Project Proposal

Recasting Movies Using IMDB Data

Gaurav Mishra , Piyush Khemka , Pulkit Sharma

110828660 , 110828688 , 110900867

{gamishra,pkhemka,pusharma}@cs.stonybrook.edu

1 Problem Statement

Given an input movie, can we suggest a suitable replacement for the lead actor male and lead actor female, if the movie were to be remade in the year specified by the user.

Example questions answered by the model:

1. Given the movie - 'Inception', who would be the best actor to play the lead role, if the movie was to be made in 1950s?
2. Given the movie - 'Some Like it Hot', who would be the best actor to play Marilyn Monroe's role, if the movie was to be made in 2016?
3. Who would play Arnold Schwarzenegger's role in Terminator if the movie was to be made in 1940s or if it was made today?

2 Data Collection

We plan to scrape IMDB using Beautiful Soup (BS4) and Scrapy Library in Python. We have extracted the list of all actors (Male and Female) up to date from Wandora[1]. This list will be fed to our crawler which will scrape off data from their respective IMDB pages.

2.1 Actor Data

From the IMDB page of actors, we will scrape their height, date of birth, list of movies starred in and salary offered for each movie (adjusted for inflation). Using the movies listed in the page we will extract the urls of all movies, this actor has starred in.

Following is a sample code which takes in actor's imdb page as input and extracts the urls of all the movies he has played part in.

```
page = urllib2.urlopen(
      "http://www.imdb.com/name/nm0000532/?ref_=tt_ov_st_sm")

soup = BeautifulSoup(page)
divs = soup.findAll("div", { "class" : "filmo-row odd" })
```

[1] http://www.wandora.org/wandora/wiki/index.php?title=IMDB_extractor

```
for div in divs:
    print div.b.a['href']
```

Sample Output

```
 /title/tt2948160/?ref_=nm_flmg_act_11
 /title/tt4292334/?ref_=nm_flmg_act_13
 /title/tt1138464/?ref_=nm_flmg_act_15
 /title/tt5912674/?ref_=nm_flmg_act_17
```

2.2 Movie Data

From the IMDB page of any movie, we will scrape its ratings, domestic gross (adjusted for inflation), release date, genre and its Motion Picture Rating.

Rotten Tomatoes and Box Office Mojo are similar websites which can complement IMDB data as required. Rotten Tomatoes even offers a public API which we plan to explore in future.

3 Approach

3.1 Features

From the IMDB dataset, features will be extracted for each actor that depicts their popularity, kind of movies acted in and its success, acting persona and physique, since these are the metrics on which movies are generally casted. Features considered:

- Average gross of all movies acted in
- Average salary
- Average rating of the movie on IMDB
- Awards won/ nominations (Academy Awards, BAFTA and Golden Globes)
- Age
- Height
- Genres of movies acted in

3.2 Similarity Measurement

A **feature vector** will be constructed from aforementioned attributes. They will be normalized, brought down to the same scale and then fed into a **weighted**[2] **K means clustering**. The clusters will yield actors who are similar to each other.

[2] weighing methodology will be determined by heuristics

3.3 Finding Best Match within a Cluster

The above model will always return the same actor (nearest point in the cluster to a given point) as a replacement for a given input actor irrespective of the type of the movie. For example Robert Downey Jr. has acted in both 'Chaplin' and 'Iron Man'. But the actor who is found most similar to Robert Downey Jr. may not be the best replacement for both the movies since one is a biopic while the other is a superhero movie.

We have identified two metrics - Age and Genre to determine the best fit within a cluster. Age and genre will be used to construct a **Ranking Function**. The actors within a cluster will be ranked by a **weighted Euclidean distance metrics** where Age will be weighed slightly more than Genre. (It is better to suggest Adam Sandler as a replacement for Christian Bale in Batman than to replace Adam Sandler as Macaulay Culkin who played the lead in Home Alone)

3.3.1 Genre fit

This will help us determine the most suitable actor in the cluster depending on the genre of the movie being recast. Example - In Robert Downey Jr's cluster, the ranking function will take into account the movie's genre and only then yield the most suitable replacement. Therefore it will likely suggest different replacements for movies like 'Iron Man' which is a Superhero movie and 'Kiss Kiss Bang Bang' which is a dark Comedy.

3.3.2 Age fit

The model will take into account the age of the actor who played the role originally. Then it will adjust the age of all other actors in the cluster according to the given input year i.e. the model will evaluate their age for the year in which the movie is being remade. If we are recasting James Bond in 2016, the model shouldn't yield Clint Eastwood since he is isn't of the right age for the role. However, he might have been a suitable replacement four decades back. Similarly if a movie is being recast in 1930s, then Leonardo Di Caprio shouldn't be suggested as a replacement since he was yet to be born then.

Adjusting the age of the actors according to the input year will help us handle both the cases of recasting movies in past and future.

1. **If a movie made in current era is to be recast in the past:**
   If the movie 'Inception' was to be made in the 1950s, then the replacement lead actor suggested by the model will be of a similar age bracket in 1950s as Di Caprio was when he played the lead role in Inception.

2. **When a movie made in the past, is to be recast today:**
   If the movie 'Home Alone' is to be remade today, then the model will yield an actor who is of the similar age bucket in today's time as the lead actor was when he played the lead role.

4 Testing Methodology

We plan to construct our model iteratively i.e. start with few features and keep adding more features to it to yield better results. To test our model at each iterative level we need a method to determine the quality of our output.

4.1 Testing quality of the model

Wikipedia lists 700 remakes[3] till date. From this list, Hollywood remakes will be extracted manually and used to test the model.

For each movie input and the year of its remake, our model will yield top 3 replacements. We will apply the model on all these remakes and use this to gauge the quality of our model. If the top 3 replacement suggested by the model actually contains the actor who starred in the remake, it will be considered a hit. This process will done in both forward timeline and backwards. Case in point: Start with Mad Max (1979) and use the model to see if it actually yields Tom Hardy as a replacement in 2016 and then start with Mad Max (2016) and test if the model yields Mel Gibson as a replacement in 1979.

5 Challenges:

1. It is difficult to evaluate the quality of the model. Person A might prefer Robert Downey Jr. as a replacement for Christian Bale in The Dark Knight, while Person B might think that Hugh Jackman would be a better fit for the role of Batman. Therefore, it is hard to quantify which measure is better, the first or the second?

   Due to the lack of any established methods to test such models, we plan to use remakes for the testing of our model. However, castings are often affected by various other unquantifiable factors like availability of dates, relationship of the actors with the studio and director and above all the preference of the director. During the remakes, the directors would have in all likelihood tried to cast the best actor available for that role and not the actor who was the most similar to the one who played the role originally.

2. Handling Sparse data

   - Best way to handle missing values for Box office collections and salary of actors.
   - Some missing fields like Date of Birth for actors may require manual filling of data. This problem may become particularly challenging for large number of missing values. We may need to combine data from more than one source to tackle this.

[3] https://en.wikipedia.org/wiki/Lists_of_film_remakes