

# Project State Report

## Movie Recommendation System

EEP596 Recommendation System

Group Member: Qingchuan Hou, Bo Wang

**Section 1: Introduction and related work, explain the problem you are trying to solve in the project and summarize the related works from the literature.**

We are trying to implement the Movies Recommendation Project. We will recommend the movies based on the multiple features of the movies. The recommendation system we try to build is not a personal recommendation system. It does not make recommendations based on the usage information of each single user. However, this model will give a general recommendation based on the similarity of movies. By entering a movie, we recommend other movies that are similar to that movie. Therefore, our dataset does not contain the exact ground truths but only the features of the movie. To solve this problem, we decided to use an unsupervised Machine Learning Model

In this project, the problem we need to solve:

- Data: What kind of useful Features we need, and how to preprocess the data to fit in the machine learning model.
- Model: What kind of unsupervised Machine Learning Model can be used for our recommendation system.

**Section 2: Method section, explain how you are modeling your problem as a machine learning model, including the architecture of your baseline model.**

First, we will need to use pandas to get the useful features from the Movies data that we get from multiple data sources and combine them as the Movies dataset, which will be used for training. The dataset includes show year, genres, average rating and number of votes. Then we normalize all the number features and count the string features in column genres and add it back to the dataframe.

The t-SNE we use is an unsupervised machine learning algorithm. The algorithm calculates the similarity between different movies by repeatedly calculating the distance between each movie vector point and other movie vector points in the multi-dimensional space. And sort the movies by the calculated similarity. Among them, the vector of the

movie is the features. Through the t-SNE algorithm, a sorted movie sequence will be generated, and movies with similar features will be adjacent in the sequence.

In the recommendation stage, by inputting the movie name, find the position of the input movie in the final generated sequence, and then recommend the adjacent movies in the sorted movie list.

### **Section 3: Experiment section, report the performance of your baseline model, and clearly explain the metrics you are using to measure the performance of your baseline model.**

Since our data have no ground truth, the model we are choosing here is unsupervised learning, which means we have no metrics that are able to evaluate the baseline model. However, we tried to input some of the examples and we checked based on our common sense, it worked great.

```
4... searching_movie = 'Avengers: Age of Ultron'
      tSNE_recommend(searching_movie, movies_sorted)
```

```
Recommend Movies:
Iron Man
Star Wars: Episode VII - The Force Awakens
Captain America: The Winter Soldier
The Hunger Games
Captain America: Civil War
X-Men: Days of Future Past
Iron Man 3
Edge of Tomorrow
Jurassic Park
```

### **Section 4: Next steps, discuss the possible design of the NN model that you implement next.**

We tried to use the Softmax Deep Neural Network Model to solve the problem, however while we tried to calculate the loss function, since the data we are using have no ground truth values and the machine learning model we use is unsupervised learning model, which result we are not able to calculate the loss function. So, we will need more research for the NN model and keep working on both of our data and project direction.

#### **Github repo:**

<https://github.com/Qing201/EEP596-RecSys-Project>

#### **IMDB Dataset:**

[IMDb Datasets](#)