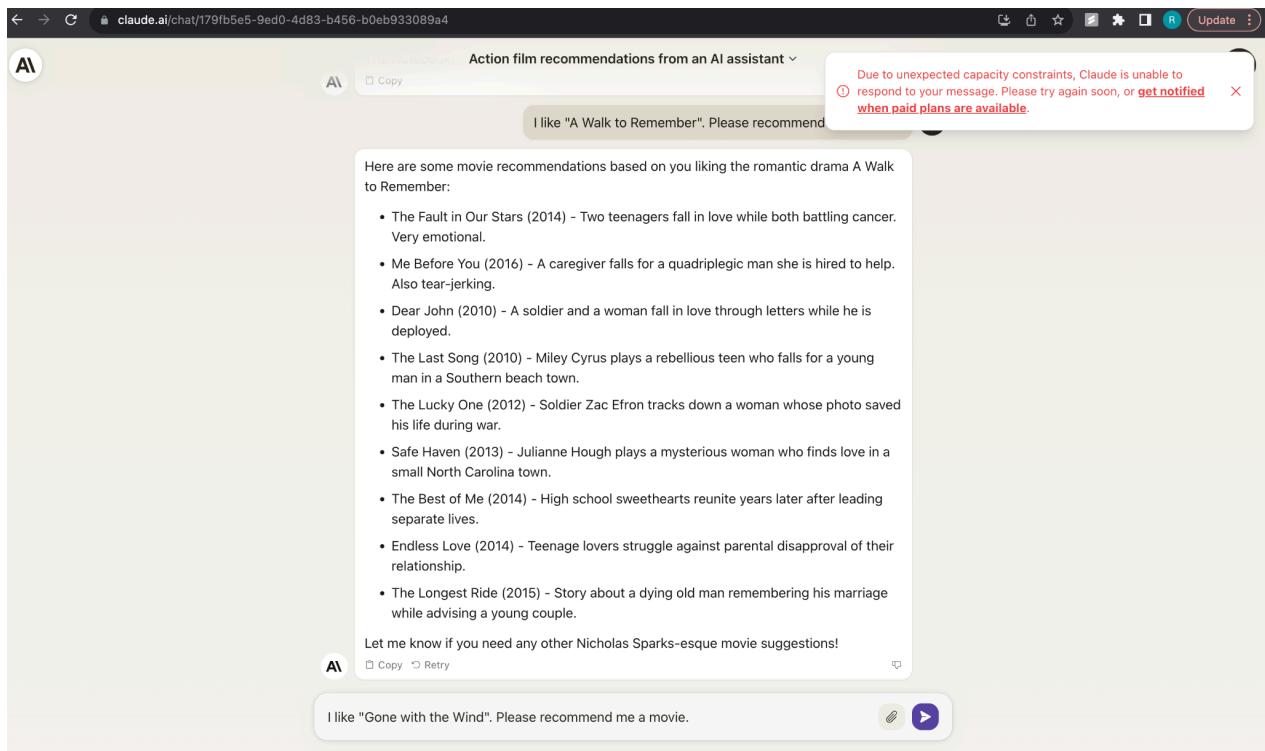
Liked Movie	Recommended Movies
Indiana Jones and the Temple of Doom	Indiana Jones and the Last Crusade, Raiders of the Lost Ark
Die Hard	Lethal Weapon, 48 Hrs., Beverly Hills Cop, Rush Hour
The Matrix	Inception, The Terminator, Total Recall, Dark City, 12 Monkeys
Groundhog Day	Edge of Tomorrow, Palm Springs, Happy Death Day, 50 First Dates, About Time
Airplane!	The Naked Gun, Top Secret!, Hot Shots!, Spaceballs, Blazing Saddles
The Notebook	Titanic, The Fault in Our Stars, Me Before You, A Walk to Remember, The Vow
A Walk to Remember	The Fault in Our Stars, Me Before You, Dear John, The Last Song, The Lucky One
Gone with the Wind	Doctor Zhivago, Out of Africa, The English Patient, Cold Mountain, Giant

## Improvements and Feedback

However, there were notable issues with the system. Periodic pauses, attributed to backend problems, impeded the seamless flow of recommendations, causing user frustration.

These pauses could disrupt the user experience and diminish the system's overall reliability.

Furthermore, Claude occasionally offered multiple movie recommendations when users requested a single suggestion. This inconsistency in response created a deviation from the implicit expectations.



### 3. Conclusion

The PALM 2 model shows proficiency in correlating movies with direct sequels and shared directorial styles, as evidenced by its correct recommendations for movies like "Avengers: Endgame" and "Psycho." However, it struggles with offering recommendations for less obvious connections or across a wider array of genres and lacks proper deduplication to prevent suggesting the input movie as a recommendation. The model requires enhancements in its dataset and recommendation algorithms to address these gaps.

If we could use **one word** to describe **each** of the **LLM** models within our narrow use case of movie recommendations; Claude is like a tool that we can use but it functions a lot like a machine. ChatGPT is like a personalized assistant **intern**. It can help us do most tasks, albeit with mistakes. Only if we explicitly teach it well enough will it do just right. For Palm 2, it is a diligent student who is good at test taking but **lacks practical nuance**. Ultimately, regardless if it's a tool, assistant or student, it still is one of the most innovative products in recent years of human history.

#### 4. Individual Contributions

Names	Project Contributions
Shashank Kallahalli Suresh	Movie recommendation Python script, Read me file, test cases for project model built
Hai Zheng	Project Structure, Project Design, Feedback, Paper Introduction Video
Jiayi Liang	Palm2 sample prompts inputs and outputs, testing prompts and performance analysis and feedback.
Raj Shah	Movie Recommendation Documentation, Claude model training, testing and analysis of results
Fatima Fazal	ChatGPT4 model training, model testing, analysis of results

#### Note:

We will continue our original project, “Movie Recommender” for the final project, but we will also incorporate some of our findings through this google hackathon experience.