INTRODUCTION

A recommendation system is a type of information filtering system which attempts to predict the preferences of a user, and make suggestions based on these preferences.

A movie recommendation is important in our social life due to its strength in providing enhanced entertainment. Such a system can suggest a set of movies to users based on their interest, or the popularities of the movies. Although, a set of movie recommendation systems have been proposed, most of these either cannot recommend a movie to the existing users efficiently or to a new user by any means. In this paper we propose a movie recommendation system that has the ability to recommend movies to a new user as well as the others. It mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. It generates movie swarms not only convenient for movie producer to plan a new movie but also useful for movie recommendation. Experimental studies on the real data reveal the efficiency and effectiveness of the proposed system.

Business Problem

In the era of information overload, it is very difficult for users to get information that they are really interested in. And for the content provider, it is also very hard for them to make their content stand out from the crowd. That is why many re-searchers and companies develop Recommender System to solve the contradiction. The mission of Recommender System is to connect users and information, which in one way helps users to find information valuable to them and in another way push the information to specific users. This is the win-win situation for both customers and content providers.

For building a recommender system from scratch, we face several different problems. Currently there are a lot of recommender systems based on the user information, so what should we do if the website has not gotten enough users. After that, we will solve the representation of a movie, which is how a system can understand a movie. That is the precondition for comparing similarity between two movies. Movie features such as genre, actor and director is a way that can categorize movies. But for each feature of the movie, there should be different weight for them and each of them plays a different role for recommendation. So we get these questions:

•How to recommend movies when there are no user information.

•What kind of movie features can be used for the recommender system.

•How to calculate the similarity between two movies.

•Is it possible to set weight for each feature.

DATA

The dataset used in the project was downloaded from movielens which contains data of 9000 movies and ratings of 600 people. The data is in csv (comma separated value) format

METHODOLOGY

Our goal is to find a new way to improve the classification of movies, which is the requirement of improving content-based recommender systems. n order to achieve the goal of the project, the first process is to do enough back-ground study, so the literature study will be conducted. The whole project is based on a big amount of movie data so that we choose quantitative research method. For philosophical assumption, positivism is selected because the project is experimental and testing character. The research approach is deductive approach as the improvement of our research will be tested by deducing and testing a theory. Expost facto research is our research strategy, the movie data is already collected and we don't change the independent variables. We use experiments to collect movie data. Computational mathematics is used data analysis because the result is based on improvement of algorithm. For the quality assurance, we have a detail explanation of algorithm to ensure test validity. The similar results will be generated when we run the same data multiple times, which is for reliability. We ensure the same data leading to same result by different researchers.

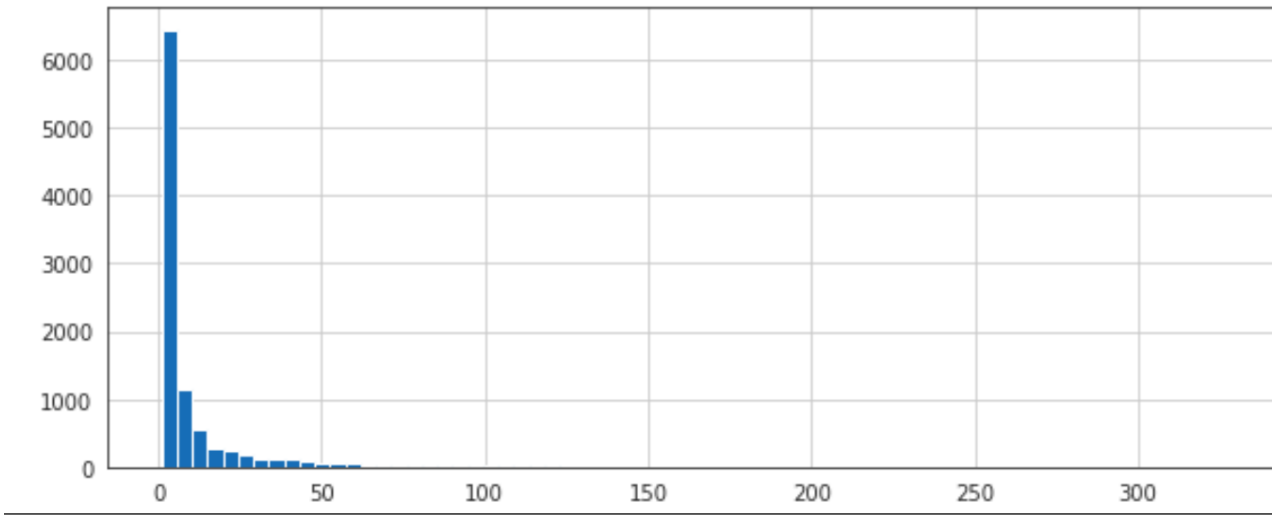
The methodology in this project consists of two parts:

- Exploratory Data Analysis: Visualise the number ratings given by people and the number of ratings to ratings given by the people using histograms and jointplot which helps for visualising the spread of ratings
- Modelling: Content based recommender system in machine learning is used to find similar movies based on the movie watched by a person based on what peoples ratings

Exploratory Data Analysis

```
plt.figure(figsize=(10,4))
ratings['no of ratings'].hist(bins=70)
```

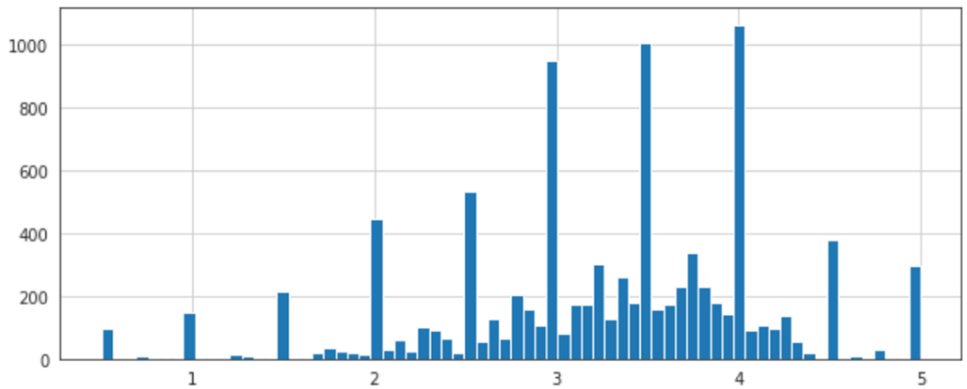
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe67125ecd0>
```



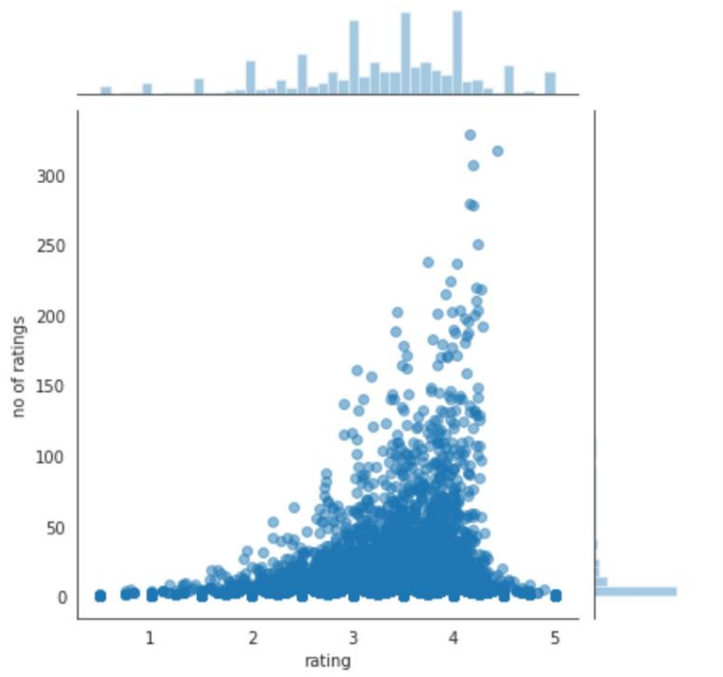
The above figure shows the relation between number of ratings given to number of movies

```
In [13]: plt.figure(figsize=(10,4))
         ratings['rating'].hist(bins=70)
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe671436040>
```



The above figure shows the relation between the different ratings given to number of movies



The above figure shows the relation between number of ratings with ratings using jointplot

MODELING

Content-based recommendation is an important approach in recommender systems. The basic idea is to recommend items that are similar with what user liked before. The core mission of content-based recommender system is to calculate the similarity between items. There are a lot of methods to model item and the most famous one is Vector Space Model. The model extracts keywords of the item and calculate the weight by TF-IDF. For example, set ki as the ith keyword of item dj, wij is the weight of ki for dj, then the content of dj can be defined as:

$$Content(dj) = \{w1j, w2j, ...\}$$

As we talked before, content-based recommender system recommends items that are similar with what user liked before. So the tastes of a user can be modelled according to the history of what the user liked. Consider ContentBasedProfile(u) as the preference vector of user u, the definition is:

$$ContentBasedProfile(u) = 1/|N(u)| sum d \in N(u) Content(d)$$

N(u) is what the user u liked before. After calculating content vector Content(.) and content preference vector ContentBasedProfile(.) of all users, given any user u and

an item d, how the user like the item is defined as the similarity between ContentBasedProfile(u) and Content(d):

$$p(u,d) = sim(ContentBasedProfile(u), Content(d))$$

Using keywords to model item is an important step for many recommender systems. But extracting keywords of an item is also a difficult problem, especially in media field, because it is very hard to extract text keywords from a video. For solving this kind of problem, there are two main ways. One is letting experts tag the items and another one is letting users tag them.

RESULT

The machine could identify and recommend movies based on other peoples rating using content based machine learning.

Conclusion

Recommender system has become more and more important because of the information overload. For content-based recommender system specifically, we attempt to find a new way to improve the accuracy of the representative of the movie. For the problems we mentioned at beginning, firstly, we use content-based recommender algorithm which means there is no cold start problem. In Section 4.1, we list all the features in our recommender system. Some of them are from other research team in the company, so the features are diversity and more accurate than others. Then we introduced the cosine similarity which is commonly used in industry. For the weight of features, we introduced TF-IIDF-DC which improve the representative of the movie.