Wong Pang Chi 18063466D

The video link of presentation:

https://connectpolyu-my.sharepoint.com/:v:/g/personal/18063466d_connect_polyu_hk/EUqjCD6avfNFsH3dfNqS-HMB7dVv97Z0vNTGivnW15buTA?e=xF9H2Y

Youtube link:
https://youtu.be/RVVSXJ4cQps

1. Introduction:

For all the tasks, I followed the requirements of "A_fundamental_template.pdf".

In this project, there are total two tasks. The first one is fill in the blank space of New_Teleplay.csv. the second task is predict user 53698's personalized rating of all teleplays.

For the task one, we mainly focus on the linear regression model that is built to predict the rating of recently published teleplay. For the teleplay data, it contains 7 features, namely, ID, name, genres, length, episodes, rating and members. Since we need to predict the rating, we store rating data as the y label for prediction. For the ID and name data, they are not important data for predicting the rating, therefore, we will not use them as the input features to build the linear regression model. As a result, for the task one, we use 4 input features which are genres, type(length), episodes and members.

For the task two, there are three sub-tasks, namely, neural network, content-based recommendation system and collaborative filtering-based recommendation system. For neural network, the predicted values of teleplay will be compared with the true value and the grading is based on prediction accuracy. For content-based recommendation system, we use it to find user 53698's favourite teleplay, and based on that, recommend similar teleplays for user 53698. For the collaborative filtering-based recommendation system, we predict user 53698's personalized rating of teleplays that are not rated yet.

2. Task 1

2.1 Data preprocessing:

For task 1, we use the data from Teleplay.csv, New_Teleplay.csv to do the prediction. We first input the data, then filtrate the useless data. Also, we do the feature encoding and scaling and fill the null data for further usage. The specific process is at below.

Input data:

Task 1 uses the data from Teleplay.csv, New_Teleplay.csv to do the prediction. For the data from Rating.csv, we use map and reduce to do the aggregation and find out the average rating of each teleplay from different users, the average rating is attached as a new feature to the corresponding teleplay.

Data filtration:

The data is read from files and stored into data frames. We want the data with higher purity, the program needs to check the null data in the columns of each row and delete it. Also, we select the suitable features for the prediction and drop the unsuitable features.

***Feature** encoding:*

The program changes the string values of type and episodes to labels. Also, since the genres are strings and a movie can have multiple genres, the program needs to do one hot encoding to indicate which genres are contained in each movie. For example, we first find out there are total 43 genres and store them in x and initialize them with 0, if the movie belongs to some genres, we assign 1 to those genres in the x array. Therefore, all the genres become 0 or 1 in the array and we can use them for further analyzing.

Feature scaling:

The program changes the scale of type, episodes and members using data smoothing techniques. We want to reduce the scales differences of features.

Fill the null data

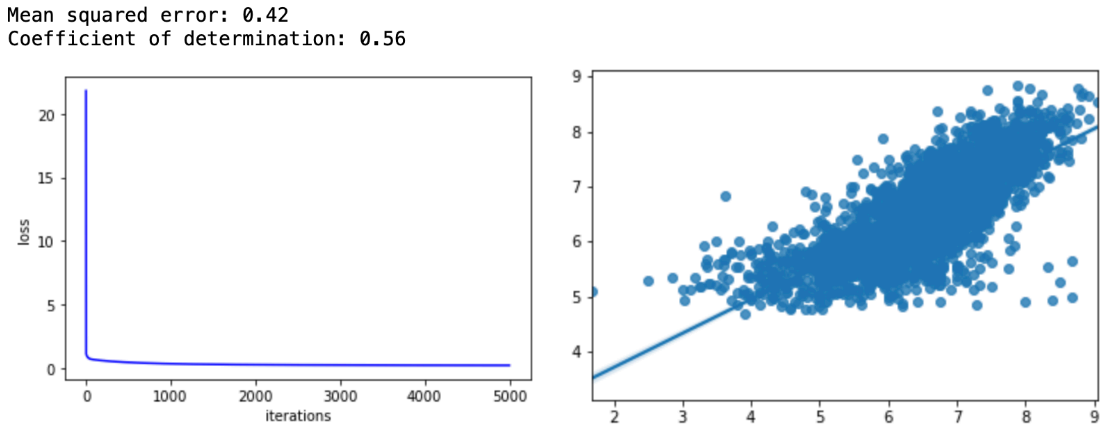
We fill the null data of predicted data (data from New_Telelpay.csv) since we need to insert all the rows of the data at the end.

2.2 Model design and implementation:

The linear regression model is used for task 1 in this project, since it is good at predicting the data, we can utilize it to predict the rating. For the linear regression model, we use gradient descent algorithm. The program separates the data into training data and test data, namely, train_x, test_x, train_y, test_y. Then, the model is trained by using those training data. After that, the program uses the test data to test the model and store the predicted rating into y_pred.

2.3 Performance evaluation and discussions:

The results are shown at below:



```
Mean squared error: 0.42
Coefficient of determination: 0.56
```

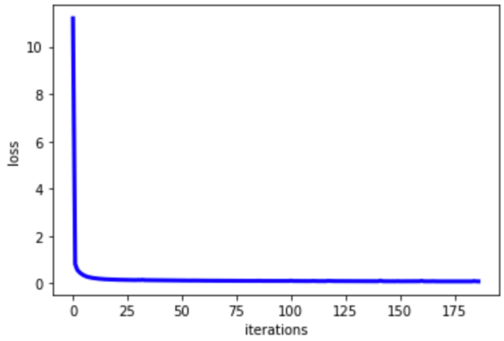
Th mean squared error is 0.42 and the R^2 value is 0.56. we can see that the loss drop rapidly in a few iterations and become stable afterwards. Also, we plot the graph of test_y and y_pred to see the difference of values roughly. We can see that the R^2 value is not that high but it is sufficient to predict a satisfied result. For the graph of test data and predicted data, only few data and far from the regression line, most of the test data are near the regression line so the performance of the model is satisfied but still can be improved.

3. Task 2

For task 2, we use the data from Teleplay.csv and Rating.csv. First, we make a neural network model to find out the prediction accuracy. Second, we make a content-based recommendation system, we use it to find user 53698's favourite teleplay, and based on that, recommend similar teleplays for user 53698. Third, we make a collaborative filtering-based recommendation system, we predict user 53698's personalized rating of teleplays that he/she have not rated yet. The specific process is at below.

3.1 Neural Network:

The neural network model is built based on Multi-Layer Preception(MLP) technique. MLP is an artificial neural network consist of multiple layers including hidden and output layer. It utilizes backpropagation technique for training data. We use this model with stochastic gradient-based optimizer to find out the prediction accuracy of teleplay data. As well as task 1, we split the teleplay data into training data and test data. The training data is used to train the model, the test data is then used to find out the prediction accuracy of the model. The result is at below:



```
Accuracy: 0.7539170884731553
```

From the graph above, we can see that the loss of neural network model drops rapidly in few iterations and become stable afterwards. The accuracy is around 0.75, we can conclude that the accuracy is high enough to predict the values.

3.2 content-based recommendation system:

3.2.1 Data preprocessing:

For the content-based recommendation system, we use the data from Teleplay.csv and Rating.csv. The specific process is at below:

Input data:

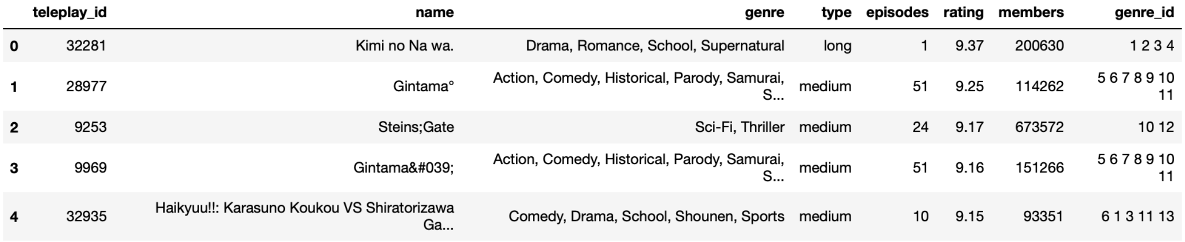
For the content-based recommendation system, we use the data from Teleplay.csv and Rating.csv.

Data filtration:

we first eliminate the data without rating and drop the data with null values.

Merge genre id to teleplay id:

We change the genres of movies to labels, namely, genre id. We insert the genre id to each teleplay movie so we can have a clear picture to find out the differences in genres of different movies. The graph is at below:



	teleplay_id	name	genre	type	episodes	rating	members	genre_id
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	long	1	9.37	200630	1 2 3 4
1	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	medium	51	9.25	114262	5 6 7 8 9 10 11
2	9253	Steins;Gate	Sci-Fi, Thriller	medium	24	9.17	673572	10 12
3	9969	Gintama&#039;	Action, Comedy, Historical, Parody, Samurai, S...	medium	51	9.16	151266	5 6 7 8 9 10 11
4	32935	Haikyuu!!: Karasuno Koukou VS Shiratorizawa Ga...	Comedy, Drama, School, Shounen, Sports	medium	10	9.15	93351	6 1 3 11 13

Group the data by mean, median and size:

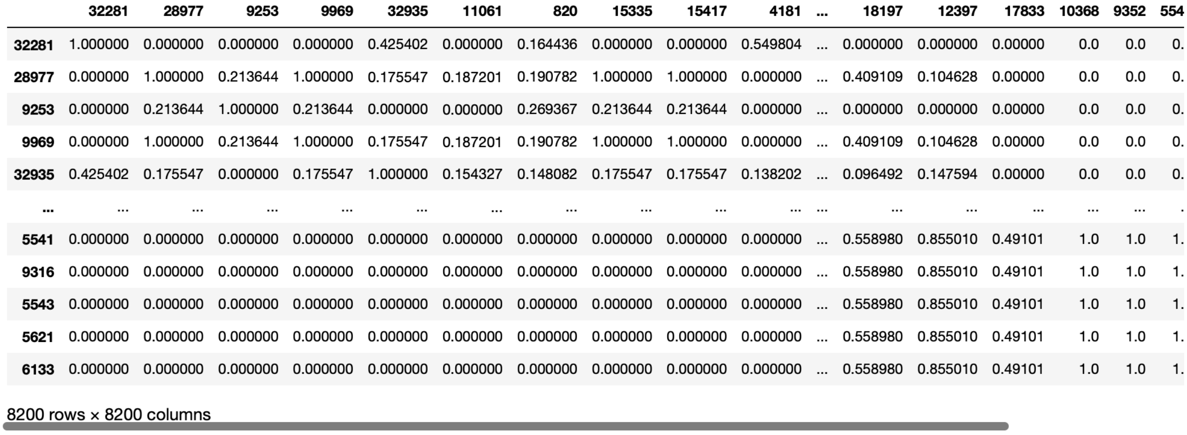
For the ratings in Rating.csv, we group the data and do the aggregation to find out the rating mean, median and size(number of users rated that movie) for further usage.

3.2.2 TD-IDF vectors:

We use TD-IDF vectors and cosine similarity techniques to compare each word of the genre data and calculate the similarity of movies based on the genres.

3.2.3 Performance evaluation and discussions:

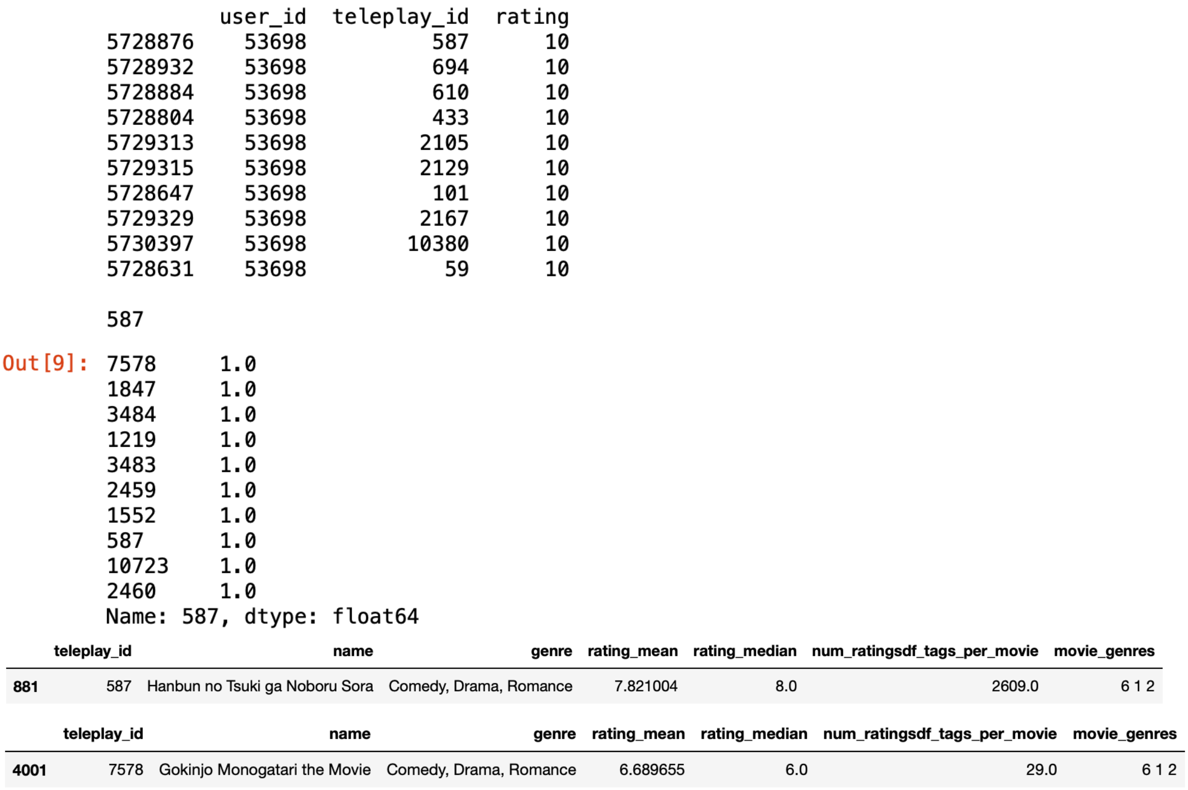
After utilize TDIDF, we now have the matrix of the similarity of teleplay. The graph is at below:



	32281	28977	9253	9969	32935	11061	820	15335	15417	4181	...	18197	12397	17833	10368	9352	554
32281	1.000000	0.000000	0.000000	0.000000	0.425402	0.000000	0.164436	0.000000	0.000000	0.549804	...	0.000000	0.000000	0.00000	0.0	0.0	0.
28977	0.000000	1.000000	0.213644	1.000000	0.175547	0.187201	0.190782	1.000000	1.000000	0.000000	...	0.409109	0.104628	0.00000	0.0	0.0	0.
9253	0.000000	0.213644	1.000000	0.213644	0.000000	0.000000	0.269367	0.213644	0.213644	0.000000	...	0.000000	0.000000	0.00000	0.0	0.0	0.
9969	0.000000	1.000000	0.213644	1.000000	0.175547	0.187201	0.190782	1.000000	1.000000	0.000000	...	0.409109	0.104628	0.00000	0.0	0.0	0.
32935	0.425402	0.175547	0.000000	0.175547	1.000000	0.154327	0.148082	0.175547	0.175547	0.138202	...	0.096492	0.147594	0.00000	0.0	0.0	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
5541	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.558980	0.855010	0.49101	1.0	1.0	1.
9316	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.558980	0.855010	0.49101	1.0	1.0	1.
5543	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.558980	0.855010	0.49101	1.0	1.0	1.
5621	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.558980	0.855010	0.49101	1.0	1.0	1.
6133	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.558980	0.855010	0.49101	1.0	1.0	1.

8200 rows × 8200 columns

As a result, we can now find out the user 53698's favorite teleplays, and recommend the similar teleplays for user 53698. See the graph below:



```
         user_id  teleplay_id  rating
5728876    53698          587      10
5728932    53698          694      10
5728884    53698          610      10
5728804    53698          433      10
5729313    53698         2105      10
5729315    53698         2129      10
5728647    53698          101      10
5729329    53698         2167      10
5730397    53698        10380      10
5728631    53698           59      10

587

Out[9]: 7578     1.0
        1847     1.0
        3484     1.0
        1219     1.0
        3483     1.0
        2459     1.0
        1552     1.0
        587      1.0
        10723    1.0
        2460     1.0
        Name: 587, dtype: float64
```

	teleplay_id	name	genre	rating_mean	rating_median	num_ratingsdf_tags_per_movie	movie_genres
881	587	Hanbun no Tsuki ga Noboru Sora	Comedy, Drama, Romance	7.821004	8.0	2609.0	6 1 2

	teleplay_id	name	genre	rating_mean	rating_median	num_ratingsdf_tags_per_movie	movie_genres
4001	7578	Gokinjo Monogatari the Movie	Comedy, Drama, Romance	6.689655	6.0	29.0	6 1 2

From the graph above, we can see that user 53698 has many favorite teleplays, many of the teleplays he/she rated is 10/10 such teleplay 587. We find out the similar movies of the teleplay with teleplay id 587 and recommend the top 10 of them to user 53698. For example, we recommend teleplay with teleplay id 7578 to user 53698 since it has the same genres as teleplay 587. As we can see, the content-based recommendation system based on user's content preference is quite useful to recommend similar teleplays.

3.3 collaborative filtering-based recommendation system:

3.3.1 Data preprocessing:

For the collaborative filtering-based recommendation system, the specific process is at below:

Input data:

For the collaborative filtering-based recommendation system, we use the data from Teleplay.csv and Rating.csv.

Data filtration:

We only choose the data with rating. It means that if the rating of user is -1, the user does not rate the movie yet, then we drop this data.

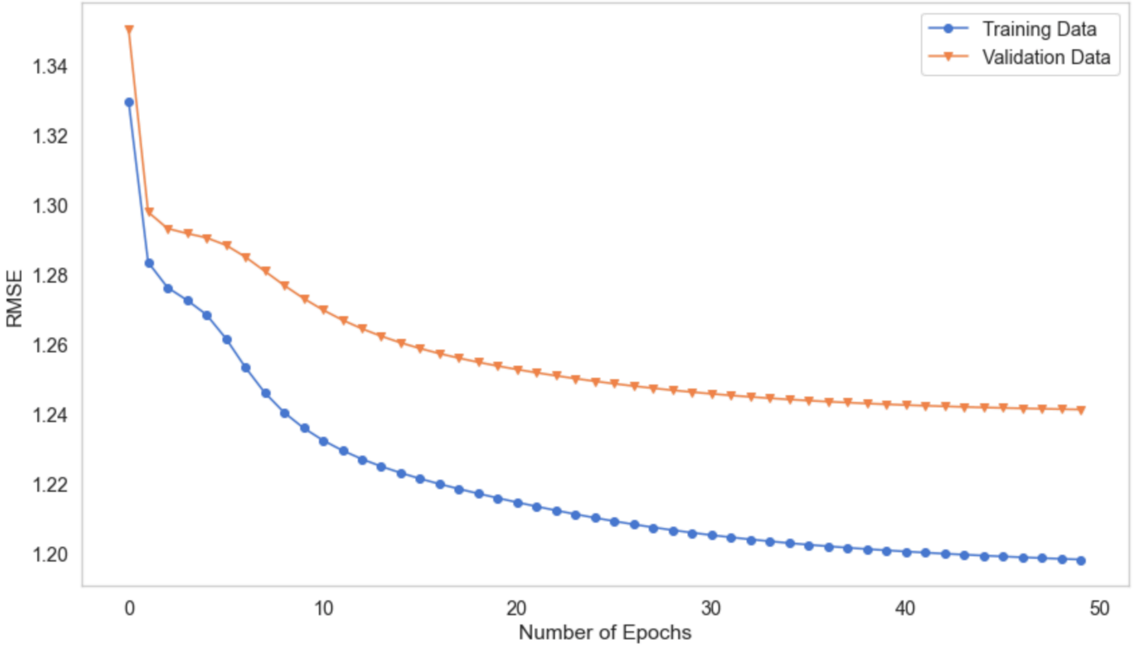
Data Preprocessing:

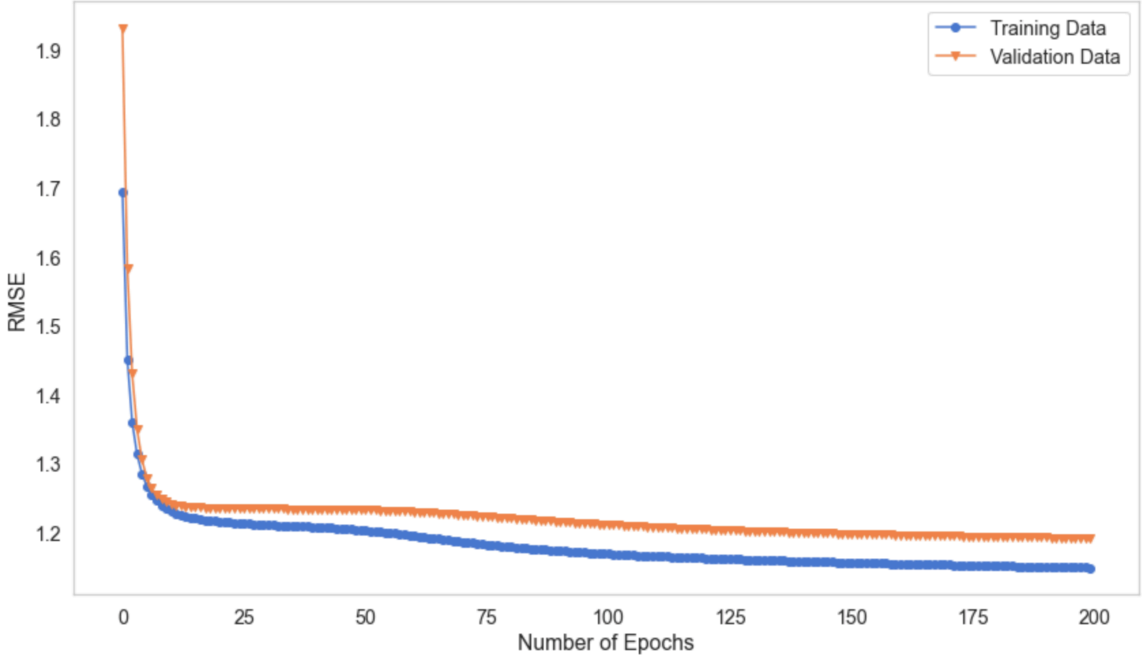
Set up the train-test split function to split the training and validation data. Also, we make a RMSE function to input to the SGD model.

3.3.2 Model design and Implementation:

For the collaborative filtering-based recommendation system, we use Stochastic Gradient Descent (SGD) model to train our data since SGD is good to train large-scale and sparse machine learning problems. As I mentioned, we split the data into training data and validation data, after we use training data to train the model, we plot the train error together with the validation error. At the end, we predict user 53698's rating to teleplays that are not rated yet and write then into the 18063466D_task2.csv.

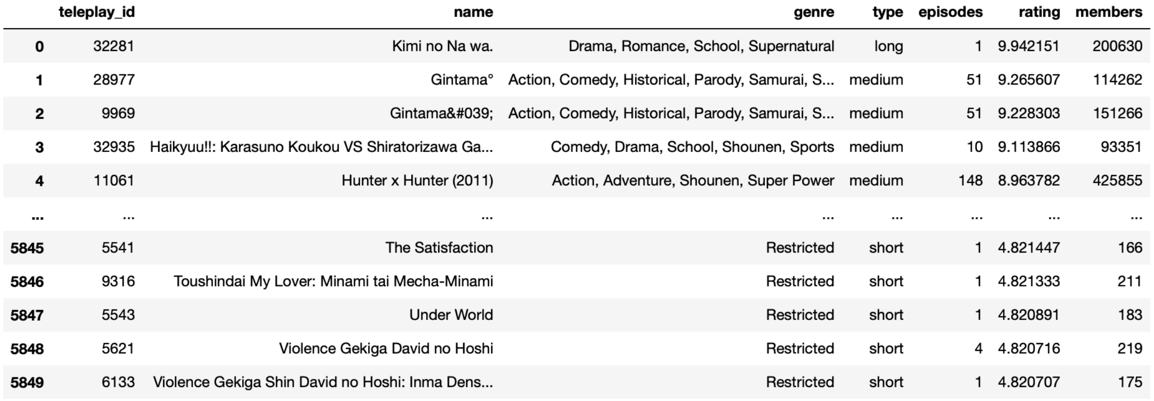
3.3.3 Performance evaluation and discussions:





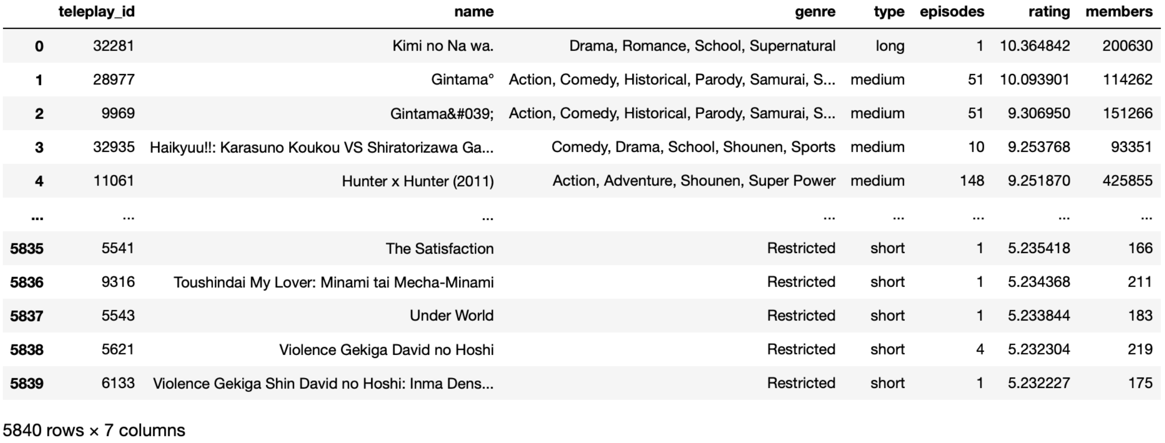
The graphs above show the RMSE loss of training data and validation data in 50 and 200 iterations respectively. For the 50 iterations graph, the RMSE of validation data and training data are between 1.24 to 1.26 and around 1.20 respectively. For the 200 iterations graph, the RMSE of validation data is between 1.2 to 1.3, the RMSE of training data is between 1.1 to 1.2. The RMSE of validation and training data does not drop so much even after 200 iterations. Also, we can see that the RMSE of validation data is higher than that of training data but the difference is small. As a result, the model is good fitted after 200 iterations based on the RMSE.

50 iterations:



	teleplay_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	long	1	9.942151	200630
1	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	medium	51	9.265607	114262
2	9969	Gintama&#039;	Action, Comedy, Historical, Parody, Samurai, S...	medium	51	9.228303	151266
3	32935	Haikyuu!!: Karasuno Koukou VS Shiratorizawa Ga...	Comedy, Drama, School, Shounen, Sports	medium	10	9.113866	93351
4	11061	Hunter x Hunter (2011)	Action, Adventure, Shounen, Super Power	medium	148	8.963782	425855
...	...	...	...	...	...	...	...
5845	5541	The Satisfaction	Restricted	short	1	4.821447	166
5846	9316	Toushindai My Lover: Minami tai Mecha-Minami	Restricted	short	1	4.821333	211
5847	5543	Under World	Restricted	short	1	4.820891	183
5848	5621	Violence Gekiga David no Hoshi	Restricted	short	4	4.820716	219
5849	6133	Violence Gekiga Shin David no Hoshi: Inma Dens...	Restricted	short	1	4.820707	175

200 iterations:



	teleplay_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	long	1	10.364842	200630
1	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	medium	51	10.093901	114262
2	9969	Gintama&#039;	Action, Comedy, Historical, Parody, Samurai, S...	medium	51	9.306950	151266
3	32935	Haikyuu!!: Karasuno Koukou VS Shiratorizawa Ga...	Comedy, Drama, School, Shounen, Sports	medium	10	9.253768	93351
4	11061	Hunter x Hunter (2011)	Action, Adventure, Shounen, Super Power	medium	148	9.251870	425855
...	...	...	...	...	...	...	...
5835	5541	The Satisfaction	Restricted	short	1	5.235418	166
5836	9316	Toushindai My Lover: Minami tai Mecha-Minami	Restricted	short	1	5.234368	211
5837	5543	Under World	Restricted	short	1	5.233844	183
5838	5621	Violence Gekiga David no Hoshi	Restricted	short	4	5.232304	219
5839	6133	Violence Gekiga Shin David no Hoshi: Inma Dens...	Restricted	short	1	5.232227	175

5840 rows × 7 columns

The graphs above shows the prediction of user 53698 in 50 and 200 iterations respectively, we can see that some of the predicted rating is slightly out of the range after 200 iterations. The result is acceptable but can be further improved to make an accurate prediction such as change some parameters in the train-test-split function or may be run more iterations to see the difference.

4. Future work:

For task1, in the future, I may try to apply the one hot encoding techniques to further separate the episode and member data. As a result, the data scale is more similar to each other and the features become more detail, the accuracy may be improved. However, too many features may also lower the accuracy of the model and cause overfitting problem, I need to spend more time to find out how to assign the features and applying regularization techniques to the model. Also, I will try to input different parameters to train the model to test which features can train the model better. Nevertheless, in this task1, I extend the input features of the model so multiple features can be input to the model. In the future, I may try to use PCA to reduce the dimensions of the features and see which method is better for training a linear regression model.

For task 2, in the future, I will try to use neural network to find the prediction accuracy of the rating from Rating.csv. I will try to improve the content-based recommendation system model by input more data such as null data. I can assign mean values to those null data, so that I can have more data to train the model of content-based recommendation system and collaborative filtering-based recommendation system. Also, for the collaborative filtering-based recommendation system model, I will try to split the training and validation data in a different way to train the model to if there are any differences.

5. Summary:

In this project, there are two tasks. First one is to design the linear regression model to predict the user rating of teleplays. Task two is implement neural network to find out the prediction accuracy of teleplay data, design recommendation systems to provide recommendation services and predict user 53698's rating to teleplays.

A summary of all sub tasks is at below:

- Linear regression model:
  - predict the rating of recently published teleplays.
- *Neural network*:
  - predicted values will be compared with the true value and the grading is based on prediction accuracy.
- *content-based recommendation system*:
  - find user 53698's favourite teleplay, and based on that, recommend similar teleplays for user 53698.
- *collaborative filtering-based recommendation system:*
  - predict user 53698's personalized rating of all teleplays.

6. References:

"numpy.array¶," *numpy.array - NumPy v1.20 Manual*. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.array.html. [Accessed: 15-Mar-2021].

"sklearn.linear_model.LinearRegression¶," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. [Accessed: 15-Mar-2021].

"sklearn.decomposition.PCA," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. [Accessed: 15-Mar-2021].

"pandas.read_csv¶," *pandas.read_csv - pandas 1.2.3 documentation*. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html. [Accessed: 15-Mar-2021].

"1.17. neural network MODELS (SUPERVISED)¶." [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html. [Accessed: 05-May-2021].