

# CineSphere

AI Powered Entertainment Guide



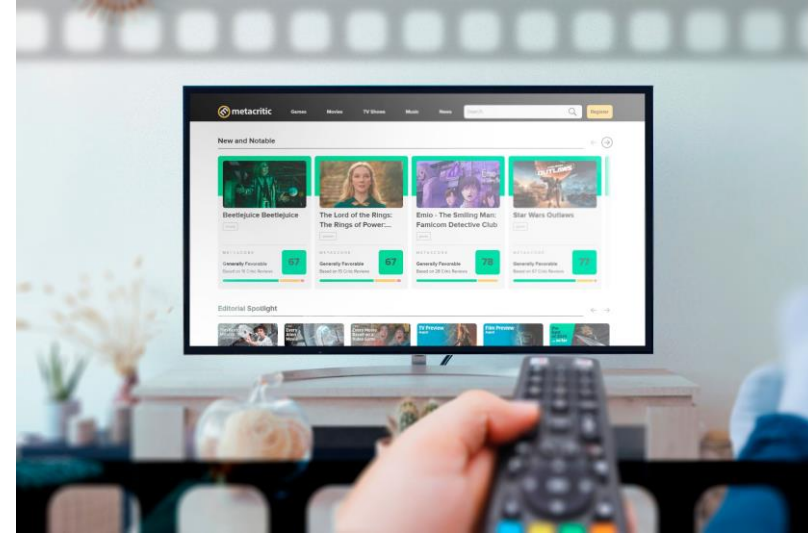
Abhishek Shankar

# Contents

- Problem Statement
- Solution
- About the data
- Building the graph
- Functionalities
- Demo + Use Cases
- Future Scopes

# Problem Statement

- Wasting time trying to find the right movie to watch on a Friday night?
- How can we simplify the discovery of world cinema while addressing diverse user queries about movies, actors, and genres, all within a personalized, interactive, and seamless experience?

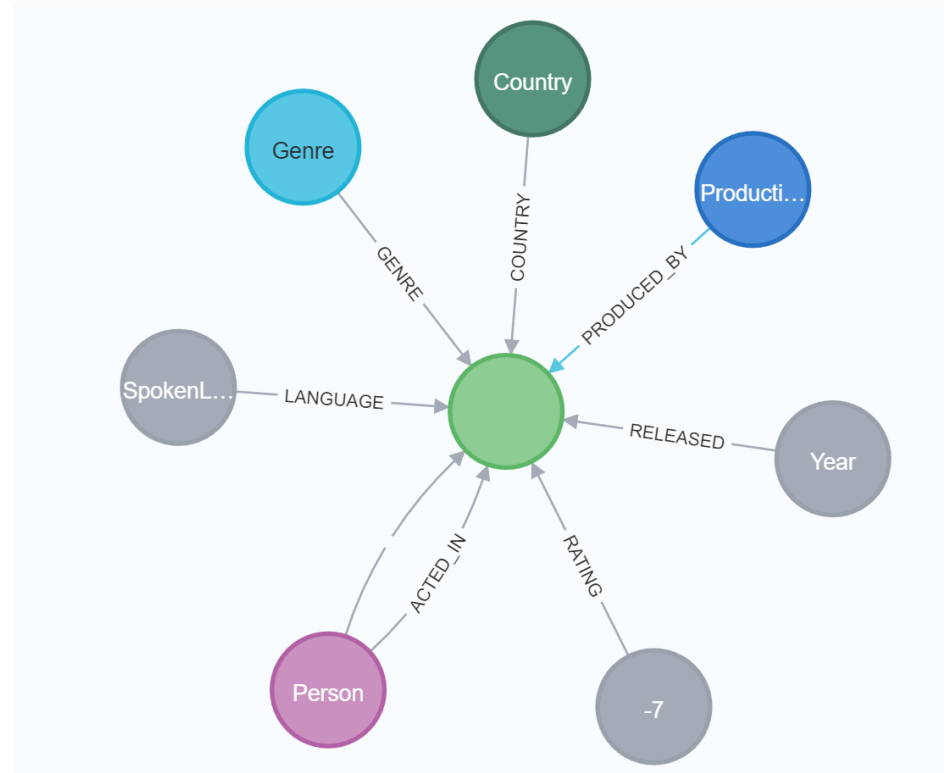


# Solution

- Interactive User Interface and experience
- Personalised movie recommendations based on user preferences
- Q&A bot to cater to user queries
- Instant movie suggestions tailored to user sentiments
- Wide variety of world cinema
- Using State of the art tools like
  - Neo4j
  - OpenAI
  - Streamlit

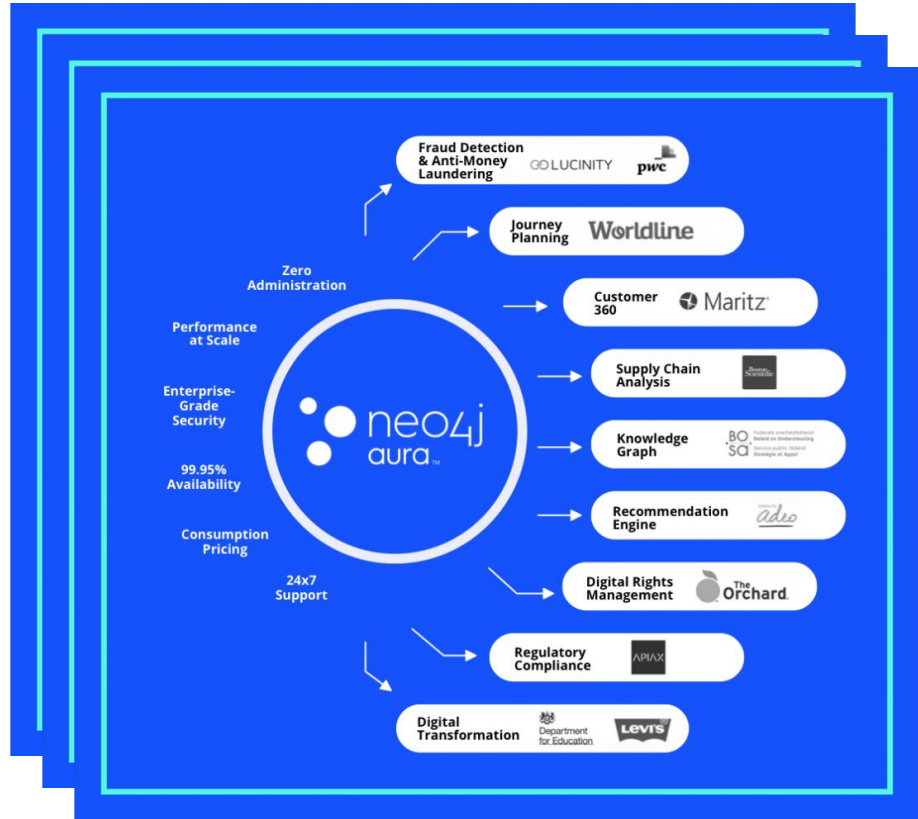
# About the Data

- The dataset was obtained from Kaggle and consists of over 45000 movies.
- Data points include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDb vote counts and vote averages



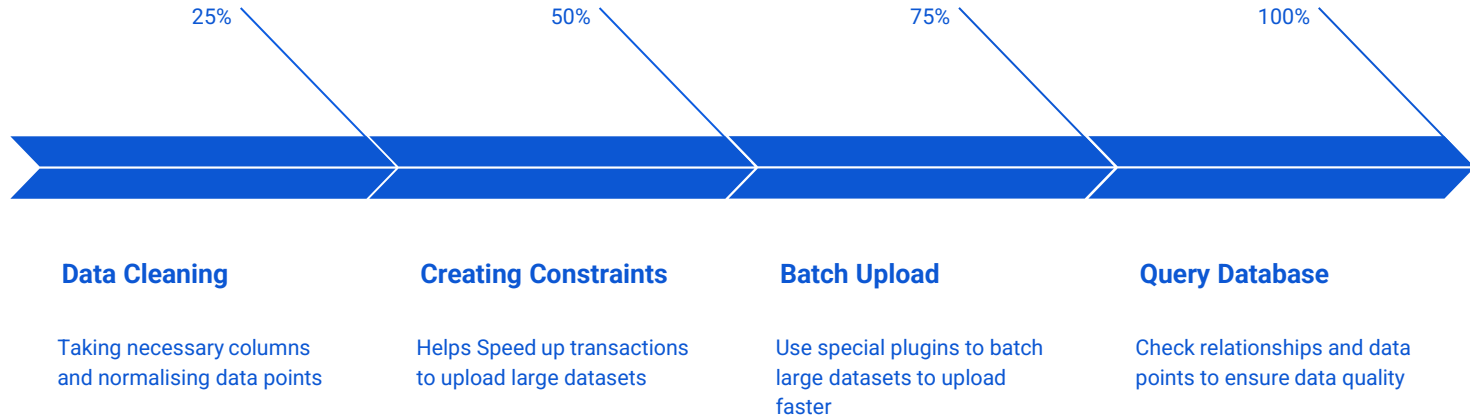
# Why Use a Graph Database

- Handling Complex Relationships
- Efficient Querying
- Recommendation Systems
- Scalability
- Real-time Recommendations



SQL Query	Macros Query on MS Excel	Cypher Query
<pre> SELECT   m.*,   r.*,   related.* FROM   movies_metadata_clean_2 m LEFT JOIN   clean_ratings r ON m.movie_id = r.movie_id LEFT JOIN   normalised_production_companies pc ON   m.movie_id = pc.movie_id LEFT JOIN   normalised_genres g ON m.movie_id = g.movie_id LEFT JOIN   normalised_spoken_languages sl ON m.movie_id = sl.movie_id LEFT JOIN   normalised_production_countries pcn ON   m.movie_id = pcn.movie_id LEFT JOIN   keywords_clean k ON m.movie_id = k.movie_id LEFT JOIN   normalised_cast2 c ON m.movie_id = c.movie_id LEFT JOIN   normalised_crew cr ON m.movie_id = cr.movie_id WHERE   m.name = 'Inception'; </pre>	<pre> Sub SearchMovieInExcel()   Dim ws As Worksheet   Dim movieName As String   Dim movieRow As Long    ' Set the worksheet containing movie data   Set ws =   ThisWorkbook.Sheets("movies_metadata_clean_2")    ' Prompt the user to enter the movie name   movieName = InputBox("Enter the movie name:")    ' Find the row number of the movie with the given name   movieRow =   Application.WorksheetFunction.Match(movieName,   ws.Range("A:A"), 0)    ' If the movie is found, display its details   If Not IsError(movieRow) Then     MsgBox "Movie found at row " &amp; movieRow     ' You can access movie details using     ws.Cells(movieRow, columnNumber)     ' For example:     ' Dim movieID As String     ' movieID = ws.Cells(movieRow, 2).Value ' Assuming ID     is in column B     ' MsgBox "Movie ID: " &amp; movieID   Else     MsgBox "Movie not found."   End If End Sub </pre>	<pre> MATCH (m:Movie {name:'Inception'})-[r]- (related) RETURN m, r, related; </pre>

# Building the graph





# What does the graph contain?

Types of Nodes

nodeType	nodeCount
["Genre"]	33
["SpokenLanguage"]	134
["Year"]	135
["Country"]	162
["Movie"]	1000
["ProductionCompany"]	23693
["User"]	175400
["Person"]	218814

Types of Relationships

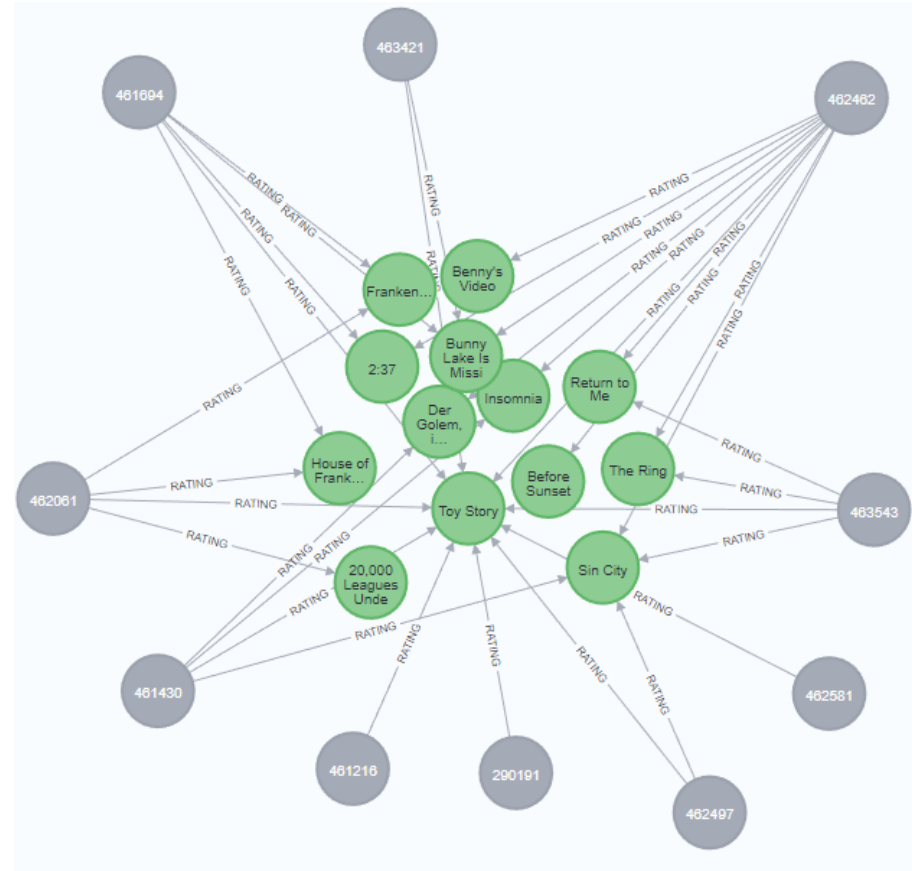
relationshipType	count
"RATING"	273916
"CREWED_IN"	28464
"ACTED_IN"	24780
"GENRE"	4864
"PRODUCED_BY"	4142
"LANGUAGE"	2682
"COUNTRY"	2552
"RELEASED"	1996

Types of Indices

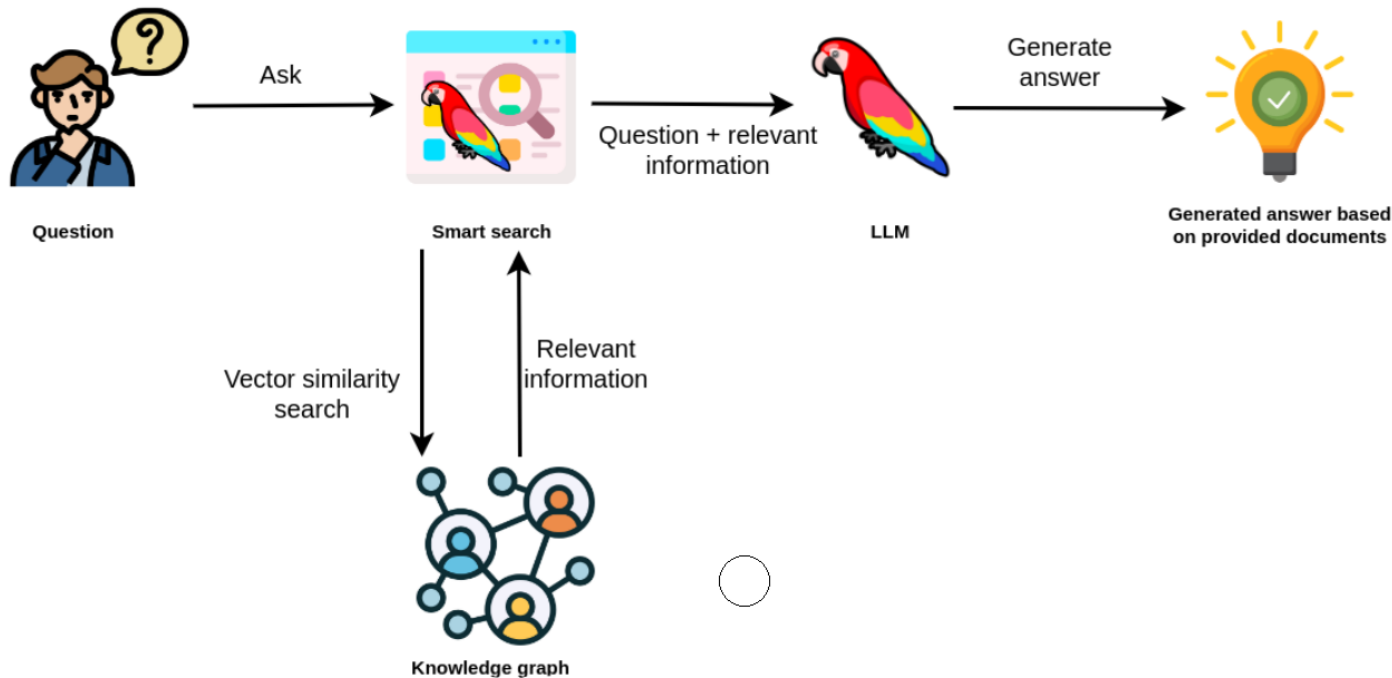
labelsOrTypes	name	state
["Movie"]	"overview_embeddings2"	"ONLINE"
["Country"]	"unique_countries_id"	"ONLINE"
["Genre"]	"unique_genre_id"	"ONLINE"
["SpokenLanguage"]	"unique_lang_id"	"ONLINE"
["Movie"]	"unique_movie_id"	"ONLINE"
["Person"]	"unique_person_id"	"ONLINE"
["ProductionCompany"]	"unique_prod_id"	"ONLINE"
["User"]	"unique_user_id"	"ONLINE"

# Functionalities - Personalised Recommendations

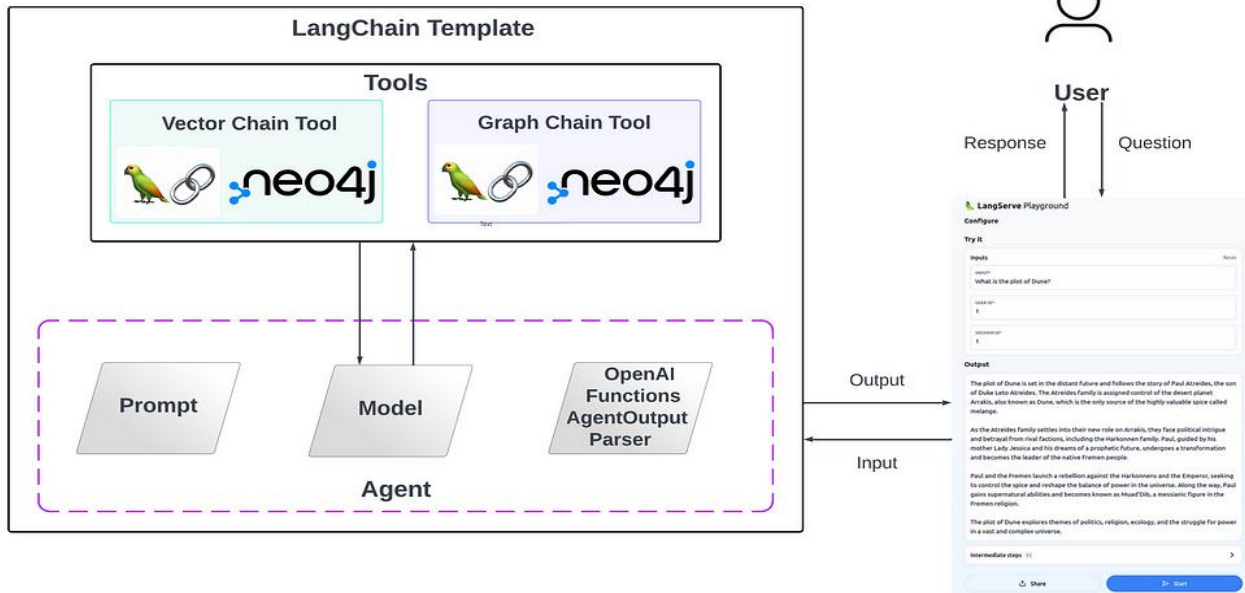
- Users are given a set of movies based on the most popular movies in the database.
- Node similarity is determined by measuring the distance between nodes, which is then filtered based on the collective characteristics of the nodes.
- We aim to identify in relation to the source node that serves as the reference point.



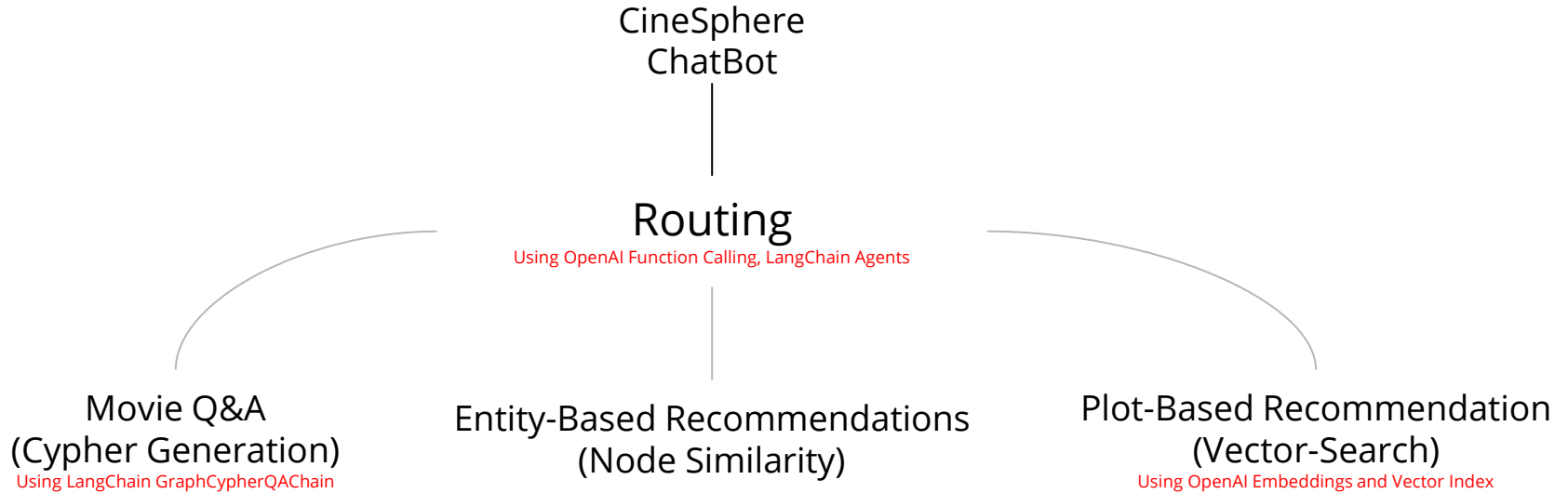
# Functionalities - Vector Search



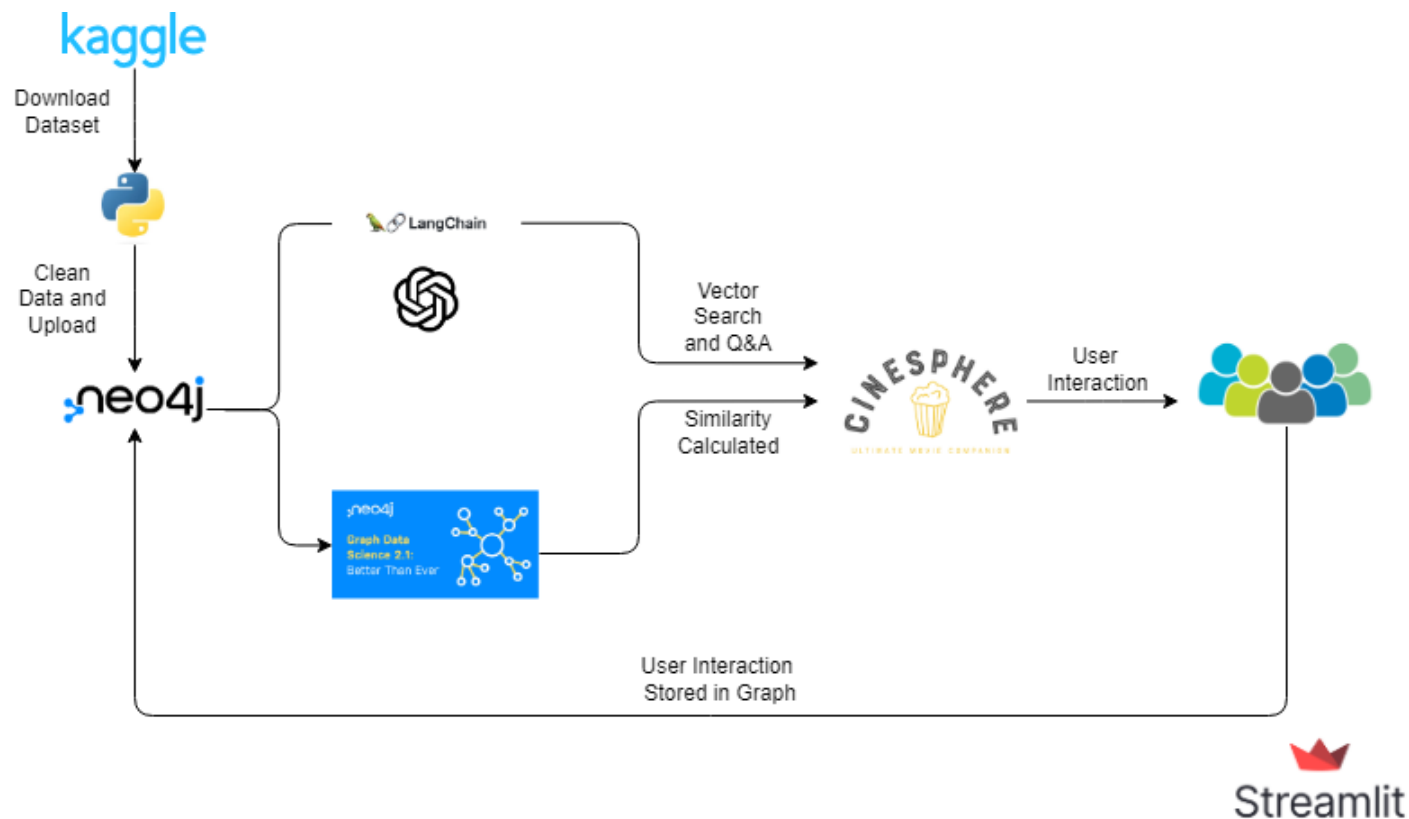
# Functionality - Q&A Bot



# Chat Interface - Multi-Agent Tool



# Product Workflow



# Future Scopes

- Optimised recommendations based on
  - More parameters
  - User Interactions
- Refined user interface to facilitate more features
- Add conversation history in Chat, and personalisation
- Reduced latency and increased concurrency
- Developing a hybrid vector search