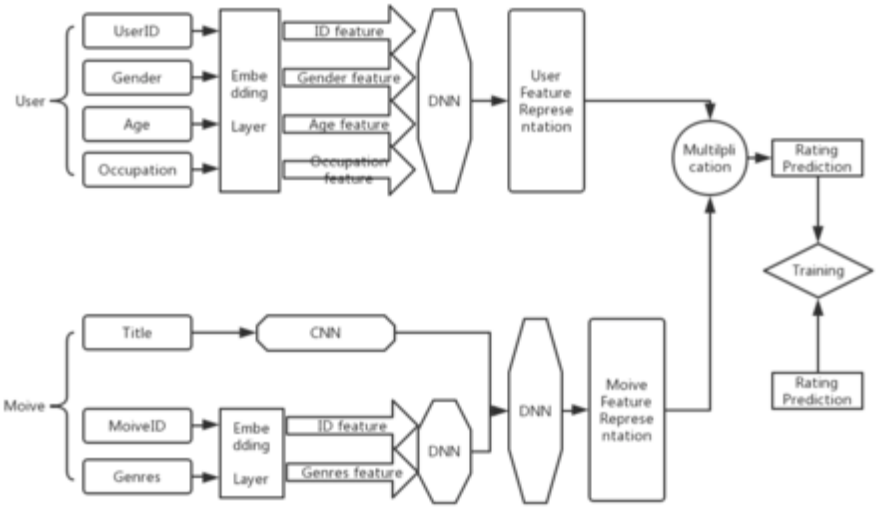
Q8

Recommendation System

Introduction

In Q8, to do recommendation system, we mainly divide into four parts: data exploration, data preprocessing, feature engineering, modelling and implentation. Detailed information are listed below.

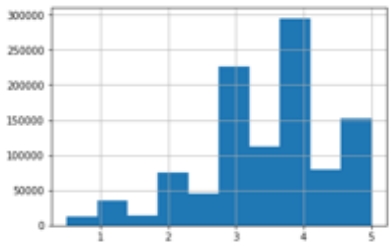


Methods:

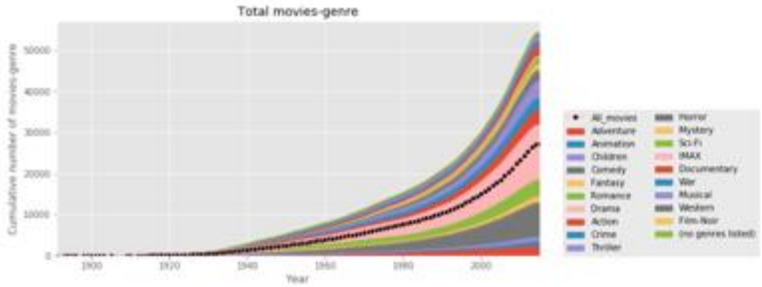
1. Data exploration

In data exploration, we mainly explore data from statistics, trend and top_N.

- Statistic: Rating



- Trend

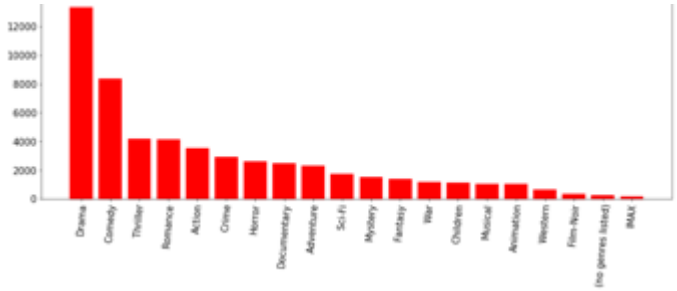


➢ Top N

(1) Top N movies

title	
Pulp Fiction (1994)	3498
Forrest Gump (1994)	3476
Silence of the Lambs, The (1991)	3247
Shawshank Redemption, The (1994)	3216
Jurassic Park (1993)	3129
Star Wars: Episode IV - A New Hope (1977)	2874
Braveheart (1995)	2799
Terminator 2: Judgment Day (1991)	2711
Matrix, The (1999)	2705
Schindler's List (1993)	2598
Toy Story (1995)	2569
Fugitive, The (1993)	2568

(2) Top N categories



2. Data Pre-processing

➢ Drop feature

(1) Timestamp in rating data

(2) Zip-code in user's data

➢ Encoding

(1) convert 'F' & 'M' in Gender into 0 & 1

(2) convert age to 0~7

(3) map genre into one-hot list

(4) extract the year from the title as a new colum

3. Feature Engineering

➢ Prepare data

In the feature engineering and model part, the method I took was to build the features and run the model cyclically on a station-by-station basis. So I need to divide the previous big table into 35 files according to the station.

Please check the '*cut_data.py*' for the detail of preparing data.

➢ User features

User average rating

User average rating on different genre

- Movies features

Movie average rating

Movie average rating by different gender

Movie average rating by different age

Movie average rating by different occupation

4. Model

First, the corresponding feature representations are obtained from the two files user_nn and movie_nn. The user feature vector and the movie feature vector are then matrix multiplied. The obtained data is the predicted output of the model, indicating that the user has a predicted score for the movie. Use the square loss function, trained by AdamgradOptimizer. finally, I Saved training data using tensorboard

- Prepare data for models
  - Normalization

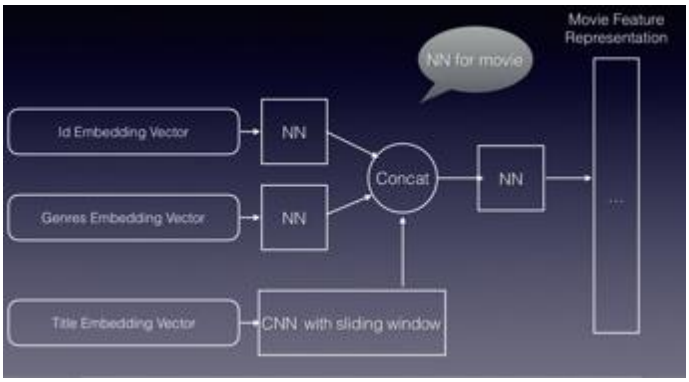
Rating data have Long-tailed distribution with lots of low-rate values. So we need to do some normalization before feed them into models. I chose log1p () to do this normalization.

  - Train and Test Set Split

Here I create a training and a test set from all of the ratings. The first step is to order the movie and user IDs and assign a sequential number to them - so that they can be added to the training and test matrices in order and so that a mapping exists back from the movie and user ID to its index in the matrix. Scondly, the train_test_split function is used to allocate 80% of ratings to the training set and 20% to the test set. Finally, the training and the test set are both matrices of the same size i.e movies in rows and users in columns by adding them to a matrix in ascending order of userId and movild using.

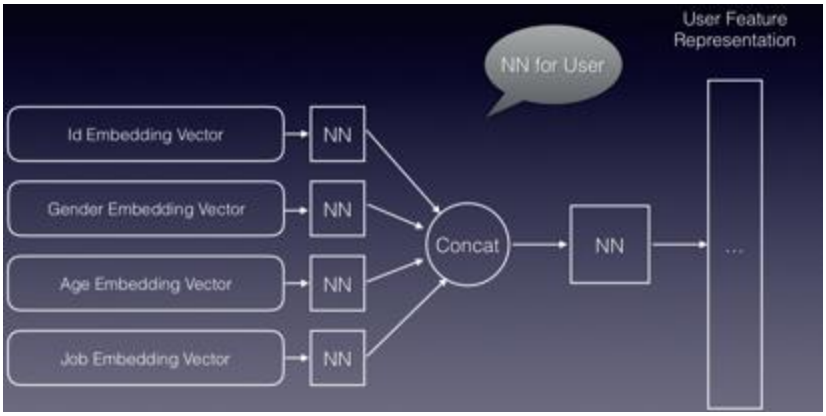
- Model Design
  - Movie NN

In this NN, it mainly includes constructing neural network models related to movies, including embedding of movie features and feature extraction using convolutional neural networks and conventional neural networks.



✧ User NN

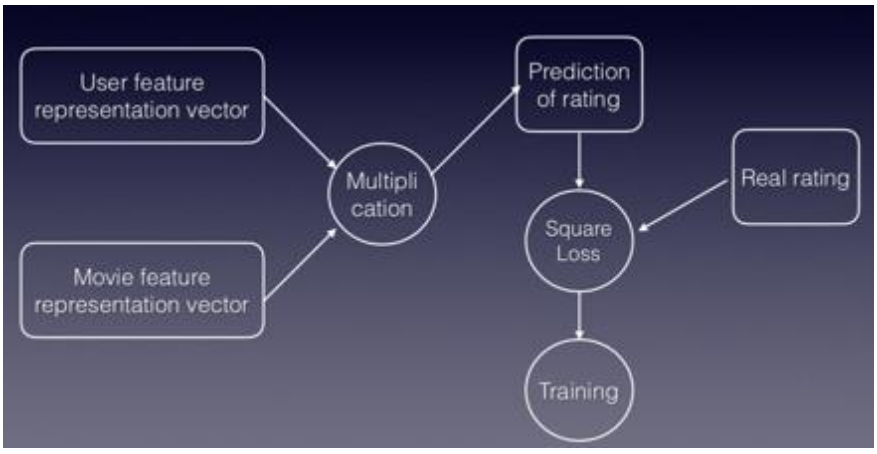
Constructing a neural network for the attributes of the user mainly includes creating embedding layers for the user and creating a neural network for user feature extraction.



5. implementation

➢ Train model

When we train the model, we input the user vector and the movie vector into the model, and obtain the prediction result. By calculating the SE between the real data and continuously optimizing the model parameters, the accuracy of the best model parameters can be achieved which is about 25%, the loss error is about 1.5



➢ Prediction

‘Rating movie ‘Given user and movie, pass forward the users and movies data through the NN, the score will be the output.