CS09 805(P) PROJECT

Movie Recommendation And Prediction Website

JISHNU P
Reg. No. VEANECS032
S8 Btech CSE (2013 Admission)

Department of Computer Science & Engineering
Vidya Academy of Science & Technology
Thalakkottukara, Thrissur - 680 501
(http://www.vidyaacademy.ac.in)

Department of Computer Science & Engineering

Vidya Academy of Science & Technology

Thalakkottukara, Thrissur - 680 501

(http://www.vidyaacademy.ac.in)

CERTIFICATE

This is to certify that the report titled **Movie Recommendation And Prediction Website** is a bona-fide record of the work related to the paper CS09 805(P) PROJECT done by **JISHNU P (Reg. No. VEANECS032)** of S8 Btech CSE (2013 admission) of Vidya Academy of Science & Technology, Thrissur - 680 501 in partial fulfillment of the requirement for the award of the Degree of Bachelor of Technology of University of Calicut.

Guide/Supervisor	**Head of Department**
Name:	Name:
Ms. Greeshma Gopinath	Ms. Sunitha C
M.E, DCE, PGDCA, MISTE, Asst.Professor	M.Tech, MISTE, MCSI
Signature :	Signature : .....
Date : April 17, 2017	Date : April 17, 2017

(Seal of Department of Computer Science & Engineering)

Acknowledgement

I would like to thank Lord Almighty, the foundation of all wisdom who has been guiding me in every step. I wish to record my indebtedness and thankfulness to all who helped me to prepare this project report titled **Movie Recommendation And Prediction Website** and present it in a satisfactory way.

My sincere thanks to **Dr.Sudha Balagopalan, Ph.D, MISTE, MIEEE, MIET** Principal, for providing me all the necessary facilities. I take the opportunity to extend my thanks to **Ms. Sunitha C , M.Tech, MISTE, MCSI** Head of Computer Science & Engineering for valuable guidance in developing the project.

I am extremely thankful to our guide and supervisors **Mr. Jayakumar T V, M.Tech, MISTE, Asst.Professor** and **Ms. Greeshma Gopinath , M.E, DCE, PGDCA, MISTE, Asst.Professor** in the Department of Computer Science & Engineering for giving me valuable suggestions and critical inputs in the preparation of this report.

I express my heartfelt thanks to my Lab Instructors for their valuable support and assistance.My sincere thanks to my parents and friends who have helped me during the course of the project work and have made it a great success.

Abstract

The system predicts the success of a movie based on its profitability by leveraging historical data from various sources. The system automatically extracts several groups of features, including who are on the cast, what a movie is about, when a movie will be released, as well as hybridfeatures that match who with what, and when with what. By analysing the necessary factors it is made possible to predict the profitability of a movie, thereby availing movie producers to have more chance of success in the final product.

The recommendation system uses the data which is collected from the user of the website that is the type of movie searched,rating given and other data.These data are used to find related movies of user which is then filtered as necessory and used to recommend movies for the user.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation

The project aims to create a website which provides recommendations based on the users taste . The website also help investors to make right decisions for investing on a new movie.

1.2 Problem Statement

Investors of a movie are making decision to invest on a movie without knowing whether the movie is profitable. So their chances of making profit are very low. It is hard to search for related movies in a website.There is no web service offering both movie prediction and recommendation.

1.3 Objective

The objective of the project is to develop a movie recommendation and prediction website.

Users can access the website for searching movies. Based on the search history, the website recommend movies for the user. Also, based on the invester's choices, the system predicts the movie's success.

1.4 Summary

The rest of the project report is structured as follows.

Chapter 2 describes the Requirement Analysis. Chapter 3 presents the Literature Survey of the project. Chapter 4 contains Materials and Methods of the project.The report ends with conclusion and reference.

Chapter 2

REQUIREMENT ANALYSIS

2.1 Hardware Requirements

1. Processor:Intel i5 or above
2. Ram:4GB and above
3. System type:64bit

2.2 Software Requirements

1. Operating System:Windows 8.1 or above and Ubuntu 16.04 LTS
2. Programming language:Python 2.7
3. Libraries : Pandas, Graphlab, Mataplot
4. Framework : Django 1.11
5. Database : Sqlite3
6. For designing : HTML, CSS, JAVASCRIPT, AJAX
7. IDE:Sublime Text

2.3 Functional Requirements

1. Accessing website: A user tries to access the website.
2. Recommendation: The system recommend movies based on the user's preferences.
3. Prediction: Based on the investor's choice , the system predicts the system predicts the movie's success.

Chapter 3

LITERATURE SURVEY

3.1 Prediction

Past works have focused primarily on gross box office revenue. However the above metric ignore how much it costs to produce a movie. Some studies categorized movies into two classes based on their revenues(success or not).

Some considered prediction as a multi-class classification and classified movies into discrete categories ranging from 'blockbuster' to 'flop'. The accuracy of a predictive model depends a lot on the extraction and engineering of features (a.k.a., independent variables). When it comes to studying movie success, three types of features have been explored: audience-based, release-based, and moviebased features.

Audience-based features are about potential audiences reception of a movie. The more optimistic, positive, or excited the audiences are about a movie, the more likely it is to have a higher revenue. Similarly, a movie with more pessimistic and negative receptions from the public may attract fewer people to fill seats. Such receptions can be retrieved from different types of media, such as Twitter, trailer comments from Youtube, blogs, new articles , and movie reviews . The volume of discussions, the sentiment of review or comments , as well as the star rating from reviews have been used as a means for assessing audiences excitement towards a movie.

Release-based features focus on the availability of a movie and the time of its release. One such feature that captures availability at release is the number of theaters a movie opens in. The more theaters that will show a movie, the more likely the movie will have a higher revenue. Many movies are targeted

for releases at a certain time, or at a time in which they would be eligible to receive an upcoming award. Holiday release is a feature commonly utilized in the prediction problem , as are seasons and dates of releases (Spring, Summer, etc.). Some studies attempted to capture the competition at the time of release, which has the potential of negatively effecting revenues.

Movie-based features are those that are directly related to a movie itself, including who are on the cast and what the movie is about. As for cast members, the most popular feature is a movies star powerwhether the movie casts star actors. Star powers of actors have been captured by actor earnings, past award nominations, actor rankings, and the number of actors Twitter followers. It was agreed that higher star powers are helpful for a movies success. However, no research has explored the profitability of actors. As it costs a great amount of money to cast a famous actor, we believe an actors record of profitability will be a better indicator of a movies profitability than her record in generating revenues. Moreover, the role of directors in a movies financial success is often overlooked or downplayed. While some research has investigated the individual success of directors , few studies have actually tried to connect directors star powers to movies financial success. Some argued that the economic performance of movies is not affected by the presence of star directors , and directors values are not as important as actors values for movie revenues . As directors play important roles in movie productions , this research examined the effect of directors on movie profitability.

In addition to individual actors and directors, the cast of a movie has also been explored from a teamwork perspective whether individuals in a team can work together and develop team chemistry . Studies of organizations and teams have revealed that team members prior experience or expertise is beneficial for team success, while the diversity of a team helps too, especially in the context of bringing creative ideas and unique experience to teams for scientific research and performing arts. Both diversity and familiarity of a cast contribute to a directors success in receiving awards. Similarly, for movies financial success, the diversity of a cast is positively correlated with the movies box-office revenue , and cast members previous experience also positively influences revenues. Nevertheless, there are several important limitations to consider. On one hand, many of the measurements for teamwork were simplistic and problematic. For example, an actors experience was based solely on the number of previous movie appearances, without considering what types of movies she has contributed to, and thus has more experience in. Also, team members degree dispersions were used to reflect a teams diversity even though a team composed of actors who have never collaborated with each other can still feature a uniform degree distribution. Although

the existence of structural holes can reflect a teams diversity, the measurement of structural holes was simplified to the density of a network. Nevertheless, the two concepts are only very loosely related. On the other hand, the data size was small in many studies. For instance, the top 10 movies (by revenue) in each year (a total sample size of 160-180 movies) were studied in. With such a small sample, an actors experience and previous collaboration cannot be completely captured. The selection bias towards more successful movies also hurt the validity of the results. Thus in this research, we leveraged much larger datasets, derived new and more accurate ways to capture individual actors experience and teams diversity, and related them to movie profitability.

In terms of what a movie is about, features such as genre, MPAA rating, whether or not a movie is a sequel, and run time have often been incorporated into success predictions. Besides such meta data about a movie, to get a better idea of a movies content, one needs to examine its plot or script. Two earlier studies leveraged the texts of movie scripts for success predictions. Some of the basic text-based features are easy to obtain, such as the number of words, number of characters, number of sentences, and the amount of dialogue present. Nevertheless, more informative textual features in these studies depend on manual annotations by human experts, such as the degree to which the story or hero is logical, the degree to which the premise is clear, the degree to which violence is present, and whether or not the story has a believable ending. As movie scripts can be very long, the manual annotations are time-consuming. Also only a small number of movies scripts are available in a uniform and professional format. Thus a predictive model based on features from scripts can only be trained on a small pool of movies, which may limit its predictive power for future movies. Therefore an automated way to analyze openly available texts about a movies content is necessary for a decision support system that needs to learn from large-scale datasets.

For our research question of predicting movie profitability at an early stage, we cannot take advantages of most audience-based features and some of the release-based features, as they would not be available when making investment decisions. For instance, YouTube comments only appear after a movie trailer is released; likewise, the number of theaters a movie is going to be released in will not be known until the end of the movies production. In addition, these features from different groups were treated as standalone and independent, whereas the interaction or match between features from different groups, such as actors star powers along with their experience with different movie genres, or the popularity of a certain type of movie during a specific time period, can provide valuable information

about a movies success.

Therefore, we will focus mainly on four types of features: Who features who is involved in a movie, When features when a movie will be released, What features from both meta data and texts of movie plot synopses (movie plot synopses are openly available from most movie data archives, yet they can still reflect movies content), as well as Hybrid featuresthe match between What and Who and the match between What and When. Our feature set includes popular features from the literature (e.g., measuring actor star powers using their total gross revenues), new features proposed to better measure previously proven factors for movie success (e.g., team expertise and diversity), as well as features representing new factors that may be related to movie success (e.g., actor-director collaboration, and market trend by genre). All the features adopted by our system can be extracted in an automated fashion by using text mining and social network analysis techniques. In addition, from a theoretical perspective, this study also examined whether previous findings about star powers of actors and directors, and teamwork are still valid when movie success is measured by profit, instead of revenues, based on a much larger dataset.

3.1.1 Who feature

The very nature of the movie industry is characterized by people who make movies. Successful actors and directors, such as Brad Pitt, George Clooney, and Christopher Nolan, are crowd favorites who are well known throughout the world. Talented individuals such as these can leverage not only their refined industry skills to make high-quality movies, but also the associated name brand effect, which draws crowds and increases sales . This effect is typically referred to as star power. Because our goal is to predict profitability, our star power features for a movie are based on its cast members records in generating both box-office revenues and profits.

1. **Tenure** of an actor reflects how much experience she/he may have in the industry. It is calculated as the time difference (in years) between the movie in which an actor most recently appeared and that in which he/she first ever appeared. For each movie, we calculate the average and total tenure for its first-billed cast actors.

2. **Actor Gross** is about how much revenue an actor has generated during her/his tenure. Each individuals total gross is the sum of revenues from all

the movies that she has starred in, while an individuals average gross is her total gross divided by the number of movies she starred in. For each movie, we calculated the sum and average of total gross, as well as the average of actor average gross, for all first-billed cast members.

3. **Director Gross** measures the past success of directors. We calculated for each director the total and average gross for movies she/he has directed.

4. **Actor Profit** measures the amount of profit an actor has earned through his/her career before the movie to be predicted. For each actor, we derived total profit, average profit, and top profit the profit of the most profitable movie the actor has appeared in.

5. **Director Profit** represents the amount of profit that a director has earned before the movie to be predicted. Similar to actors, we considered total profit, average profit, and top profit for each director.

3.1.2 What Features

In addition to who are in the cast, another natural and important indicator of a movies future profitability is what the movie is about. Such information is usually available with high certainty prior to movie funding efforts. To reflect what a movie is about, the what features in our model include both meta features, such as genre (e.g., action, sci-fi, family) and rating (e.g., PG13, and R), but also fine-grained description of a movies contentits plot synopsis.

In text mining, texts from plot synopses can be represented as traditional unigrams and bigrams, but such representation will have high dimensionality and, as a result, suffers from sparsity. At a higher level, topic model techniques, such as Latent Dirichlet Allocation (LDA) , can give a better picture of what a plot is about. The input for LDA is a textual corpus from plot synopses and the output is a group of topics, each being represented by a probabilistic distribution over words. Those words with high probabilities on a topic are considered representative keywords for the topic. Each plot synopsis is also assigned a probabilistic distribution over all the topics. Such topic distribution vector of a movies plot reflects the content of the movie at an aggregated level and can be used as features for predictive modeling. In addition to these topics derived from LDA, some movies plots are adaptations from other sources, especially when the original sources had achieved certain levels of success. For example, The Hunger Games and Harry Potter are both adapted from best-selling novels. As such, one of our what features was

about adaptations: whether a movies plot was adapted from a comic, a true story, or a book/novel.

Genre is category of artistic composition, as in music or literature, characterized by similarities in form, style, or subject matter. The genres of films were categorized as: Adventure, Musical, Action, Romance, Teen, Romantic Comedy, Biography, Animation, Comedy, Crime, Documentary, Period / Drama, Drama, Horror, Family, Science Fiction, Fantasy Each nation depending on the culture and preferences of its citizen might have a different taste of movie genre in motion picture industry. Some countries appeared to produce culturally distinctive versions of genres formats. In Italy for example locally produced comedies performed very well but could be described as sex comedies and are unlike comedies produced in other territories.

Once it is looked at the statistic we see from the below table 3.1 that Comedy is the most common genre between movies however it is not the highest grossing genre. The highest grossing genres in box office are Adventure ($73,319,913), action ($56,257,259), romantic comedy ($28,007,155) and horror. The least average grossing movie genre on the contrary is drama even though in quantity it is the most preferred one which is a contradiction to mall over.

MPAA (Motion Picture Association of America) : MPAA rates the movies according to films' thematic and content suitability for certain audiences. The primary MPAA ratings are G (General Audiences), PG (Parental Guidance Suggested/Some material might not be suitable for children), PG-13 (Parents Strongly Cautioned/Some material may be inappropriate for children under the age of 13), R (Restricted/Under 17 not admitted without parent or adult guardian), and NC-17 (No One 17 and Under Admitted) (Source: Wikipedia). Many film companies re-edit or re-shot their movies in order to increase their ratings to PG and PG-13 because these ratings exclude virtually no one from seeing the movies. Sawhney and Eliashberg (1996) found in their study that movies with restricted rating (R) bring lower box office revenue compare to the unrestricted ones. One the other hand, Litman (1999) in his study found no significant relationship between MPAA ratings and box office performance.

R is by far the most common rating, half of the movies within the whole industry is rated with R. It is followed by PG-13, G is the least frequent rating among films. (De Vany, Arthur; Walls, W, David 1999) NC-17 is extremely rare between Hollywood movies because it restricts the movieto be seen by people under age 17. What is more he most common genre is drama followed by comedy which is usually rated with PG-13 and R according to the aforementioned study. On the other hand, the only genre which has been found significant regarding the

Table 3.1: Top-Grossing Genres of movies shot from 1995 to 2012

		Movies	Total Gross	Average Gross	Market Share
1	Comedy	1,751	$44,792,158,044	$25,580,901	23.48%
2	Adventure	521	$38,199,674,469	$73,319,913	20.03%
3	Drama	3,132	$33,621,012,632	$10,734,678	17.63%
4	Action	570	$32,066,637,809	$56,257,259	16.81%
5	Thriller/Suspense	561	$15,495,734,985	$27,621,631	8.12%
6	Romantic Comedy	403	$11,286,883,357	$28,007,155	5.92%
7	Horror	329	$9,093,205,812	$27,638,923	4.77%
8	Documentary	1,076	$2,063,950,710	$1,918,170	1.08%
9	Musical	113	$1,865,013,970	$16,504,548	0.98%
10	Black Comedy	85	$781,440,299	$9,193,415	0.41%
11	Western	36	$685,432,870	$19,039,802	0.36%
12	Concert/Performance	41	$293,960,413	$7,169,766	0.15%
13	Multiple Genres	20	$8,280,303	$414,015	0.00%
14	Genre Unknown	5	$1,685,983	$337,197	0.00%

Source: www.the-numbers.com

box office revenue is the science fiction genre in Litman's study (Litman, 1983) while thriller has been found the most popular genre and romance is the least popular genre found in Neelamegham and Chinatagunta's study (Neelamegham and Chinatagunta, 1999). Briefly as in all the variables we have Genre and MPAA ratings and their significance are also quite controversial in literature. Analst in his study found that movies which include erotic scenes and violence in it attracts more audiences Anast (1967).

On the other hand statistically speaking the table has been taken from www.the-numbers.com from where we also took the budget and box office revenue data of our case studies. All the movies shot between 1995 and 2012 can be seen on the table 3.2 which are grouped according to their MPPA rating. Accordingly, the majority of the movies are PG-13 rated which is followed by R (restricted) rated movies. It is evident that the highest average revenue gross ($42,384,195) is realized by PG-13 rated films which is followed by R rated films ($15,337,934).

Because of this evidence that PG-13 and R rated films bring higher average gross, most of the films by movie makers are tried to be fit in PG-13 and R rated category.

Table 3.2: Top-Grossing MPAA Ratings 1995 to 2012

	Movies	Total Gross	Average Gross	Market Share
1 PG-13	2,028	$85,955,147,762	$42,384,195	45.08%
2 R	3,575	$54,833,115,390	$15,337,934	28.76%
3 PG	986	$36,830,727,925	$37,353,679	19.32%
4 G	276	$10,634,593,071	$38,531,134	5.58%
5 Not Rated	2,279	$1,760,744,156	$772,595	0.92%
6 NC-17	21	$72,872,987	$3,470,142	0.04%
7 Open	5	$7,678,311	$1,535,662	0.00%

Source: www.the-number.com

3.1.3 When Features

With the movie industry being an avenue for entertainment, its market sees peaks and declines over time, which may speak to how well a pre-production movie may fare in the future. Thus we incorporated the following when features in our model:

1. **Average Annual Profit**: Average annual income is the average profit across all movies in the year prior to the planned release of movie m, where m is a movie released at any time of year. This feature captures the overall profitability of the movie industry before a movie is released.

2. **Release dates**: combines several features about when a movie will be released, including whether it will be a holiday release and which season of the year (spring, summer, fall, winter). While a holiday or summer release may attract more of an audience and thus generate more revenues , it also

requires higher budget for marketing and distributions during these competitive periods. Although the exact release date is not completely definitive before filming, a target trajectory usually exists at the early stage of movie production.

3. **Time of release** : The most important and secure time to release a film is Christmas time(Litman, 1983) . However since all the movie makers aim to release their films around Christmas time or peak times, the competition rises during these seasons. Many films if they are not as strong as Hollywood movies in marketing terms in order to avoid the competition do not released in Christmas time. In addition, it is rarely seen that in Christmas time or any time of the year we see two blockbusters released at the same week. Movie industry is a well informed market, before the release of a movie the industry is publicly informed about the release time so others movies strategically behave. On the contrary some authors believe that the best time to release the film is summer season (Sochay, 1994) . Sochay adds that the peak time or season of the movie industry changes from year to year depending on competition and hence the strategically releasing times. The literature mainly gathers around the idea that during the peak holiday times of the year movies bring higher box office revenue.

3.1.4 Hybrid Features

Besides standalone features about who are in a movie, what a movie is about, and when a movie will be released, it is also important to capture the match between these features. Our hybrid features try to reflect such matches between what and who, as well as between what and when. For example, it may be important to form a team of actors based on their previous experience with the genre of the movie being planned, instead of just their star powers. Similarly, the investment on a movie whose genre is gaining popularity may increase the chance of success

1. What + Who In observing the movie industry as a whole, and the actors that tell the stories, we can distinguish various so-called roles that these actors seem to adopt. For example, Seth Rogan is typified by his appearance in comedies, and Bruce Willis exhibits a proficiency as an action movie star. Should a movie then, granted this observation, try to include those who have extensive experience in its genre? Or conversely, does a surprising cast draw a greater audience to theaters (e.g., having Bruce Willis in a comedy or having action star Arnold Schwartzenegger in a romantic love story)?

Although these questions have not been addressed in the literature, we believe that better measurements of an actors expertise with regard to movie genres can help us more accurately determine the expertise and diversity of a movies cast.

2. What + When Similar to the overall market volume for movies, which may change from year to year, consumers preferences of movies may also evolve with time. For example, while movies like American Pie and National Lampoons Van Wilder were popular in the late 90s and early 2000s, movie-goers recently have been flocking to horror movies, such as Paranormal Activity, and those characterized by superheroes, such as The Avengers and Captain America. Although the latter category is nothing definitively new to the silver screen, the movie industry has seen greater levels of success in recent years with this particular focus and, as such, a greater influx of such movies. Meanwhile, competitions may also affect the profitability of movie m because other movies released during a similar time period may detract from movie ms viewer-base.

   Thus, in addition to capturing when a movie will be released, we also consider how movies with similar genre performed in the previous year, as well as the level of competition during a movies planned release time.

The block diagram shown in figure 3.1 shows the basic process involved in movie prediction for the user. The data taken from the structured database is categorized into the main features of the film that is what, who, when and hybrid, which is a combination of other features. These features are analysed in the predictive model using the specified algorithm. This process of using domain knowledge of the data to create features is called feature engineering. using this method it is able to analyse the various main features and its sub features to generate a near practical result.

3.2 Recommendation

Recommendation can also be given by taking into consideration time of the day, users mood etc. In recommendation as well as predictions, various statistical processes for estimating the relationships among variables are considered.

In traditional recommender systems the recommendation procedure is done with considering about users past rating history and items features (tagging) and

basically no contextual information is taken into account for generating recommendations. With the creation of the new generation recommender system, contextual information has become one of the most valuable knowledge sources to improve recommendations and provide more user specific recommendations under that similar circumstance (i.e., contexts) are related with similar user preferences.

In recommender systems variety of research had been conducted to deliver the most relevant content to user in different recommendation scopes for users, such as music recommendation, location-based services or even recommendation system for tourism. Application of recommender system are found in variety of application today, such as in youtube, amazon sales etc. These take into consideration the users database inorder to personalize the search. The abundance of information available on the Web and in Digital Libraries, in combination with their dynamic and heterogeneous nature, has determined a rapidly increasing difficulty in finding what we want when we need it and in a manner which best meets our requirements. As a consequence, the role of user modeling and personalized information access is becoming crucial: users need a personalized support in sifting through large amounts of available information, according to their interests and tastes. Many information sources embody recommender systems as a way of personalizing their content for users.

Recommender systems have the effect of guiding users in a personalized way to interesting or useful objects in a large space of possible options. Recommendation algorithms use input about a customers interests to generate a list of recommended items. At Amazon.com, recommendation algorithms are used to personalize the online store for each customer, for example showing programming titles to a software engineer and baby toys to a new mother .The problem of recommending items has been studied extensively, and two main paradigms have emerged. Content-based recommendation systems try to recommend items similar to those a given user has liked in the past, whereas systems designed according to the collaborative recommendation paradigm identify users whose preferences are similar to those of the given user and recommend items they have liked.

Systems implementing a content-based recommendation approach analyze a set of documents and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the objects rated by that user . The profile is a structured representation of user interests, adopted to recommend new interesting items. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object. The result is a relevance judgment that represents the users level

of interest in that object. If a profile accurately reflects user preferences, it is of tremendous advantage for the effectiveness of an information access process. For instance, it could be used to filter search results by deciding whether a user is interested in a specific Web page or not and, in the negative case, preventing it from being displayed.

In movie recommender system some of the works focus on inputting rich and detailed meta-data (genre and other features) to the system in order to enhance the quality of recommendations. In a movie recommender system, this facility was provided to allow user to even set new keywords for movies which resulted in providing more accurate recommendations based on features which set users. The big advantage of this solution is that it adapts to changes regarding what the users find important, something which can change over time. However, manually categorization of content can be expensive, time-consuming, error-prone and highly subjective. Due to this, many systems aim to automate the procedure.

Another approach of providing high quality recommendation can be achieved by utilizing contextual-information such as time of the day, users mood, etc. According to most of the users who are located in the same geographical location have the similar taste to each other. For instance, users who are located in Florida mostly like to watch Fantasy, Animation types of the movie however, the user from Minnesota State are mostly interested to War, Drama type of the movie.

Therefore, in recommending items the location of the user is not considered in any steps, as a result it is possible for a user who is looking for a resultant in Chicago get recommendations about some restaurants that are located in other areas such as Seattle. Various algorithms examine by researchers in order to enhance the traditional recommender systems. For instance, authors presented a novel hybrid approach for movie recommender systems whereby items are recommended based on the collaborative between users as well as analysing contents. In another work by Bogers, an algorithm based on Markov random walk proposed which applied on a movie dataset by considering regarding some contextual information. In addition, Rendle et al. applied matrix factorization in a movie dataset with the aim of enhancing the quality and accuracy of the recommendation as well as reducing the complexity of building their model. Simon Funk proposed Regularized singular value decomposition (RSVD) to enhance the accuracy of the recommendation which reduces the dimensionality of the original rating matrix.

The block diagram shown in figure 3.2 shows the basic process involved in recommendation of a movie. The properties of the movie files are extracted from users database, then it is related with the data from the movie database using the algorithm given. Then the most related movies to the given movie files are

displayed.

3.3 Relevence

The certainty with regard to movies success is largely uncertain, despite the capital investment. with hits and flops are released every year. If there is a better way to analyse the movie success this will lead to large success rates in film industry.

Researchers have undertaken task of predicting movie success using various approaches. However from an investors point of view, one would want to be as assured as possible that his/her investment will ultimately lead to returns or profits.

Traditional recommender systems considered users past rating history and items features. Hence, the users search would be more personalized with his preferred movie genre.

The main objectives of the system can be narrowed down to the points given below.

1. To predict the movie success before the movie is even made.
2. To help investors to make better investments.
3. To help users to interact with movies and to give recommendation for them based on their tastes.

3.4 Overview

A website is made for user interactions to the system. The website lists the movies and also recommends movies based on the search preferences. The website also takes attributes of new movie as input to predict the movie profitability.

Chapter 4

MATERIALS & METHODS

4.1 Proposed System

In this project, we propose a website where user can rate movies listed on the website and get recommendations based on that rating. A search functionality is added so that the user can easily find the movies that the user likes and he can also view all the movies that the user has rated.

In this website the user can also get an approximate profitability metric(ROI) of future movies. The user can select the director and actors of some movie that is to be predicted and the profitability along with the selected director and actors are displayed in the website.

4.2 Design Discription

4.2.1 Prediction

1. **Cleaning**: In this phase the dataset obtained from www.kaggle.com is cleaned by removing unwanted features and removing rows with missing information. As if we remove all the rows with missing values the no of items in the dataset reduces so we have performed mean imputation on some of the features so that we can get a considerable amount of data that we can use for our machine learning model.

2. **Feature Extraction**: In this phase as the features in the dataset are not strong to implement a prediction model we have created features that are

relevant to the model from the cleaned dataset.

3. **Prediction**: LASSO (least absolute shrinkage and selection operator) regression is used as the machine learning model for prediction. We have selected LASSO because it can penalise those features that arent relevant to the system. LASSO will also take care of those features that have multicollinearity.

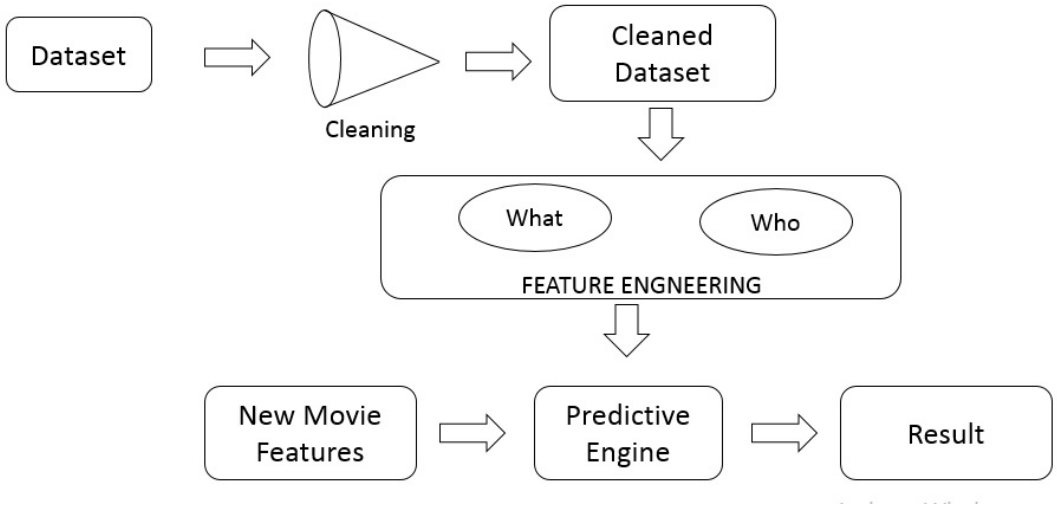

Figure 4.1: Prediction System

4.2.2 Recommendation

1. **Feature selection**: The dataset which is obtained from https://grouplens.org/datasets/movielens/ contains mainly three datasets and for recommendation module we need the ratings.csv dataset which has features such as user_id,movie_id,rating. These deatures are selected for recommending movies to a user.

2. **Recommendation**: The recommendations are obtained by using a python library called graphlab which is mainly used for recommendation systems.The

input to the recommendation algorithm are the features that we have explained above. After training the system with the features the system can recommend movies to the user_id that is provided to the system.

4.3 Dataflow Diagram

4.3.1 level 0 DFD

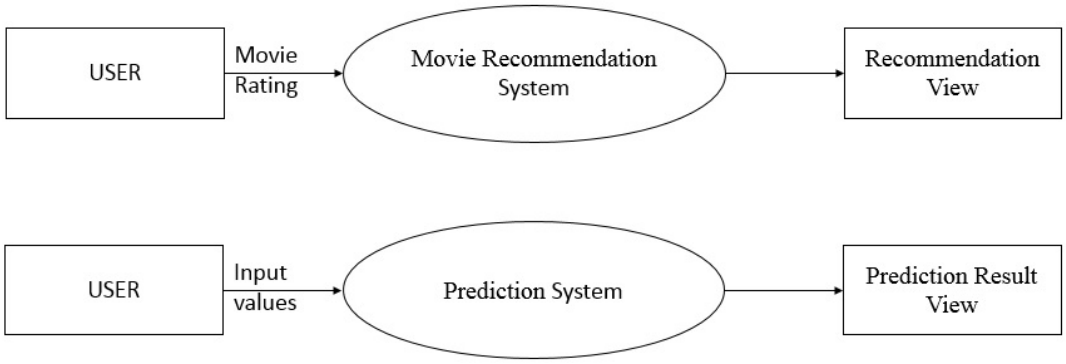

Figure 4.2: level 0 dfd

In the figure 4.2, the level 0 dataflow diagram shows the basic layout of the system. The prediction and recommendation process are done separately. In the recommendation system cleaned dataset is given as input. In prediction system the user give input movie data for prediction and the output is based on the training dataset given.

4.3.2 level 1 DFD

In figure 4.3 and 4.4, the level 1 dataflow diagram of recommendation and prediction systems are shown.

In recommendation system, the user login to give ratings for various movies. According to the ratings given, the movies are recommended to the user.

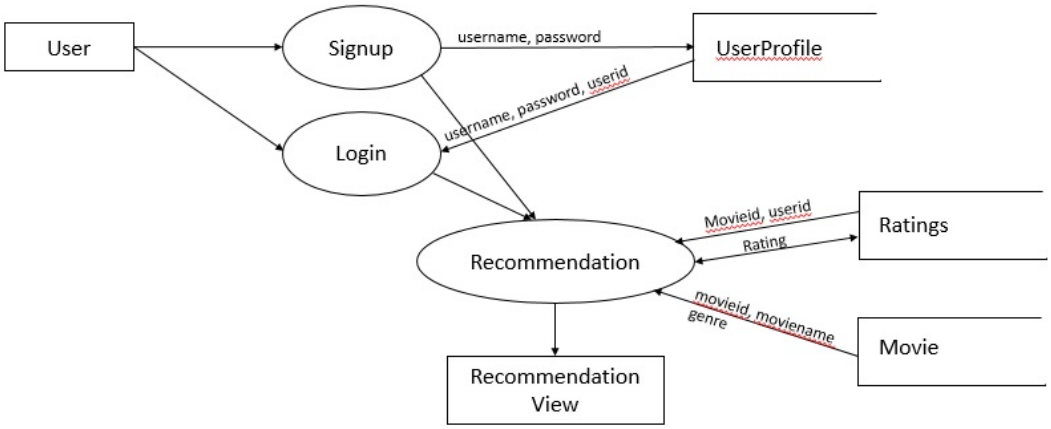

Figure 4.3: levlel 1 Recommendation System

In prediction system, the user gives input movie name, director name and 3 actor names. The average profit and number of movies of the director and actors are analyzed and prediction is made.

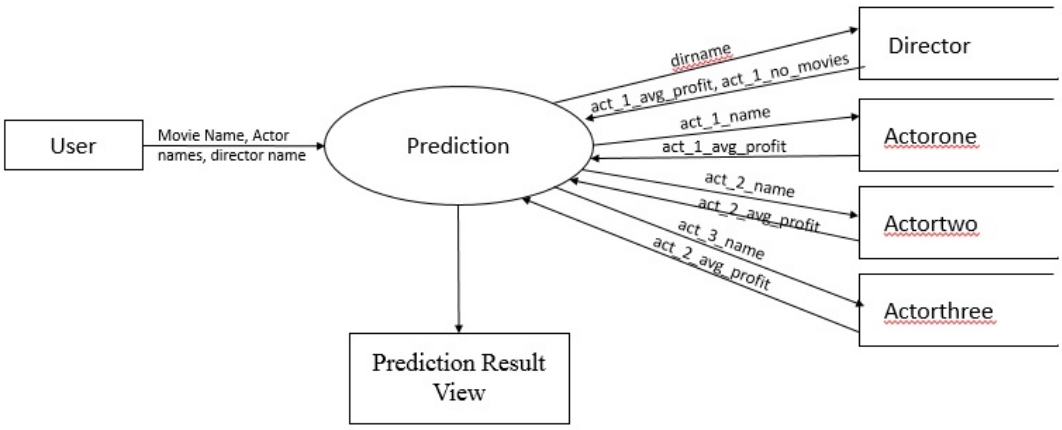

Figure 4.4: level 1 Prediction System

4.3.3 ER Diagram

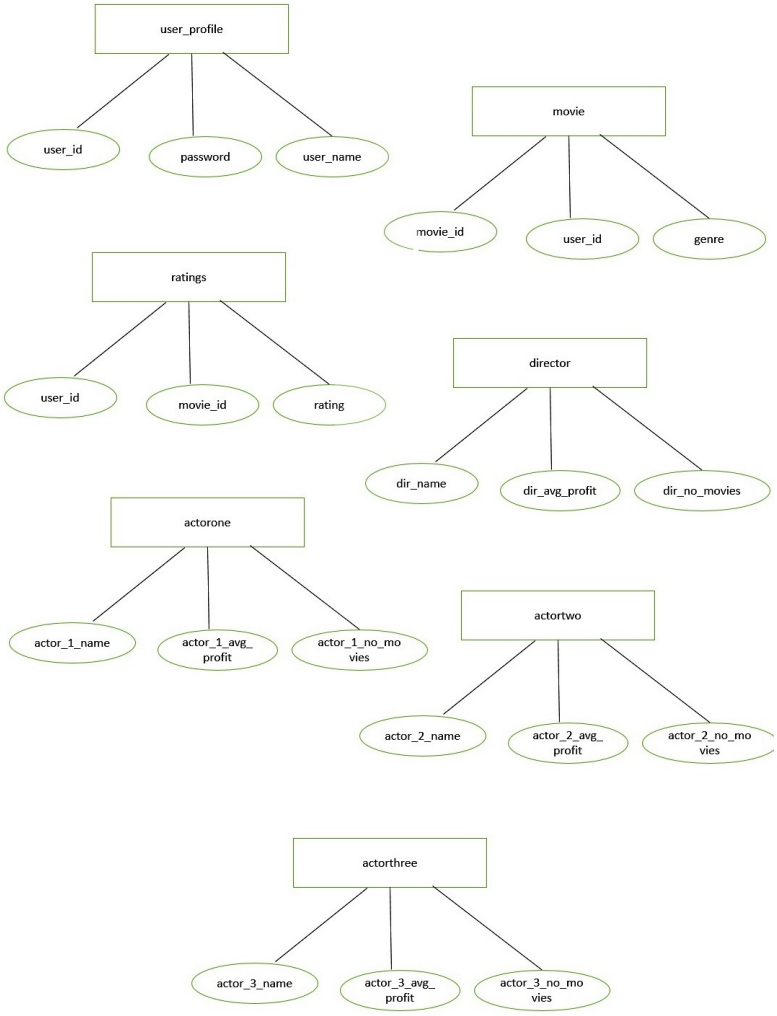

Figure 4.5: ER Diagram

Chapter 5

Implementation Details

5.1 Cleaning

Data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Some data cleansing solutions will clean data by cross checking with a validated data set. A common data cleansing practice is data enhancement, where data is made more complete by adding related information

The movie dataset is collected from Kaggle and from Grouplens websites. In the cleaning process the dataset is analyzed. Unwanted attributes such as color of the movie, number of critic for reviews, director facebook likes, actor facebook likes, etc. are removed. Also movies with genre documentary is removed. Movies of language English is selected all other languages are not taken. Nan values of the attributes such as budget and gross revenue are filled by taking the mean values. In the case if other crucial non replacable values such as director name or actor name are missing, then the whole column need to be removed.

The movies of a certain time period is taken under consideration. The movies from before that time need to be rejected because the actors and directors of that time might not be alive at this time period and the revenue income as well as the budget of the films at that time have less changes.

The plot shown in figure shows that the movies from time period 1990 to 2015 have changes in their representation. In both plots year vs budget and also in year vs gross a meaningful change happens in this time period. Hence the movies of

years 1990 to 2015 is taken into consideration

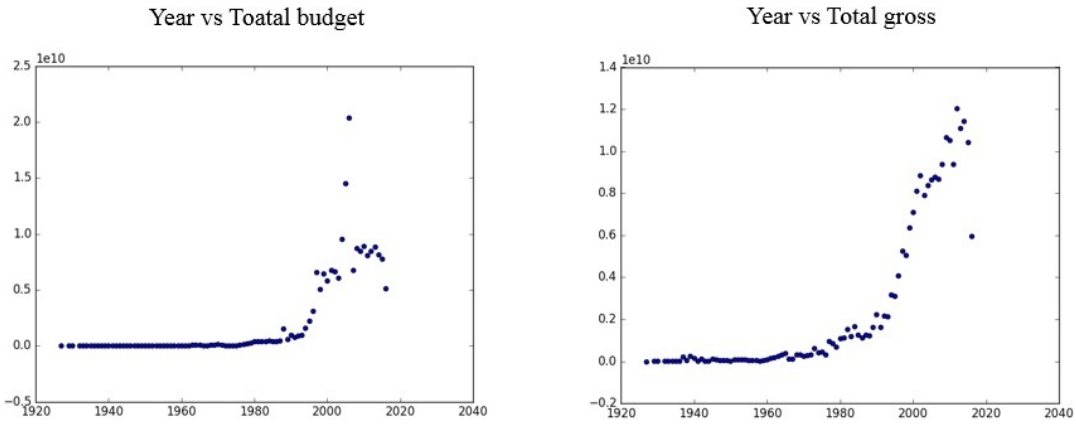

Figure 5.1: Graphical representation of movie revenue

5.2 Feature Engineering

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive.

In this process, the features of movie is taken in such a way that, the features which have more effect on the functionality is taken. In the case of predicting the sucess of a movie, the director average profit is a crucial factor also the number of movies of the director, same is the case for actors taken into consideration. Only the most relevent values are taken into cosideration.

Return of interest:The success of a movie is calculated by r.o.i. it is a measure of movie's profitability. With both revenue and budget data for each movie in our dataset, we can certainly calculate the row profit value for each movie. There are only some movies in our dataset having positive profits so investors do need a decision support system help then pick the right movie to invest in. Neverthless, while using profit values in intuitive gaining a profit of $10,000 from a movie that

costs $1 million to produce is certainly not an attractive investment this in our experiments we adopted a popular metric of profitability ”return on investment”.

It considers both profit and budget and the higher the R.O.I is the more profitable a movie is, and vice versa. Interestingly the data suggests that profitability as measured by R.O.I is not necessarily reflected by box office revenues. Having a Great box office revenue does not necessarily mean R.O.I. This further highlights the need for accurate prediction of profitability. The ROI equation is as given below.

$$R.O.I = \frac{Gross - Budget}{Budget}$$

5.3 Recommendation

In recommendation system the movies are listed in the website where users can rate them. Based on that rating Movies are recommended to users. The dataset used for recommendation is https://grouplens.org/datasets/movielens/. The dataset contains 100,000 ratings and 1,300 tag applications applied to 9,000 movies by 700 users. We have used graphlab for creating an appropriate model for recommendation.

The core idea works in 2 steps:

1. Find similar items by using a similarity metric3

2. For a user, recommend the items most similar to the items (s)he already likes

To give you a high level overview, this is done by making an item-item matrix in which we keep a record of the pair of items which were rated together. In this case, an item is a movie. Once we have the matrix, we use it to determine the best recommendations for a user based on the movies he has already rated. Note that there a few more things to take care in actual implementation which would require deeper mathematical introspection.

There are 3 types of item similarity metrics supported by graphlab. These are:

1. Jaccard Similarity:

- Similarity is based on the number of users which have rated item A and B divided by the number of users who have rated either A or B
- It is typically used where we dont have a numeric rating but just a boolean value like a product being bought or an add being clicked

2. Cosine Similarity:
   - Similarity is the cosine of the angle between the 2 vectors of the item vectors of A and B
   - Closer the vectors, smaller will be the angle and larger the cosine
3. Pearson Similarity
   - Similarity is the pearson coefficient between the two vectors.

And we have used pearson similarity which is shown below.

Where :
n = Number of pairs of scores

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\, n\Sigma x^2 - (\Sigma x)^2 \,]\,[\, n\Sigma y^2 - (\Sigma y)^2 \,]}}$$

5.4 Prediction

In the prediction system, it takes in input from user. The input taken are movie name, director name and 3 billing actors name. After correlation analysis we take only the attributes budget, movie count, avaerage profit. From the database the profit and number of movies of the director and actors are got for the analysis. In this method the movies taken are from 1990 onwards.

The regression used is lasso. In statistics and machine learning, lasso (least absolute shrinkage and selection operator) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. Lasso regression is to avoid unnecessary attrubutes and to get independant features.

Often we want conduct a process called regularization, wherein we penalize the number of features in a model in order to only keep the most important features. This can be particularly important when you have a dataset with 100,000+ features.Lasso regression is a common modeling technique to do regularization. The math behind it is pretty interesting, but practically, what you need to know is that Lasso regression comes with a parameter, alpha, and the higher the alpha, the most feature coefficients are zero. That is, when alpha is 0, Lasso regression produces the same coefficients as a linear regression. When alpha is very very large, all coefficients are zero.

the lasso cost function and lasso hypothesis is as given below

lasso hypothesis , $h_\theta(x)$ or $\hat{y} = \sum_{i=0}^{N} \theta_i x_i$

Lasso cost function, $\cos t = \frac{1}{2m}\left[\sum_{i=1}^{M}(\hat{y}^i - y^i)^2 + \lambda\sum_{j=1}^{N}\left|\theta_j\right|\right]$

$\hat{y}$ = budget*0 + director_avg_profit*0.02455 + director_movies_count*0 + actor1_avg_profit* 0.1136 + actor2_avg_profit*0.2318 + actor3_avg_profit*0.6270+ actor1_movies_count*0+ actor2_movies_count*0 + actor3_movies_count*0 - 0.05689

5.5 Tools Used

5.5.1 Django

Django was released publicly under a BSD license in July 2005. The framework was named after guitarist Django Reinhardt. Django is a high-level Python Web

framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. Its free and open source web framework written in python, which follow model-view-template (MVT) architectural pattern. The core Django framework can be seen as an MVC architecture. It consists of an object-relational mapper(ORM) that mediates between data models(defined as Python classes) and a relational database(”Model”), a system for processing HTTP requests with a web templating system(”View”), and a regular-expression-based URL dispatcher (”Controller”).

Also included in the core framework are:

- a lightweight and standalone web server for development and testing.
- a form serialization and validation system that can translate between HTML forms and values suitable for storage in the database.
- a template system that utilizes the concept of inheritance borrowed from object-oriented programming.
- a caching framework that can use any of several cache methods.
- support for middleware classes that can intervene at various stages of request processing and carry out custom functions.
- an internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals.
- an internationalization system, including translations of Django’s own components into a variety of languages.
- A serialization system that can produce and read XML and/or JSON representations of Django model instances.
- a system for extending the capabilities of the template engine.
- an interface to Python’s built in unit test framework.

Django’s primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and ”pluggability” of components, rapid development, and the principle of don’t repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically

through introspection and configured via admin models. Some well-known sites that use Django include the Public Broadcasting Service, Pinterest, Instagram, Mozilla, TheWashingtonTimes, Bitbucket, and Nextdoor. Two architectures are MVT and MVC.

- MVT

  Models: Describes your data structure/database schema.

  Views: Controls what a user sees.

  Templates:How a user sees it.

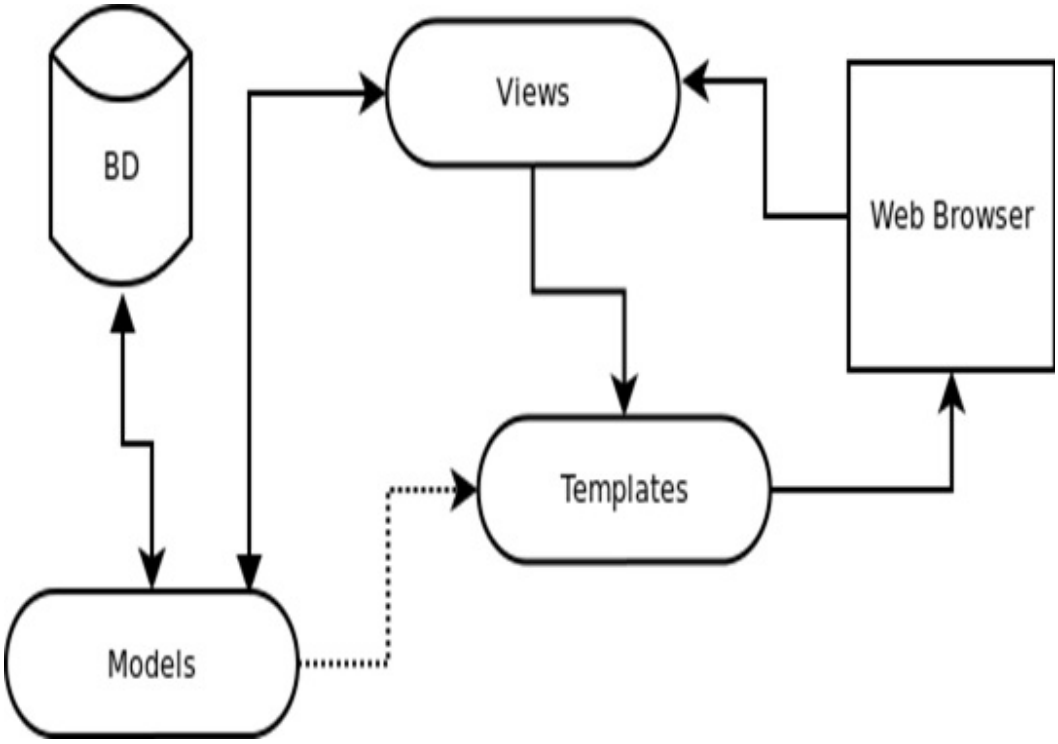

Figure 5.2: MVT Pattern

- MVC

  Models: Describes your data structure/database schema.

  Views: Controls what a user sees.

  Controller: The Django Framework,URL parsing.

5.5.2 Python 2.7

Python 2.7.0 was released on July 3rd, 2010. Python 2.7 is scheduled to be the last major version in the 2.x series before it moves into an extended maintenance period. This release contains many of the features that were first released in Python 3.1. Improvements in this release include:

- An ordered dictionary type.
- New unit test features including test skipping, new assert methods, and test discovery.
- A much faster I/O module.
- Automatic numbering of fields in the str.format() method.
- Float repr improvements back-ported from 3.x.
- Tile support for Tkinter.
- A back-port of the memory view object from 3.x.
- Set literals.
- Set and dictionary comprehensions.
- Dictionary views.
- New syntax for nested with statements.
- The sysconfig module.

SCIKIT-LEARN: The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. Its name stems from the notion that it is a ”SciKit” (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy. The original codebase was later rewritten by other developers. Of the various scikits, scikit-learn as well as scikit-image were described as ”well-maintained and popular” in November 2012. As of 2017, scikit-learn is under active development.

Scikit-learn is largely written in Python, used for machine learning with some core algorithms written in Cython to achieve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

MATHPLOT: MathPlot is an add-on library which provides a framework for easy plotting of mathematical functions, sampled data and generic 2D plots.

PANDAS: pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

pandas is a NUMFocus sponsored project. This will help ensure the success of development of pandas as a world-class open-source project.

GraphLab: The GraphLab Create recommender toolkit provides a unified interface to train a variety of recommender models and use them to make recommendations. Recommender models can be created using graphlab.recommender.create() or loaded from a previously saved model using graphlab.load_model(). The input data must be an SFrame with a column containing user ids, a column containing item ids, and optionally a column containing target values such as movie ratings, etc. When a target is not provided (as is the case in implicit feedback settings), then a collaborative filtering model based on item-item similarity is returned.

A recommender model object can perform key tasks including predict, recommend, evaluate, and save. Model attributes and statistics may be obtained via m.get(), where m is a model object. In particular, trained model parameters may be accessed using m.get(coefficients) or equivalently m[coefficients].

sqlite3: The sqlite3 module provides a DB-API 2.0 compliant interface to the SQLite relational database. SQLite is an in-process database, designed to be embedded in applications, instead of using a separate database server program such as MySQL, PostgreSQL, or Oracle. SQLite is fast, rigorously tested, and flexible, making it suitable for prototyping and production deployment for some applications.

An SQLite database is stored as a single file on the file system. The library manages access to the file, including locking it to prevent corruption when multiple writers use it. The database is created the first time the file is accessed, but the application is responsible for managing the table definitions, or schema, within the database.

5.6 Code

```
import pandas as pd
import matplotlib.pyplot as plt
import os

def pre(df):

    z=[x.find('Documentary') for x in df['genres']]
    c=[]
    for x in z:
        if x==0:
            c.append(False)
        else:
            c.append(True)
    df=df[c]
    rem_list = ['color','num_critic_for_reviews','director_facebook_likes',
            'actor_3_facebook_likes','actor_1_facebook_likes','num_voted_users',
            'cast_total_facebook_likes','facenumber_in_poster','num_user_for_reviews',
            'actor_2_facebook_likes','movie_facebook_likes']

    df = df.drop(rem_list,1)
    impbud=df['budget'].mode()
    budget=df['budget'].fillna(impbud)
    impgross=df['gross'].mode()
    gross=df['gross'].fillna(impgross)
    imputed=pd.DataFrame({"bud":budget,"revenue":gross})
    df=df.join(imputed)
    df=df.dropna()
    df=df[df['language']=='English']
    df['profit']=df.apply (lambda row: profit (row),axis=1)
    #creates a new column profit in df which has the profit of each movie
    #initialize the variables for director profit and gross
    return df


def profit(row): #function that  computes the profit of each movie
    return (row['revenue']-row['bud'])/row['bud']

def compute(df,placeholder): #function to compute all the required attributes for machine learning

    count=[]
    tot_prof=[]
    avg_prof=[]
    top_prof=[]
    tot_gross=[]
    avg_gross=[]

    _name = placeholder + "_name"
    _tot_profit = placeholder + "_total_profit"
    _avg_profit = placeholder + "_avg_profit"
    _count = placeholder + "_movie_count"
    _tot_gross = placeholder +"_tot_gross"
    _avg_gross = placeholder +"_avg_gross"
    _top_profit = placeholder +"_top_profit"
```

```
    unique_name= df[_name].unique()#selecting the unique list of director name
    #looping to find director gross(total and average),
    #director profit(total,average and top)
    for name in unique_name:

        x=df.ix[df[_name]==name]
        #selecting the movies of each director
        gross=x['revenue'].sum()
        #finding sum of gross for  director's movies
        prof=x['profit'].sum()
        #finding sum of profit for  director's movies
        no_mov= (x[_name].count())
        #counting the number of movies of a director
        count.append(no_mov)
        #appending the number of movies to count
        tot_gross.append(gross)
        #appending to total_gross list the gross calculated above
        tot_prof.append(prof)
        #appending to total_prof list the profit calculated above
        avg_gross.append(gross/no_mov)
        #appending to avg_gross list the average gross
        avg_prof.append(prof/no_mov)
        #appending to avg_profit list the average profit
        top_prof.append(x['profit'].max())
        #appending most profitable movie profit of a director

    #print len(count),",",len(tot_gross),",",len(tot_prof),",",
    #len(avg_gross),",",len(avg_prof),",",len(top_prof)
    #print tot_prof

        #create a new dataframe with calculated values
    dfdir = pd.DataFrame({_name:unique_name,_tot_profit:tot_prof,_count:count,
        _tot_gross:tot_gross,_avg_profit:avg_prof,_avg_gross:avg_gross,_top_profit:top_prof})
    csvname = _name+'.csv'
    dfdir.to_csv(csvname)
    df = pd.merge(df,dfdir,on=_name)
    return df

def doit():
    BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
    Datpath = os.path.join(BASE_DIR,"csv/movie_metadata.csv" )
    df = pd.read_csv(Datpath)
    df = pre(df)
    dfdir=compute(df,"director")

    dfact1 = compute(dfdir,"actor_1")

    dfact2 = compute(dfact1,"actor_2")

    dfact3 = compute(dfact2,"actor_3")
    return dfact3
if __name__=="__main__":
    doit()
```

Figure 5.3: Cleaning code

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model,svm
from sklearn.cross_validation import train_test_split,cross_val_score
import clean
from collections import defaultdict
from sklearn import preprocessing
from sklearn import preprocessing
from collections import defaultdict
import pickle

def predict_profit(feature_pred=None):

    df = clean.doit()
    df=df[df['title_year']>=1990]
    df.keys()
    m=preprocessing.LabelEncoder()
    u=m.fit_transform(df['content_rating'])
    y=pd.Series(u,index=df.index)
    ya=pd.DataFrame({"Rating":y})
    df=df.join(ya)
    s = df['genres'].str.split('|').apply(pd.Series, 1)
    s=s.fillna('')
    le = defaultdict(preprocessing.LabelEncoder)
    genres_num = s.apply(lambda x: le[x.name].fit_transform(x))
    df=df.join(genres_num)
    feature = df.ix[:,['bud','director_avg_profit',
    'director_movie_count','actor_1_avg_profit',
    'actor_1_movie_count','actor_2_avg_profit'
    ,'actor_2_movie_count','actor_3_avg_profit',
    'actor_3_movie_count']]#,'title_year',0,1,2,3,4,'Rating']]
    label= df['profit']
    feat_train,feat_test,lab_train,lab_test = train_test_split(feature,label,
        random_state=1)
    regress = linear_model.LassoLarsCV(cv=10,precompute=False)
    regress.fit(feat_train,lab_train)
    sco=cross_val_score(regress,feat_test,lab_test,cv=10)
    cross_score = sco.mean()
    print "cross validated score:",cross_score
    print "coefficients:",regress.coef_
    print "intercept:",regress.intercept_

    plt.clf()
    plt.scatter(feat_train['actor_1_avg_profit'],lab_train,color='blue',label='
        training data')
    plt.scatter(feat_test['actor_1_avg_profit'],lab_test,color='red',label='
        testing data')
    plt.plot(feat_test['actor_1_avg_profit'],regress.predict(feat_test),color='
        black',linewidth='2')
    plt.xlabel('director_profit')
    plt.ylabel('profit_of_movie')
    plt.show()
    with open("prediction.pickle","wb") as f:
        pickle.dump(regress,f)
    # if predict_profit is not None:
    #     return regress.predict(predict_profit)
if __name__=="__main__":
```

Figure 5.4: Prediction code

```python
# Recommendation using graphlab
import sqlite3
import graphlab as gl
import os
import pickle

def recommend(userid):
    BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
    Datpath = os.path.join(BASE_DIR, 'db.sqlite3')
    print Datpath
    conn = sqlite3.connect(Datpath)
    cur = conn.cursor()

    actions= gl.SFrame.from_sql(conn, "SELECT * FROM movieapp_ratings")

    training_data, validation_data = gl.recommender.util.random_split_by_user(actions, 'userid', 'movieid')
    model = gl.recommender.create(training_data, 'userid', 'movieid')
    results = model.recommend([int(userid)])
    model.save("my_model")
    return results
def loadmodel(userid):
    model = gl.load_model("my_model")
    results = model.recommend([int(userid)])
    return results

```

Figure 5.5: Recommendation code

5.7 Result

The accuracy or cross validated r-squared value of the prediction model was 0.7555. Here the accuracy rate can be represented as 75.55

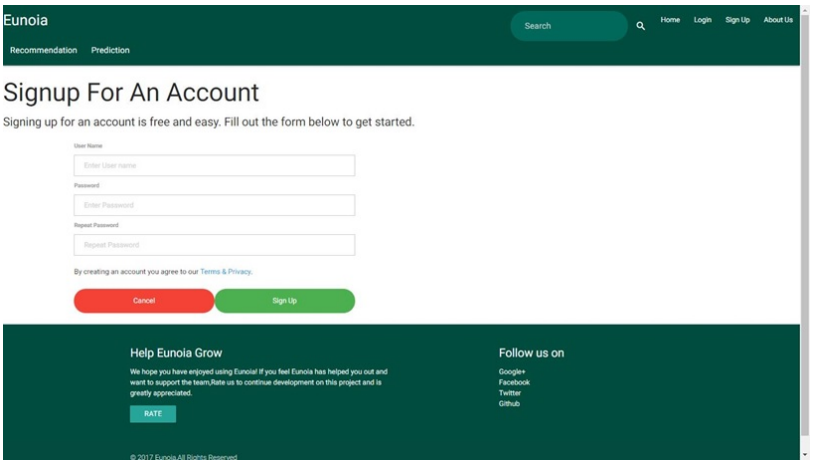

Figure 5.6: Signup Form

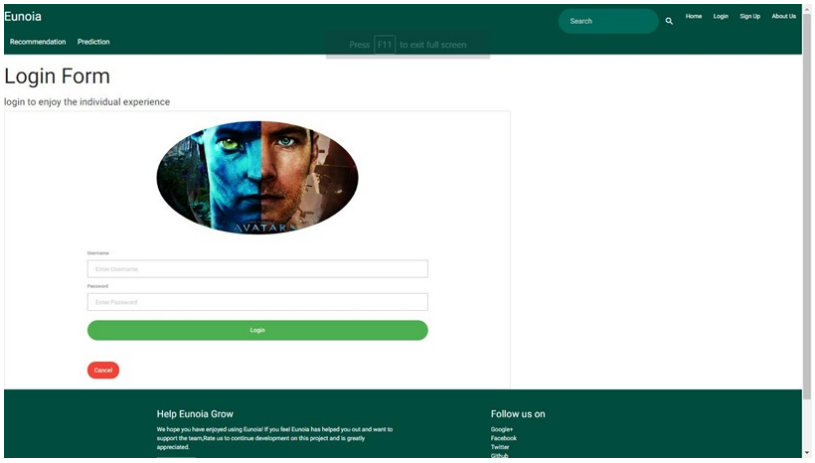

Figure 5.7: Login Form

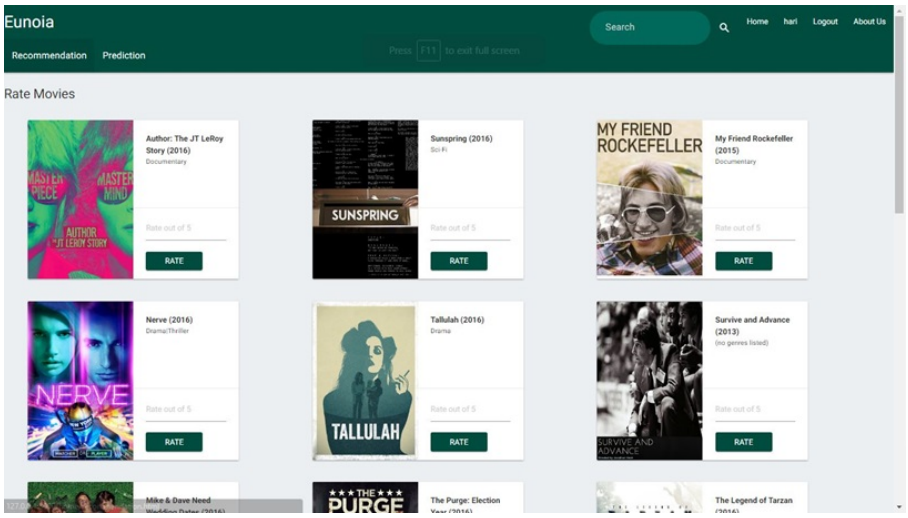

Figure 5.8: Recommendation rating

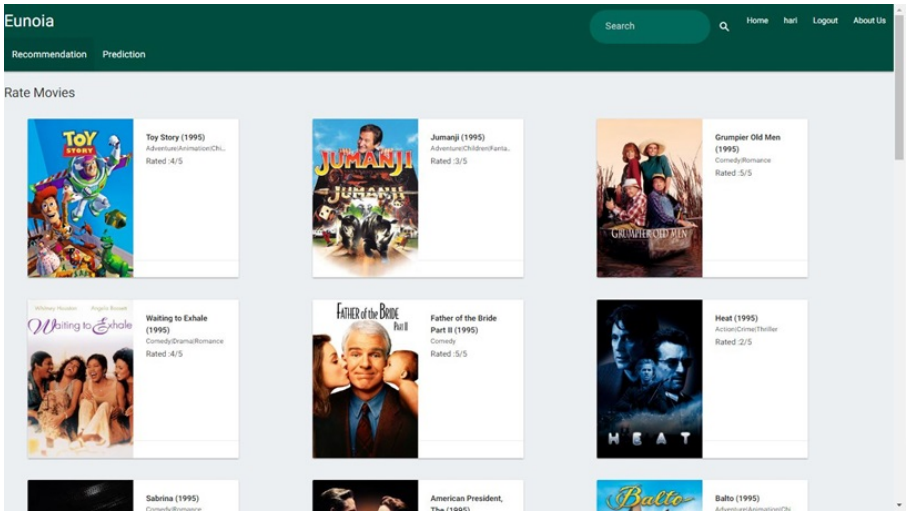

Figure 5.9: Rated movies

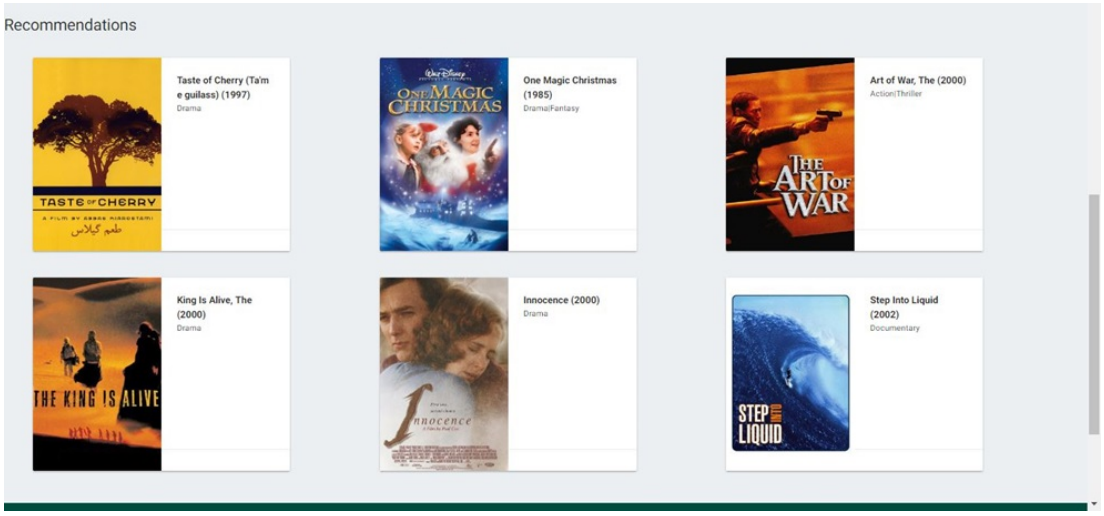

Figure 5.10: Recommendations

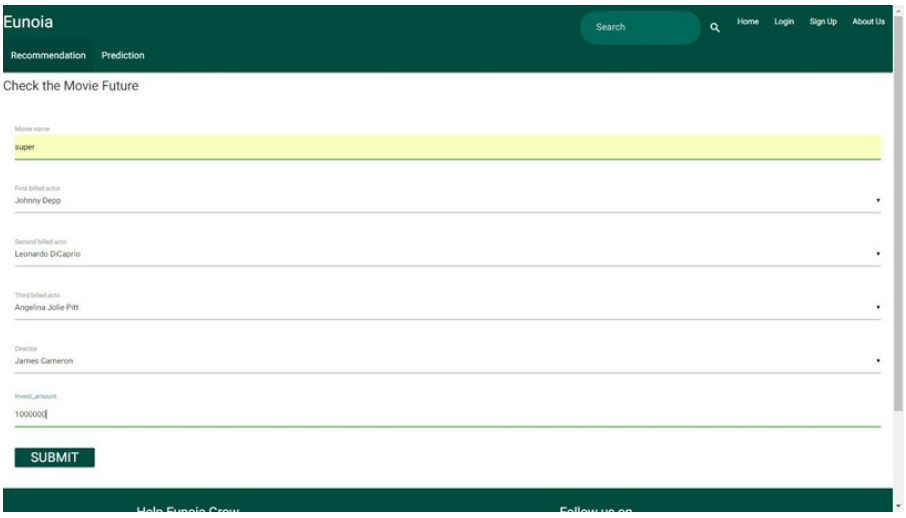

Figure 5.11: Prediction form

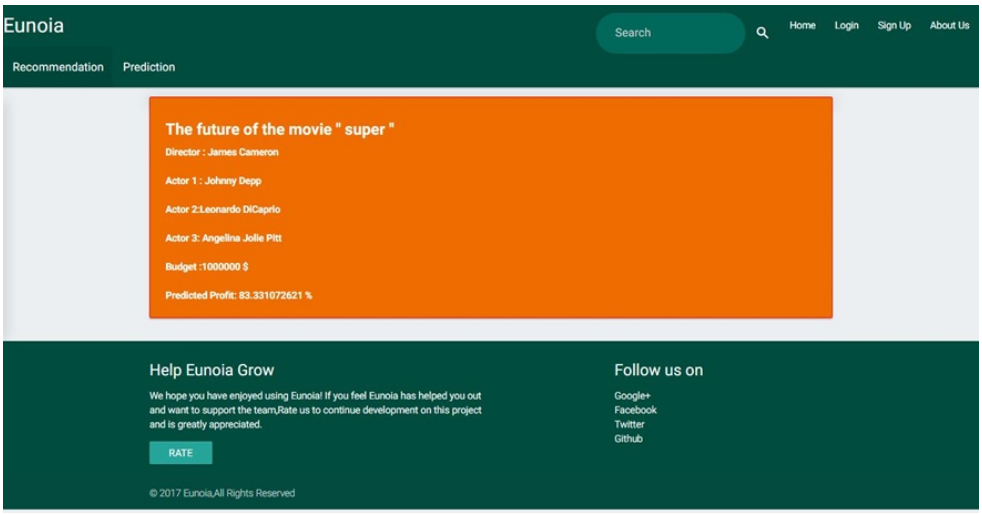

Figure 5.12: Prediction result

Chapter 6

CONCLUSION AND FUTURE WORKS

The proposed system is a decision support system to aid invester's decision on which movies to invest in.The system learns from freely available historical data from various sources and tries to predict the success of movies.

Also the users of the system will be able to get the recommendation based on the taste of the user. The proposed system concentrates only on the hollywood movies.

Reference

[1] *Michael T. Lash and Kang Zhao .Early Predictions of Movie Success: the Who,What, and When of Profitability, 2016.*

[2] *K. R. Apala, M. Jose, S. Motnam, C.-C. Chan, K. J. Liszka, and F. de Gregorio.Prediction of Movies Box Office Performance Using Social Media. In Proc. of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 13, pages 12091214, 2013.*

[3] *S. Asur and B. A. Huberman. Predicting the Future With Social Media. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pages 492499, 2010.*

[4] *D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:9931022, 2003.*

[5] *P. Boccardelli, F. Brunetta, and F. Vicentini. What is Critical to Success in the Movie Industry? A Study on Key Success Factors in the Italian Motion Picture Industry. Dynamics of Institutions and Markets in Europe (DIME), 46(46), 2008.*

[6] *Kasra Madadipouya.A Location based movie recommender syatem using collaborative filtering, 2016.*

[7] *Pasquale Lops, Marco de Gemmis and Giovanni Semeraro, Content-based Recommender Systems: State of the Art and Trends, Springer Science+Business Media, LLC 2011*

[8] *Akira Ishii, Hisashi Arakaki, Naoya Matsuda, Sanae Umemura, Tamiko Urushidani, Naoya Yamagata and Narihiko Yoshda, Mathematical model*

for hit phenomena as stochastic process of interactions of human interactions, Department of Applied Mathematics and Physics, Tottori University Koyama, Tottori 680-8552, Japan

[9] *Dan Cocuzzo, Stephen Wu,Hit or Flop: Box Office Prediction for Feature Films, December 13, 2013.*

[10] *Ajay Prasadh Viswanathan, Movie Success Predictor, Department Of Computer Science and Engineering,National Institute Of Techonology Thuvakudi,Trichy, India.*

Department of Computer Science & Engineering
Vidya Academy of Science & Technology
Thalakkottukara, Thrissur - 680 501
(http://www.vidyaacademy.ac.in)