Interim Project Report

Ananya Jain (2019408)
Gitansh Raj Satija (2019241)
Rohan Jain (2019095)

CineSight

October 31st, 2022

Problem Statement

1. **Development:** Creating an integrated platform with a materialized approach for searching and comparing consolidated data of the movies and use that to recommend movies based on user interest .
2. **Research and Analysis:** Creating a collaboration graph between directors, actors, actresses for analyzing the growth of industry members and collaboration in the industry.

Motivation(why?)

Cinematography is one of the bliss of human development. People all across the world spend their free time by watching movies across different platforms. Sometimes it becomes difficult to find the movies to our liking. Thus, we put forth the concept of a platform that contains a consolidated view for all movie related details across different movie sources so that people don't need to refer to multiple different sources while wanting to know more about movies, its cast and directors, and trace their journeys. Cinema is a collaborative field just like any other. Thus, it might be possible to gain many insights regarding the collaborative nature of the industry based on different parameters, using social network analysis methods that are also used to study collaboration between different researchers, lawyers and more...

End User

1. Common People

   Our primary end users will be Common People (movie buffs or everyday users) who will be using the platform for gaining insights regarding cinema, finding recommendations or comparing movies.

2. Industry Analysts and Researchers

   Our other target is Industry Analysts and Researchers who can gain insights from our heterogenous data graph and also reuse them for extended purposes.

Data Sources and Collection [link]

1. IMDb Dataset

   IMDB publishes all its data daily in different files:

   i. title.akas.tsv.gz
   ii. title.basics.tsv.gz
   iii. title.crew.tsv.gz
   iv. title.principals.tsv.gz
   v. title.ratings.tsv.gz
   vi. name.basics.tsv.gz

   These files contain data related to movies that is useful to us in our Global Schema. We fetch these files and extract them. Using Pandas as an ETL tool, we can extract information using these tsv format files. We have automated the whole process so we can do this every day for data updation. The process for data updation is listed later. Schema can be found here.

2. TMDb Dataset

   For the initial data dump, we found a kaggle dataset fetched via TMDB API that contains details of xx movies. TMDB publishes a few daily files which are:

   i. movie_ids.json.gz
   ii. person_ids.json.gz
   iii. collection_ids.json.gz
   iv. tv_networks_ids.json.gz
   v. keyword_ids.json.gz
   vi. production_companies_ids.json.gz

   These files contain a list of valid IDs on a given date. We use this file to find out the missing IDs and fetch the data using Python and TMDB API for those rows. Schema can be found here.

3. MovieLens Dataset

   This dataset is static and already available for download on the grouplens site. We just downloaded the file, extracted it and our data source was ready. Schema can be found here.

4. Celebrity API Dataset

   The Celebrity API is provided by API Ninjas. For extracting data from this source, we wrote a Python code which hits the API exposed by API Ninjas along with our API Key. For each query, we give the name of a celebrity and it returns the most similar names that match our query. After fetching the data, we will apply entity matching between the required celebrity and the list of celebrities returned by the API. Schema can be seen using the given sample:-

```
[
  {
    "name": "michael jordan",
    "net_worth": 2200000000,
    "gender": "male",
    "nationality": "us",
    "occupation": [
      "basketball_player",
      "athlete",
      "spokesperson",
      "entrepreneur",
      "actor"
    ],
    "height": 1.98,
    "birthday": "1963-02-17",
  }
]
```


Methodology

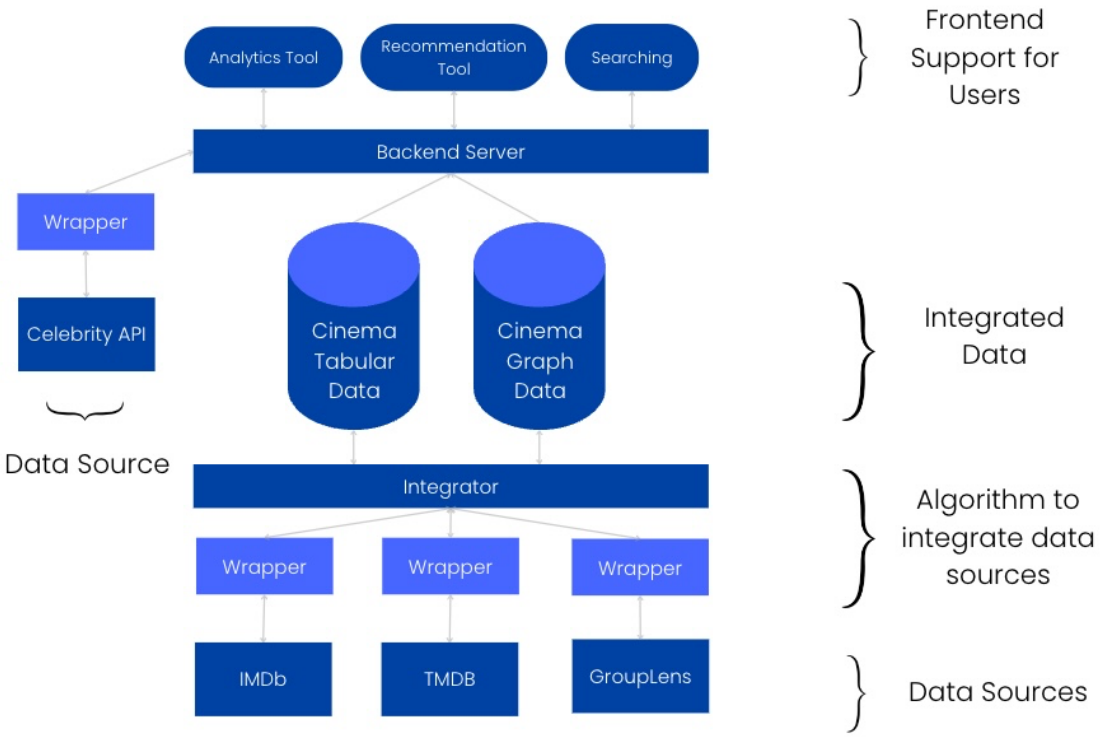
We plan to integrate data sources in a materialized approach to form our own database. We have chosen materialized approach over mediator approach due to the following reasons:

- Graph: We plan to make a heterogeneous graph which will require the presence of all the data before we can start running our queries on it
- There is no need for data to be updated in real time in our use case. Data that is 2-3 days old also works.
- Movielens database is static

Collaborative Graph:

Our graph will be a heterogeneous network, composed of two different kinds of nodes: one where people are the vertices and the number of movies between people are the weights for the edges. The second category of nodes represent movies, having unweighted edges with people that worked with . Taking an example, The node representing Priyanka Chopra is connected to

the Barfi(Movie) node and also connected to Ranbir Kapoor's node with a weight 3 to depict 3 movies in which they worked together. For this, we will make use of an hybrid approach where we are using the materialized information to make our graph and the Celebrity API will be used to fetch updated information like net_worth of the client when a user focuses on any node of the graph.



Global Schema

Movie

MovieID: Primary Key Title : String Original Title: String Release_date : Datetime Runtime : Float Isadult : Boolean ImdbLink : String

Overview : String
Popularity : Float
Picture Path : String
CollectionBelongsTo : String
Revenue : Float
Budget : Float
Tagline : String
homepageLink: String

Language

Language ID
MovieID : Foreign Key

Keywords

KeywordsID
KeywordsID : Foreign Key

Genre

GenreID
MovieID : Foreign Key

Person

PersonID : Primary key
Name : String
Birthyear : Integer
Deathyear : Integer
PrimaryProfession : SET
KnownForTitles : SET
AliasName : String
Gender : String
Biography : String
Profile_path : String
Popularity : Float
Place_of_birth : String
Homepage : String
HomeCountry : String
IMDBLInk : String

PersonID: Foreign Key
MovieID : Foreign Key
Ranking
charactersPotrayed
role

MovieReview and Ratings

imdb_avgVotes : Float
imdb_numVotes : Float
tmdb_voteAverage : Float
Reviews : String
rtAllCriticsRating : Float
rtTopCriticsRating : Float
rtAudienceRating : Float
movielensRating: Rating
tmdb_voteCount : Float

ProductionHouses

Homepage : String
Description : String
Headquarter : String
Companyid : String
Logo_path : String
Name_ : String
Origin_country : String
Parent_company : String

Companyid : Foreign Key
MovieId : Foreign Key
Movie_countries ; SET
Movie_locations : SET

Global and Local Schema Mapping

1. Incase of presence in all 2 or more databases the following attributes can be fetched from any of the databases depending on speed of data fetch / Validity of data in databases:

	IMDB	**TMDB**	**MOVIELENS**
Title	Title.akas.tsv.gz : title	TITLE	spanishTitle
Original Title	primaryTitle	ORIGINAL_TITLE	title
Language	language	ORIGINAL_LANGUAGE	
Release_date	startYear	RELEASE_DATE	year
Runtime	runtimeMinutes	RUNTIME	
Picture Path		POSTER_PATH	imdbPictureURL rtPictureURL
Isadult	isAdult (boolean) - 0: non-adult title; 1: adult title	ADULT	
ImdbLink	titleId		
Overview		OVERVIEW	
Popularity		POPULARITY	
CollectionBelongsTo		COLLECTION	
Revenue		REVENUE	
Budget		BUDGET	
Tagline		TAGLINE	
homepageLink		HOMEPAGE	

1. Incase of these attributes union of the database attributes would be considered

	IMDB	**TMDB**	**MOVIELENS**
Genre	genres		movie_genres.genre
Keywords			tags.value

Incase of presence in all 2 or more databases the following attributes can be fetched from any of the databases depending on speed of data fetch / Validity of data in databases:

	IMDB	**TMDB**	**MOVIELENS**	**CelebAPI**
Name	primaryName	NAME_	movie_directors.directorName,movie_actors.actor_name	name
Birthyear	birthYear	BIRTHDAY		birthday
Deathyear	deathYear	DEATHDAY		
PrimaryProfession	primaryProfession			occupation
KnownForTitles	knownForTitles			
AliasName		ALSO_KNOWN_AS		
Gender		GENDER		gender
Biography		BIOGRAPHY		
Profile_path		PROFILE_PATH		
Popularity		POPULARITY		
Place_of_birth		PLACE_OF_BIRTH		
Homepage		HOMEPAGE		
HomeCountry		COUNTRY		nationality
IMDBLInk		ADULT IMDB_ID		

Net worth and Height are additional attributes that are present in the schema for Graphs.

A wrapper works for mapping the attributes . Here first from the respective databases these people are mapped to their movies. These movies are mapped to ids in the global schema and further used to accommodate the global schema columns.

	IMDB	TMDB	MOVIELENS
charactersPotrayed	characters		
role	directors writers category job	DEPARTMENT	movie_directors.directorName movie_actors.actorName
ranking	ranking		

	IMDB	TMDB	MOVIELENS
imdb_avgVotes	averageRating		
imdb_numVotes	numVotes		
movielensRating			Rating
Reviews		MOVIE_REVIEW	
tmdb_voteAverage		VOTE_AVERAGE	
tmdb_voteCount		VOTE_COUNT	
rtAllCriticsRating			rtAllCriticsRating
rtTopCriticsRating			rtTopCriticsRating
rtAudienceRating			rtAudienceRating

	IMDB	TMDB	MOVIELENS
Homepage		HOMEPAGE	
Description		DESCRIPTION_	
Headquarter		HEADQUARTERS	
Logo_path		LOGO_PATH	

Name		NAME_	
Origin_country		ORIGIN_COUNTRY	
Parent_company		PARENT_COMPANY	
Movie_countries			movie_countries
Movie_locations			movie_locations

Data updation

IMDb Dataset

For updating IMDB data, we will have to pull their daily file export everyday at a given time. After downloading and extracting the files, we use Pandas to see the records that have been updated from the previous day. We use these records to then update our database accordingly.

TMDb Dataset

For incremental updates, TMDB API releases a changelist of movieIds for the specified time period. So, we plan to use that API to get a list of ids that have changed in the past 48 hours and then re fetch the data for those ids and update our database accordingly.

MovieLens Dataset

MovieLens Dataset is static and will not update in future.

Celebrity API Dataset

We will be using instantaneous data from the celebrity API so we are fetching the updated results only.

Query Types and Decomposition

General movie overview: How a movie is rated across different platforms and reviews

ImDB	tmDB	movieLens
• averageRating • numVotes	• VOTE_AVERAGE • VOTE_COUNT • MOVIE_REVIEW	• Rating • rtAllCriticsRating • rtTopCriticsRating • rtAudienceRating

Starcast in the movie: Actors in a movie,name of character played, and actor ranking and actor information

ImDB	tmDB	movieLens
• characters (string) - the name of the character played if applicable, else '\N' • primaryName (string)– name by which the person is most often credited • birthYear – in YYYY format • deathYear – in YYYY format if applicable, else '\N' • primaryProfession (array of strings)– the top-3 professions of the person • knownForTitles (array of tconsts) – titles the person is known for	• BIRTHDAY • ALSO_KNOWN_AS • DEATHDAY • GENDER • BIOGRAPHY • PROFILE_PATH • POPULARITY • PLACE_OF_BIRTH • ADULT IMDB_ID • HOMEPAGE • COUNTRY	• Ranking • actorName • birthday • gender

Extravagance of the movie: Budget, locations shot at , production house,revenue

ImDB	tmDB	movieLens

	• COMPANY
	  ○ HOMEPAGE
	  ○ DESCRIPTION
	  ○ HEADQUARTER
	  ○ COMPANYID
	  ○ LOGO_PATH
		• Movie_countries
		• movie_locations

	○ NAME_ ORIGIN_COUNTRY ○ PARENT_COMPANY • MOVIE ○ REVENUE ○ BUDGET	

Complete movie data

ImDB	tmDB	movieLens
• isAdult • Crew • titleId (link)	• MOVIE_KEYWORD • OVERVIEW • POPULARITY • MOVIE_COMPANY • KEYWORD • PICTURE PATH • COLLECTION	• rtPictureURL • Value (tags)

Movie Recommendation

ImDB	tmDB	movieLens
• genres • keywords	• MOVIE_GENRE • MOVIE_KEYWORD • MOVIE_RECOMMEND ○ RECOMENDED_TITLE • MOVIE_SIMILAR ○ SIMILAR_ORIGINAL_TITLE	• User_taggedmovies • Tags • genre

Actors/Actresses belonging to the top N cast members and have been part of more than M movies which had a positive revenue. *(Such type of queries can be used for aggregation and different analysis as well.)*

ImDB	tmDB	movieLens
• primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at	• MOVIE ○ ORIGINAL_TITLE ○ REVENUE ○ BUDGET • PERSON_NAME	• movieId • ActorName

the point of release • primaryName (string)– name by which the person is most often credited • primaryProfession (array of strings)– the top-3 professions of the person	• HOMEPAGE • COUNTRY • GENDER	

Construction of Graph

ImDB	tmDB	movieLens	Celeb

• primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
• primaryName (string)– name by which the person is most often credited
• primaryProfession (array of strings)– the top-3 professions of the person
	• MOVIE
	  ○ ORIGINAL_TITLE
	  ○ REVENUE
	  ○ BUDGET
	• PERSON_NAME
	• HOMEPAGE
	• COUNTRY
	• GENDER
		• ActorName
		• movieId
			• Net worth
			• ActorName
			• Birthday
			• Height

Graphical Analysis

```
Number of Actor Nodes- 716
Number of Producer Nodes- 622
Number of Director Nodes- 246
Number of Casting Director Nodes- 241
Total Number of Nodes- 1825
```

Heterogenous graph with nodes belonging to different crew members.

Density of Graph 0.00785267964431627

Crew members having top 5 degree values in their respective categories

```
Actors
[('Angelina Jolie', 86),
 ('Tom Cruise', 79),
 ('Leonardo DiCaprio', 77),
 ('Brad Pitt', 75),
 ('Arnold Schwarzenegger', 74)]
```

```
Directors
[('Ridley Scott', 48),
 ('Michael Bay', 48),
 ('Ron Howard', 41),
 ('Peter Jackson', 37),
 ('Don Hahn', 31)]
```

```
Prodcuers
[('Neal H. Moritz', 70),
 ('Hugh Jackman', 69),
 ('Tom Hanks', 65),
 ('Jerry Bruckheimer', 63),
 ('Larry J. Franco', 63)]
```

```
Casting Directors
[('Lucinda Syson', 113),
 ('Sarah Finn', 112),
 ('John Papsidera', 103),
 ('Ronna Kress', 93),
 ('Mary Hidalgo', 93)]
```

Crew members having top 5 values of closeness centrality in their respective categories

```
Actors
[('Angelina Jolie', 0.39221533554025845),
 ('Brad Pitt', 0.3842750473235369),
 ('Tom Cruise', 0.3796299643339117),
 ('Will Smith', 0.37607584100137126),
 ('Johnny Depp', 0.3737970867170089)]
```

```
Directors
[('Brad Peyton', 0.29745416526938145),
 ('Mike Newell', 0.2994653844867022),
 ('Jonathan Mostow', 0.32022920610294736),
 ('Peter Hyams', 0.26367216268039967),
 ('Will Finn', 0.2581937724543047)]
```

```
Casting Directors
[('Lucinda Syson', 0.40386166418501246),
 ('Sarah Finn', 0.39248269432385785),
 ('Ronna Kress', 0.38651070434533413),
 ('John Papsidera', 0.3841041444783852),
 ('Janet Hirshenson', 0.3834220505481239)]
```

```
Prodcuers
[('Brian Grazer', 0.3778858756769412),
 ('Neal H. Moritz', 0.3742830634278003),
 ('Jerry Bruckheimer', 0.3735545713060766),
 ('Tom Hanks', 0.3690846875468586),
 ('Hugh Jackman', 0.36595685121171573)]
```

Crew members having top 5 values of betweenness centrality in their respective categories

```
Actors
[('Arnold Schwarzenegger', 0.028147915880146313),
 ('Angelina Jolie', 0.026644584336829428),
 ('Tom Cruise', 0.022563294670007693),
 ('Brad Pitt', 0.021984697846443822),
 ('Leonardo DiCaprio', 0.01987926007647464)]
```

```
Prodcuers
[('Tom Hanks', 0.03110445522495553),
 ('Neal H. Moritz', 0.01725756220890225),
 ('Brian Grazer', 0.017115182303862258),
 ('Hugh Jackman', 0.017077584236605502),
 ('Douglas Wick', 0.01679204471971285)]
```

```
Directors
[('Ridley Scott', 0.011246156997316846),
 ('Don Hahn', 0.00965904006635408),
 ('Michael Bay', 0.008490881118711116),
 ('Brad Bird', 0.0065126696322548566),
 ('Paul Verhoeven', 0.005227935012895766)]
```

```
Casting Directors
[('Mary Hidalgo', 0.05008523600644913),
 ('Lucinda Syson', 0.04087103030419087),
 ('Sarah Finn', 0.0358973222450102),
 ('John Papsidera', 0.03557092459283543),
 ('Avy Kaufman', 0.03149010845361335)]
```

Crew members having top 5 values of clustering coefficient in their respective categories

```
Actors
[('Tom Hiddleston', 0.9642857142857143),
 ('Josh Duhamel', 0.9454545454545454),
 ('Theo James', 0.9090909090909091),
 ('Shailene Woodley', 0.9090909090909091),
 ('Gwyneth Paltrow', 0.8888888888888888)]
```

```
Casting Directors
[('Robin Gurland', 0.9333333333333333),
 ('Marianne Stanicheva', 0.9272727272727272),
 ('Venus Kanani', 0.9090909090909091),
 ('Jennifer Rudnicke', 0.8241758241758241),
 ('Avi Lerner', 0.7948717948717948)]
```

```
Prodcuers
[('Nina Jacobson', 0.9523809523809523),
 ('Bill Miller', 0.9523809523809523),
 ('Susan Downey', 0.9454545454545454),
 ('Dan Lin', 0.9454545454545454),
 ('Jerry Weintraub', 0.9333333333333333)]
```

```
Directors
[('Guy Ritchie', 0.9454545454545454),
 ('Steven Soderbergh', 0.9333333333333333),
 ('Joe Russo', 0.9166666666666666),
 ('Anthony Russo', 0.9166666666666666),
 ('Joss Whedon', 0.8928571428571429)]
```

Their is an extensive usability of this graphical model as it can be used to plot subgraphs of different industries and see if there is any bias amongst the crew members while choosing the actors in the movies that they are involved with or not. This can also be used to see who is promoting more cross industry collaborations.