[Machine Learning for Social Media Site for Visual Programming]

Project Proposal

Information Technology Capstone Project

COMP5703/5707/5708

Group Members

1. Wenjing Deng (470349281)
2. Hefei Chen (470488188)
3. Kaijia Chen (480218353)
4. Wentao Wu (480416845)
5. Zaiwo Liu (490191684)
6. Zhou Sha (480322650)

ABSTRACT

The proposal report is mainly introducing a social media site which aims to offer a platform for programmers to share projects and exchange their ideas. A recommendation system is contained which recommends some similar projects for users. It helps them find projects quickly based on their interests. There are two parts of our project, one is web development, the other is the recommendation system. The background introduction and relevant information of our project will be involved in first. Then we define the project plan which details project objectives, project scope, and potential problems. The methodologies and resources needed for our project will be discussed in the next part. We will use the item-based collaborative filtering model in our recommendation system. Also, the expected outcomes and the planned timeline will be mentioned at last.

TABLE OF CONTENTS

1. INTRODUCTION

We aim at building a brilliant platform that allows veterans to use their imagination and creativity while allowing entry-level programming enthusiasts to export their ideas. On this platform, you can upload your codes or join different groups for programming communication. In addition, our website hopes to establish a link between programmers and programmers, the so-called recommendation system. Once you upload your project, you will find people in other corners of the world who have the same vision as you. By referring to or tracking similar projects of other users, you can have a clearer understanding of the performance of the project. At the same time, you can also share your own projects on this page, so that your creativity is not just in the world of your own, but into the treasure of others. Our website has the information sharing and communication system among users that is lacking in the current mainstream code exchange website. In here, you not only have many learning resources that can be used as a reference, but can also publish your own design ideas on the Internet and discuss with the global creators.

2. RELATED LITERATURE

2.1 Collaborative Filtering

Collaborative filtering has significant advantages over traditional content-based filtering. Mainly because the machine will make mistakes during the scanning of keywords when processing content, and collaborative filtering is not based on this. In the essay Explaining Collaborative Filtering Recommendations, it mentions that CF can filter a more comprehensive form of data, including text, music and fund trends. Moreover, there are many things that cannot be expressed in words, such as taste or quality, can be expressed intuitively through people's favorite level, which greatly enhances the superiority of collaborative filtering (Herlocker, 2001).

The CF algorithm can be divided into two steps (Paolo & Paolo, 2004). The first step is the similarity assessment, which involves comparing the ratings provided by pairs of users (rows in the matrix) to calculate their similarity. Usually, we treat Pearson correlation coefficient as the most effective similarity assessment technique. The second step is the actual rating forecast, which includes predicting the rating of the active user's activity. The predicted rating is a weighted sum of ratings given by other users to the item, where the weight is the similarity rate of the active user to other users. In this way, ratings expressed by very similar users have a greater impact on the rating predicted by active users. The formula for the second step is as follows:

$$p_{a,i} = \overline{r}_a + \frac{\sum_{u=1}^{k} w_{a,u(r_{u,i} - \bar{r}_u)}}{\sum_{u=1}^{k} w_{a,u}}$$

From the formula, represents the predicted rating that active user a may provide for project i, is the average of the ratings provided by user u, w is the user similarity weight of a and u calculated in the first step and k is the number of users who consider the rating of item i in the weighted sum (called neighbors).

2.2 User-based CF and item-based CF

User-based CF is a simple algorithm that predicts what the current user will like by finding other users who behave like the behavior of the current user and using their ratings on other items to do the recommendation. In the thesis Evaluating collaborative filtering recommender systems, the author gave such an example. Now judge whether to recommend product A to Mary. It is known that Mary does not rate for product A, but gives products B, C and D different rates. User-based CF searches for similar users based on Mary's rates for B, C, D and collects their rates for A products. These users' ratings for the product A are weighted by their level of agreement with Mary's ratings in order to predict Mary's preference (Terveen, Herlocker, & Konstan, 2004).

But there are still problems in user-based CF. The major problem is the scalability of the collaborative filtering algorithms. These algorithms may have good performance when searching for thousands of potential neighbors in real-time, but when it comes to meet the demand for modern systems is to search for potential neighbors in the tens of millions of levels, it has poor performance (Sarwar, Karypis, Konstan, & Riedl, 2001). To solve those problems, we introduce the item-based CF.

Item-based CF uses the similarity between the rating patterns of the projects to make recommendations. If two projects tend to make the same users like and dislike them, then they are similar and users are expected to have similar preferences for similar projects. Thus, in its overall structure, the approach is similar to earlier content-based recommendations and personalization methods, but project similarity is derived from user preference patterns rather than from project data. In the original form, project-based CF does not solve any problems: finding the most similar project to generate predictions and recommendations is still necessary. In systems with more users than projects, it allows neighborhood discovery in the smaller of the two dimensions and it provides a major performance boost by well pre-computing the similarity matrix (Terveen, Herlocker, & Konstan, 2004).

2.3 Deep Collaborative Filtering

Recommender system ranges from content-based recommender system to collaborative filtering recommendations, while collaborative filtering extends from basic user-based collaborative filtering, item-based collaborative filtering, to model-based collaborative filtering. Perhaps deep learning is not as good as the image processing algorithm in the recommendation system, but deep learning does play a contributory effect. DCF is able to extract features directly from the content, and it is easy to process noise data. In addition, we can use RNN cyclic neural networks to model dynamic or sequence data, which to a high degree solves the cold-start problem caused by newly registered users (Jian, Jianhua, Kai, Yi, & Zuoyin, 2016).

The flow chart below shows the Deep Crossing model which is proposed by Microsoft. By adding the embedding layer, it is possible to transform the sparse features into low-dimensional dense features. Then use the concat layer to connect the segmented feature vectors. The next step is to complete the feature combination and transformation through the multi-layer neural network, and finally calculate the CTR rate through the scoring layer (Ying, T. Ryan, Jian, Haijing, & Yu, 2016). Unlike the classic DNN, the multilayer perceptron used by Deep crossing is composed of a residual network.

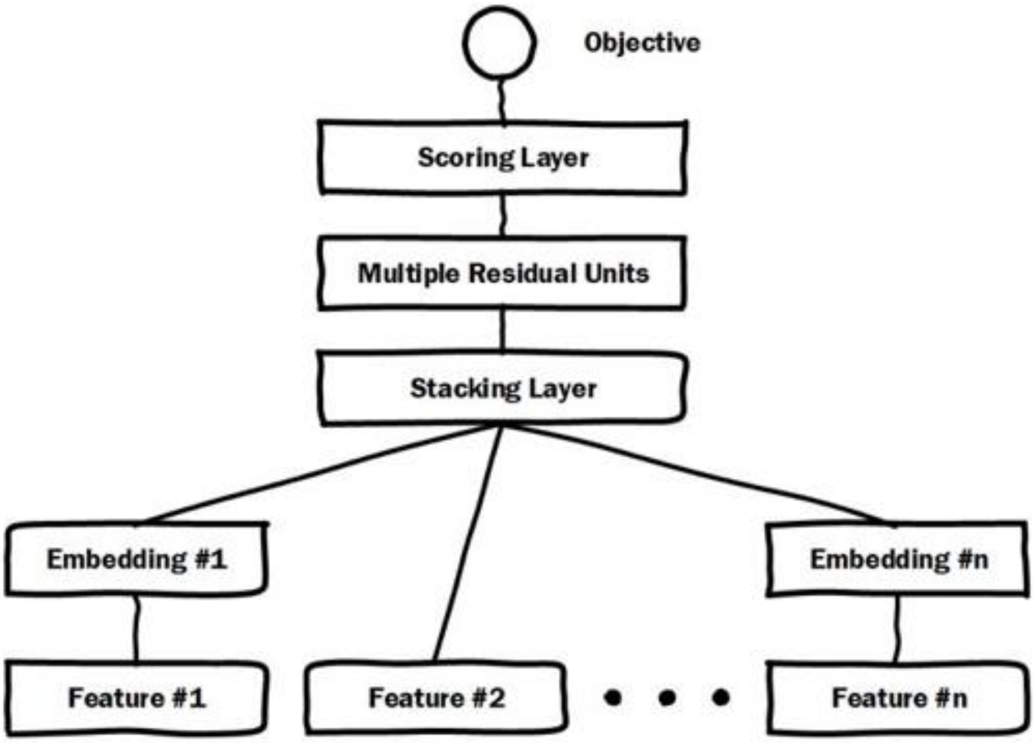


Figure 1: Embedding solution of DCF

FNN is another method to do the embedding initialization. The flow chart is shown below (Weinan, Tianming, & Jun, 2016):

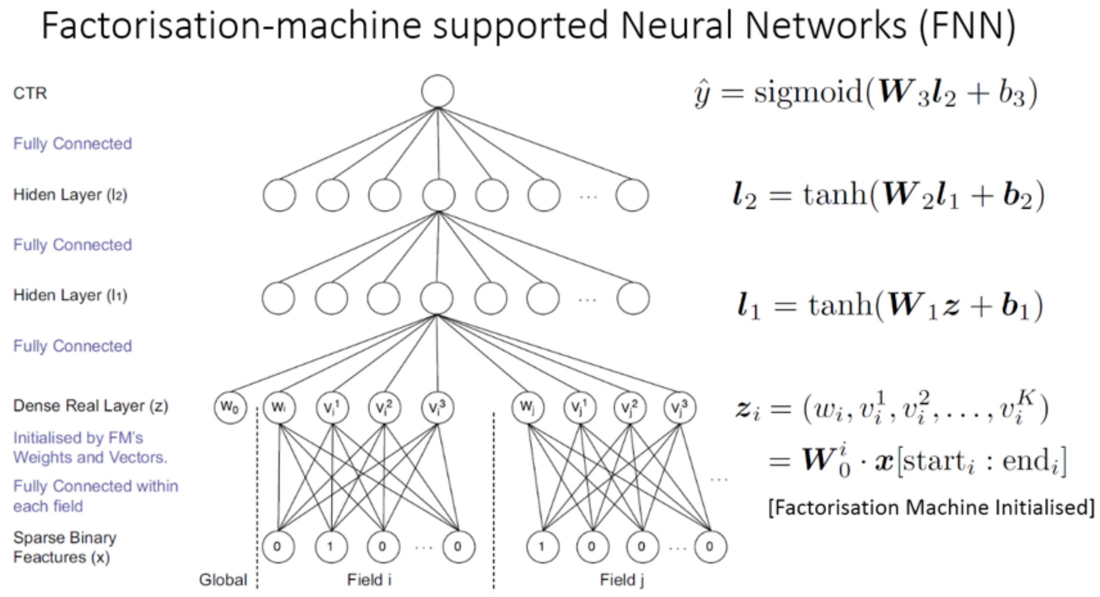

Factorisation-machine supported Neural Networks (FNN)
CTR
Fully Connected
Hiden Layer (l2)
Fully Connected
Hiden Layer (l1)
Fully Connected
Dense Real Layer (z)
Initialised by FM's Weights and Vectors.
Fully Connected within each field
Sparse Binary Feactures (x)
Global
Field i
Field j
$\hat{y} = \mathrm{sigmoid}(\boldsymbol{W}_3\boldsymbol{l}_2 + b_3)$
$\boldsymbol{l}_2 = \tanh(\boldsymbol{W}_2\boldsymbol{l}_1 + \boldsymbol{b}_2)$
$\boldsymbol{l}_1 = \tanh(\boldsymbol{W}_1\boldsymbol{z} + \boldsymbol{b}_1)$
$\boldsymbol{z}_i = (w_i, v_i^1, v_i^2, \ldots, v_i^K)$
$= \boldsymbol{W}_0^i \cdot \boldsymbol{x}[\mathrm{start}_i : \mathrm{end}_i]$
[Factorisation Machine Initialised]

Figure 2: Basic algorithm and flow chart for FNN

Actually, we know that the features x of the dataset used in recommender system is definitely discrete and sparse. So in FNN we just divide x into N fields. For each field, there's only one value rate 1 and the rest of those we rate it as 0. So the field i is calculated as shown: $x[start_i: end_i]$. W_0^i is the matrix for embedding and then put the result into the full-connected neural network.

2.4 Optimization Measures for DCF

In traditional deep collaborative filtering, we use kernel function to ascend the dimension in order to make it easier to do the clustering part. In the essay Kernelized Synaptic Weight Matrices, it introduces an optimization method. Instead of using the server kernel function, it creates a kernel layer and treats it as the active function. From the flowchart below we can see for each different part of the data set, we arrange a different kernel to execute the encoding part (Muller, N. P. Martel, & Indiveri, 2018).

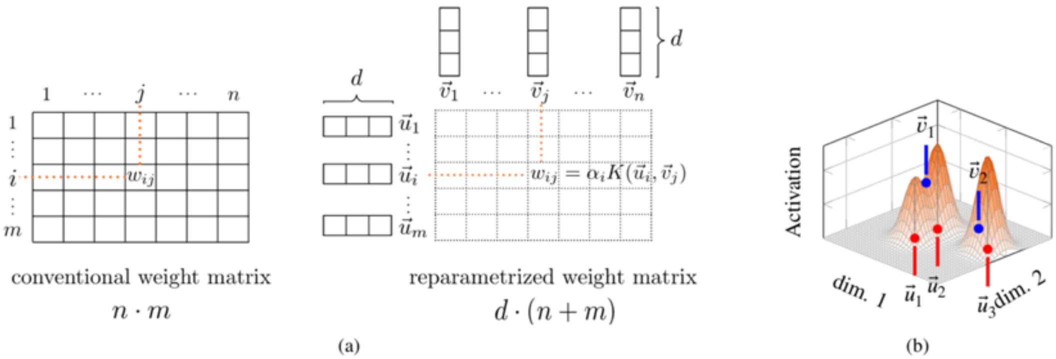


Figure 3: Building a kernel layer to improve the DCF

2.5 Baseline Algorithm for Recommendation System

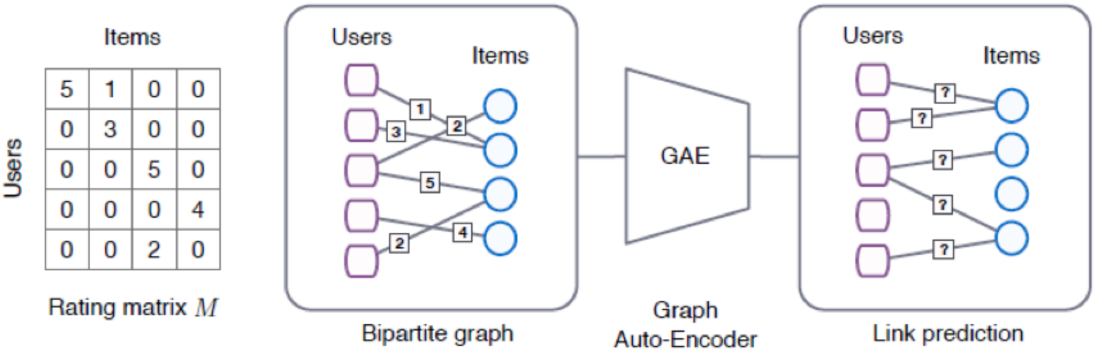
The kernel layer it creates can assign accurate and based on a scan of the original dataset. For the baseline algorithm, The thesis On the Difficulty of Evaluating Baselines: A Study on Recommender System (Rendle, 2019) gives a brief introduction. Different from the collaborative filtering, the idea of the baseline algorithm is to set up a baseline and put user bias and film bias into the equation. The formula for this algorithm is as follows:

$$bui = \mu + bu + bi$$

In the above formula, the result of the equation is to give an estimate of user1 for item1 in the prepared baseline model. μ is the average of all users rating the movie; bu is the user bias (bu will be negative if the user asks for a higher and the score is relatively low; conversely, if the user frequently corrects many item scores, then bu is positive); bi It is the project deviation that reflects the popularity of the project. (Berg, Kipf, & Welling, 2017)According to the paper, if user1 is unknown, the deviation bu is assumed to be zero. The same applies to item1. Although the Baseline model is simple, it actually contains personalized information about users and projects. In the face of large-scale data, simple algorithms can reduce a lot of computing time.

2.6 Graph auto-encoders

Using graph convolutional matrix to do the encoding and decoding part is another way to fetch the accuracy of the recommender system according to the essay Graph Convolutional Matrix Completion (Berg, Kipf, & Welling, 2017). The flowchart for theory is shown below:



5	1	0	0
0	3	0	0
0	0	5	0
0	0	0	4
0	0	2	0

Figure 4: Flowchart for graph convolutional matrix

The matrix on the left side of the above picture is called the rating matrix M, which contains all users' scores on the product. When the user does not score the product, the value is automatically set to zero. On the right is a user-item interaction diagram with a binary structure. The edge corresponds to an interaction event, and the number on the edge donate the rating the user gave to a particular item. The matrix completion task can be converted to a link prediction problem and modeled using an end-to-end trainable graphical autoencoder.

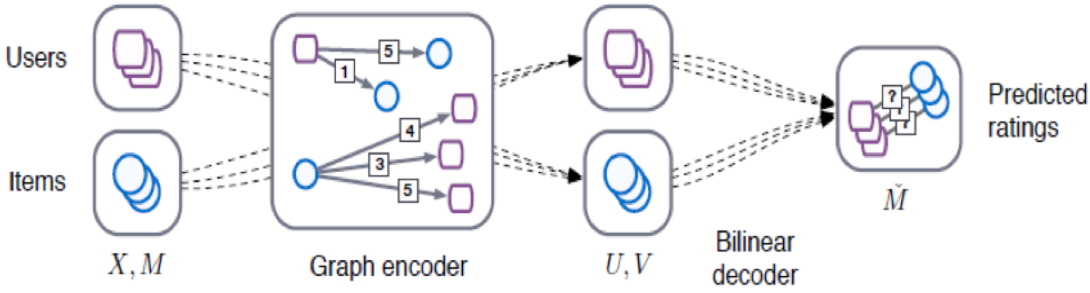


Figure 5: Embedding solution for graph auto-encoder

For the decoding part, the essay uses bilinear decoder to achieve the accuracy. The formula is shown below:

$$p(\check{M}_{ij} = |r\ ) = \frac{e^{u_i^T Q_r v_j}}{\sum_{s \in R} e^{u_i^T Q_s v_j}}$$

3. RESEARCH/PROJECT PROBLEMS

3.1 Research/Project Aims & Objectives

By analyzing and summarizing the project description and the results of multiple meetings with the client, the project goal can be interpreted as designing and implementing a social media website containing a project recommendation system, which allows users to learn, share and communicate projects and project-related content. Specifically, this objective can be divided into two parts. One is to design and implement a fully functional social media website, this part is mainly about software development. The other part is to design a project recommendation system and provide the corresponding API to facilitate website to invoke.

3.2 Research/Project Questions

In order to achieve the project goals and objectives, we need to clarify the requirements of the project and the problems that the client wants to solve. After discussion at the meetings, there are a few issues to be aware of.

First, the entire project is divided into two main sections, social media sites, and recommendation systems. Among them, the development of social media sites is divided into front-end design and back-end design. Therefore, ensuring synergy and consistency between these three components is a top priority for this project. Thereby, it is necessary to clarify the project's schedule, deliverable and other vital factors.

Second, there should be a clear definition of the needs of social media websites, such as the features and function, the target customers, as well as the programming language and framework of the website.

Third, for the recommendation system, it is important to define the algorithms and models for developing the recommendation system, and the data set which needs to be applied. In this project, the client requires a collaborative filtering-based recommendation model with Netflix Movie as a training and test data set for building the recommendation system.

3.3 Research/Project Scope

3.3.1 Project Scope Statement

This project scope statement serves as a baseline document to confirm the scope of the project (CS-26) and to ensure that all stakeholders have an understanding and consensus of the scope of the project. The scope of this version is stated on September 5, 2019.

3.3.2 Scope Description(Characteristics and Requirements)

The scope of the project is a social media site designed, built and deployed for students, teachers, educators and interested enthusiasts. The website mainly includes related functions of social media websites, such as user registration and login, project upload/download, comment and message system, and project information system. At the same time, the website also contains a recommendation system to help users quickly find the required items and a database for recording all relevant information.

Implementing social media functions are a fundamental requirement of the entire project. The user registration and login system is a basic function of the entire social networking site and effectively supports other features. For example, the user can upload and download the project only after logging in.

Project upload/download is one of the core functions of the website. It is convenient for users to share their own projects and learn other people's projects.

The comment and message function can satisfy the communication and the discussion of the project between users.

The project information system is mainly displayed on the specific page to show the information related to the project, including the project profile, core code, download link, message area and author information.

The recommendation system is an important objective of the project. Based on the relevant collaborative filtering recommendation model, the user is provided with items related to the browsed project, which is convenient for the user to locate and query the required project information.

The database is to store all relevant information, including user information, project information, history, recommendation information and other project-related information.

4. METHODOLOGIES

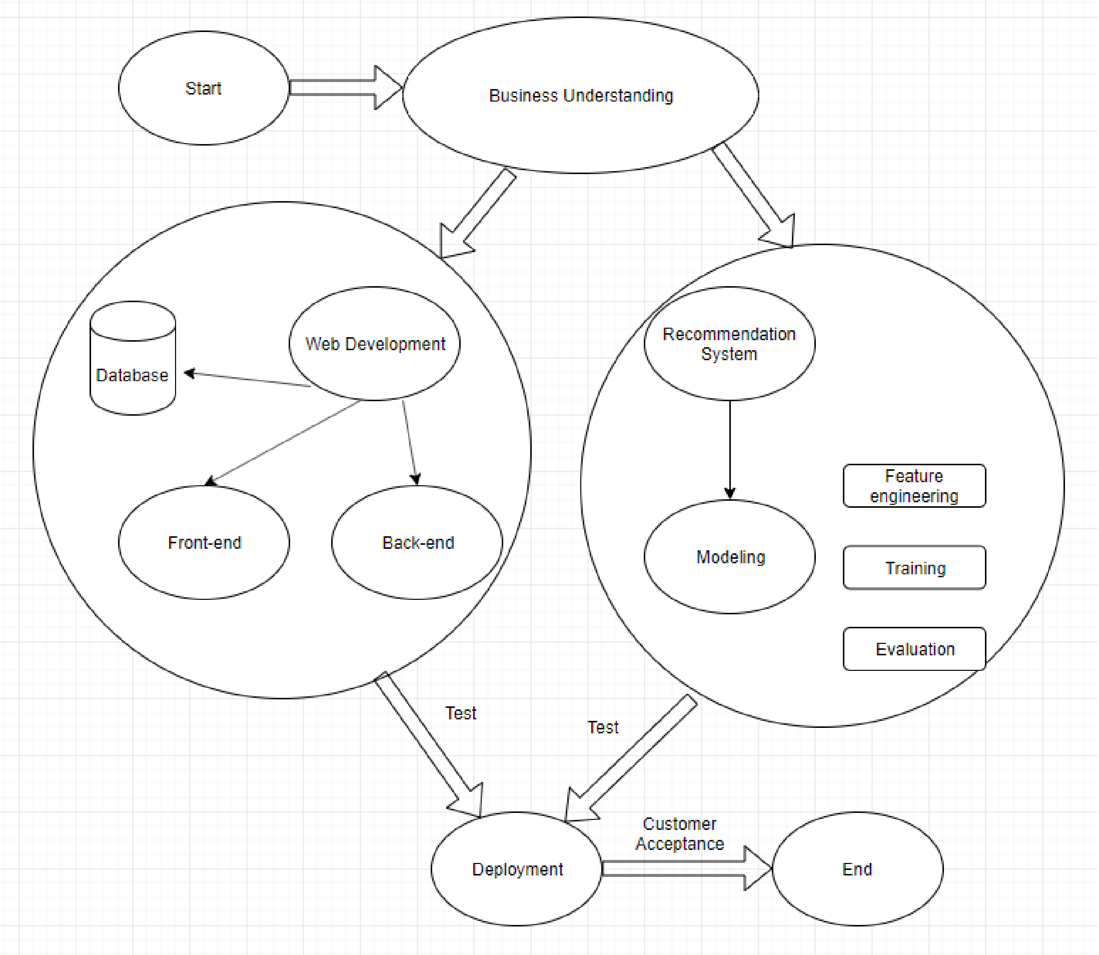


Figure 6: Project life cycle

The figure above shows our project life cycle. The whole project is separated into 2 parts, web development and recommendation system. Our group will focus on the recommendation system part. So we will use collaborative filtering model to extract features of data, and train it, then evaluate the quality of the model. After some tests, If the result is good, we will create an API to the back-end team do call it. The result could show on the front-end page so that help users receive more information about their interesting project.

4.1 Methods

Describe the methods you will use to solve the problem you are addressing.

For web development, we will use AJAoX, etc. For the recommendation system, we will use ranking-critic methods to train a collaborative filtering model.

Item-based collaborative filtering will be used as a basic model in the recommendation system. Due to the large scale data, we will use the amortized ranking-critical training method to improve this system's effectiveness. It is a new method that modified by Sam Lobel, etc. which was published on 10 June, 2019. This method contains actor and critic parts. So the first stage we will do is create this improved model. The idea is borrowed from reinforcement learning and scalable learning-to-rank algorithm. The critic is to approximate ranking metric, while the actor is to optimise for this metric (Sam Lobel, Chunyuan Li, Jianfeng Gao & Lawrence Carin, 2019).

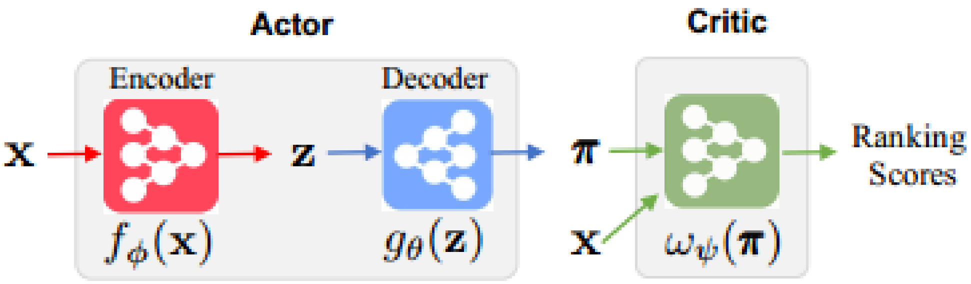


Figure 7: Actor-critic method

4.2 Data Collection

Due to the lack of project data resource, we use Netflix movie data from Kaggle instead. Our recommendation system will run on this dataset to test the model performance.

4.3 Data Analysis

We will use python to analyse the data. There are four files, and each file contains 4 columns:

1. Movie ID

2. Customer ID

3. Rating (1 to 5)

4. Date they gave the ratings

There is another file contains the mapping of Movie ID to the movie background like name, year of release, etc ("Netflix - Movie recommendation | Kaggle", 2019).

Because some parts of the dataset are so messy, so we need to do some data cleaning. And we will slice the data due to its huge data. After the preprocessing stage, we have a significant improvement in efficiency.

4.4 Deployment

When we complete the recommendation system, we will provide an API to the back-end team of the web so that they can call our recommendation system directly. Processing will take place on AWS. Then we will update our system based on their feedback. Deploying our website and recommendation system on a client's computer requires Windows or Mac OS with a resolution greater than 720P and access to the Internet.

4.5 Testing

Before the project is completed, our team will start the testing part. We will divide the testing process into the following steps:

1. Unit testing.

Unit testing is the first stage of the testing. It is the most basic part of the testing, but also one of the most important parts to ensure that all components of our recommendation system can work properly. Finding problems earlier means that it takes less time to fix them.

2. Integration testing.

After unit testing, we will conduct integration testing. It can check whether the interface between recommendation system and the web is correct, and ensure that our recommendation system and web can work completely.

3. System testing.

System testing will take the integrated web and recommendation system as a part of the computer system, combining with other parts of the system, and test the computer system strictly in the actual operating environment to find potential problems.

4. Regression testing.

Regression testing is to retest when defects are found and modified. It is used to check whether the defects found have been corrected, and the changes made do not cause new problems.

5. Performance testing.

Performance testing is a non-functional test used to determine the behavior of our recommendation system and website under various conditions. We will carry out the load test, stress test, endurance test and etc. Ultimately, we need to keep our website and recommendation system in a reasonable state of operation.

6. Compatibility testing.

Compatibility testing will measure how our website and recommendation systems work in different environments. It checks whether the site is compatible with different operating systems, browsers or resolutions. It will ensure that our website and recommendation system functions are consistently performed in any environment that users use.

5. RESOURCES

5.1 Hardware & Software

The website will run on the server available for all major browsers. To enable the running of the website, it requires design tools for the UI and programming tools for the API, it also needs database management tools for backend management. As for project management, some apps such as Slack, Github, Wechat, Google Docs, Trello are used. Since our group is working on the recommendation system design using machine learning algorithms, we will use Python as the programming language and use some packages like Pandas/Keras for the model design.

5.2 Materials

The related papers are needed to initiate our design, and open-source data and sample code will be used, for example, we will use the MovieLens dataset on Kaggle.com for our model training.

5.3 Roles & Responsibilities

Our team has several roles like project manager, project team member, computer programmer, software tester and technical writer. Each of us in our team is having similar roles by fulfilling all the responsibilities of these roles. For example, when we have a meeting, each of us will write down things we find important, and exchange what we have right after the meeting. We will also all be involved with model design and model test. At the end of this project, we will all write instructions on the model we used for the recommendation system of the website.

6. EXPECTED OUTCOMES

In this project, we are expected to build a platform for the IoT related visual programmers to exchange their ideas and share their projects. The sites will have a recommendation system for users to find their interested projects.

6.1 Project Deliverables

A social media site with many functions such as share, comment, search with interests, and recommend projects to the user. An instruction of this website.

6.2 Implications

IoT is a trending area in the whole world, which can affect the future generation and simplify the ways of living. However, its cores and foundations are still based on the Internet with programming skills. Out project can help the visual programmers in this area to find a cosy place to communication, and a professional and easier way to exchange ideas and projects. We believe our platform can accelerate the development of IoT related programming.

7. MILESTONES / SCHEDULE

7.1 Breakdown of tasks

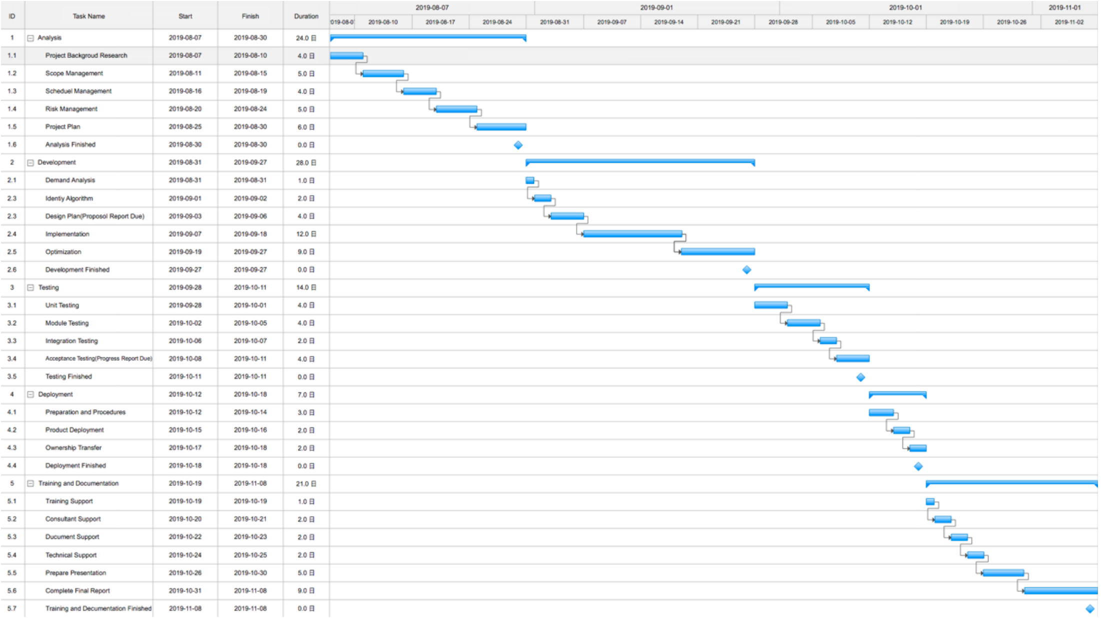
We divided our project into five main stages which are Analysis, Development, Testing, Deployment, Training and documentation. The table below outlined the milestones of the project weekly.

1 Milestone	Tasks	Reporting	Date
Week-1	Understand the project & research background knowledge	None	2019-08-09
Week-2	Meet with our client and define project plan	Client meeting to review the project	2019-08-16
Week-3	Identify the requirements of the project clearly & Analyze two websites	Client meeting to identify the requirement about the project	2019-08-23
Week-4	Focus on the recommendation part & Do literature review	Client meeting to compare the difference between two websites	2019-08-30
Week-5	Proposal Report Due & Define recommendation system algorithm	Client meeting to understand the details of papers	2019-09-06
Week-6	Complete design plan	Client meeting to review the design plan	2019-09-13
Week-7	Implement the recommendation system	Client meeting to review Implementation	2019-09-20
Week-8	Optimization	Client meeting to review final performance	2019-09-27
Week-9	Testing & Progress Report Due	Client meeting to review test outcome	2019-10-11
Week-10	Deployment	Client meeting to deploy the system	2019-10-18
Week-11	Training & Documentation	Client meeting to review documents	2019-10-25
Week-12	Final Presentation	None	2019-11-01
Week-13	Final Report (thesis)	None	2019-11-08

Table 1: The milestones of the project weekly

7.2 Timeline

We develop an overview of the project schedule by Gantt chart below. It explains the planned timeline and the breakdown of tasks in detail.



ID	Task Name	Start	Finish	Duration
1	⊟ Analysis	2019-08-07	2019-08-30	24.0 日
1.1	Project Backgroud Research	2019-08-07	2019-08-10	4.0 日
1.2	Scope Management	2019-08-11	2019-08-15	5.0 日
1.3	Scheduel Management	2019-08-16	2019-08-19	4.0 日
1.4	Risk Management	2019-08-20	2019-08-24	5.0 日
1.5	Project Plan	2019-08-25	2019-08-30	6.0 日
1.6	Analysis Finished	2019-08-30	2019-08-30	0.0 日
2	⊟ Development	2019-08-31	2019-09-27	28.0 日
2.1	Demand Analysis	2019-08-31	2019-08-31	1.0 日
2.3	Identiy Algorithm	2019-09-01	2019-09-02	2.0 日
2.3	Design Plan(Proposol Report Due)	2019-09-03	2019-09-06	4.0 日
2.4	Implementation	2019-09-07	2019-09-18	12.0 日
2.5	Optimization	2019-09-19	2019-09-27	9.0 日
2.6	Development Finished	2019-09-27	2019-09-27	0.0 日
3	⊟ Testing	2019-09-28	2019-10-11	14.0 日
3.1	Unit Testing	2019-09-28	2019-10-01	4.0 日
3.2	Module Testing	2019-10-02	2019-10-05	4.0 日
3.3	Integration Testing	2019-10-06	2019-10-07	2.0 日
3.4	Acceptance Testing(Progress Report Due)	2019-10-08	2019-10-11	4.0 日
3.5	Testing Finished	2019-10-11	2019-10-11	0.0 日
4	⊟ Deployment	2019-10-12	2019-10-18	7.0 日
4.1	Preparation and Procedures	2019-10-12	2019-10-14	3.0 日
4.2	Product Deployment	2019-10-15	2019-10-16	2.0 日
4.3	Ownership Transfer	2019-10-17	2019-10-18	2.0 日
4.4	Deployment Finished	2019-10-18	2019-10-18	0.0 日
5	⊟ Training and Documentation	2019-10-19	2019-11-08	21.0 日
5.1	Training Support	2019-10-19	2019-10-19	1.0 日
5.2	Consultant Support	2019-10-20	2019-10-21	2.0 日
5.3	Ducument Support	2019-10-22	2019-10-23	2.0 日
5.4	Technical Support	2019-10-24	2019-10-25	2.0 日
5.5	Prepare Presentation	2019-10-26	2019-10-30	5.0 日
5.6	Complete Final Report	2019-10-31	2019-11-08	9.0 日
5.7	Training and Decumentation Finished	2019-11-08	2019-11-08	0.0 日

Figure 8: Gantt chart

REFERENCES

Berg, R., Kipf, T., & Welling, M. (2017). *Graph Convolutional Matrix Completion.*

Herlocker, J. &. (2001). Explaining Collaborative Filtering Recommendations. *Proceedings of the ACM Conference on Computer Supported Cooperative Work.* Minneapolis.

Jian, W., Jianhua, H., Kai, C., Yi, Z., & Zuoyin, T. (2016, 11 28). Collaborative Filtering and Deep Learning Based Recommendation System For Cold Start Items. *Expert Systems With Applications*, p. 8.

Muller, L. K., N. P. Martel, J., & Indiveri, G. (2018). Kernelized Synaptic Weight Matrices. *International Conference on Machine Learning.*

Netflix - Movie recommendation | Kaggle. (2019). Retrieved 6 September 2019, from https://www.kaggle.com/laowingkin/netflix-movie-recommendation#Data-manipulation

Paolo, M., & Paolo, A. (2004). Trust-aware Recommender Systems. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences* (pp. 492-508). Cyprus: Agia Napa.

Rendle, S. &. (2019, 5). *On the Difficulty of Evaluating Baselines: A Study on Recommender Systems.* Israel. Retrieved from https://www.researchgate.net/publication/332897835_On_the_Difficulty_of_Evaluating_Baselines_A_Study_on_Recommender_Systems/citation/download

Sam Lobel, Chunyuan Li, Jianfeng Gao & Lawrence Carin(2019). Towards Amortized Ranking-Critical Training for Collaborative Filtering.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. *Proceedings of ACM World Wide Web Conference*, (p. 3). Hong Kong.

Terveen, L. G., Herlocker, J. L., & Konstan, J. A. (2004, 1 22). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, pp. 5-33.

Weinan, Z., Tianming, D., & Jun, W. (2016). Deep Learning over Multi-field Categorical Data: A Case Study on User Response Prediction. In F. Nicola, & C. Fabio, *Advances in Information Retrieval. ECIR 2016. Lecture Notes in Computer Science* (pp. 45-57). Springer, Cham.

Ying, S., T. Ryan, H., Jian, J., Haijing, W., & Yu, D. (2016). Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features. *KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 255-256). California: ACM