Final Report

Konovalova Oksana, BS21-DS-02

Introduction

A recommender system is a type of information filtering system that suggests items or content to users based on their interests, preferences, or past behavior. With countless options available, the need for personalized recommendations has become more crucial than ever. This report delves into the realm of movie recommendation system the deep learning framework Keras.

Data analysis

In analyzing the data, three files (u.data, u.user and u.item) from the entire dataset were considered because these files were used in future work.

When analyzing u.data, several conclusions were drawn. First, there is no missing data. Second, the rating is mostly in the range of 3 to 5 (fig. 1). Finally, there is a "timestamp" column that is not needed for further work.

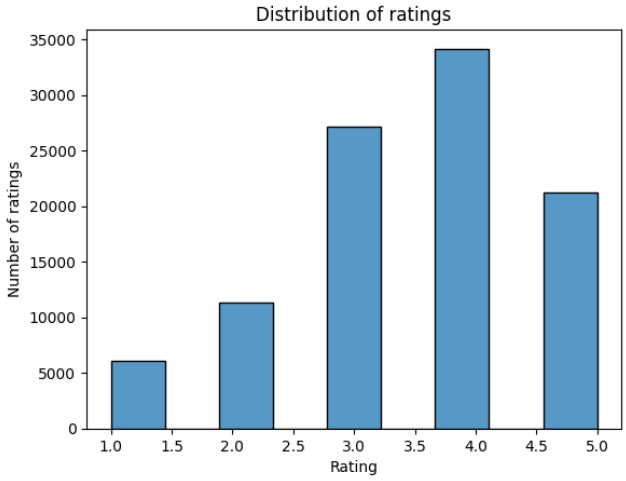

Fig. 1

Several conclusions were also drawn when analyzing the u.user data. Firstly, there are no missing data, which is pleasing as this data is important for model training and it is not found in all datasets. Secondly, it can be seen that the dataset has more people between 20 and 35 years old (fig. 2) and thus a majority of students (fig. 3). Third, there are more males than females by more than twice as much (fig.4). Finally, there is also a column here that is not needed for further work, which is "zip_code". When preprocessing this table, the "gender" column was changed from M to 0 and W to 1. Also, the occupation information was encoded using LabelEncoder.

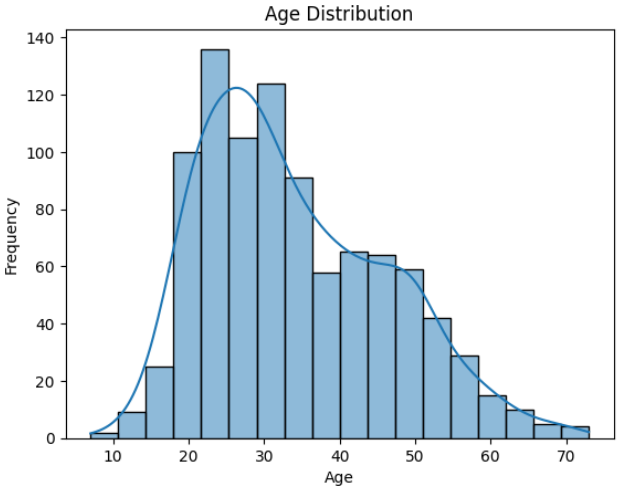

Fig. 2

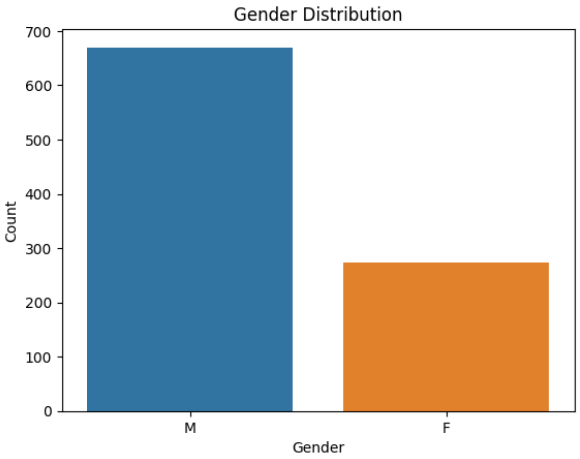

Fig. 4

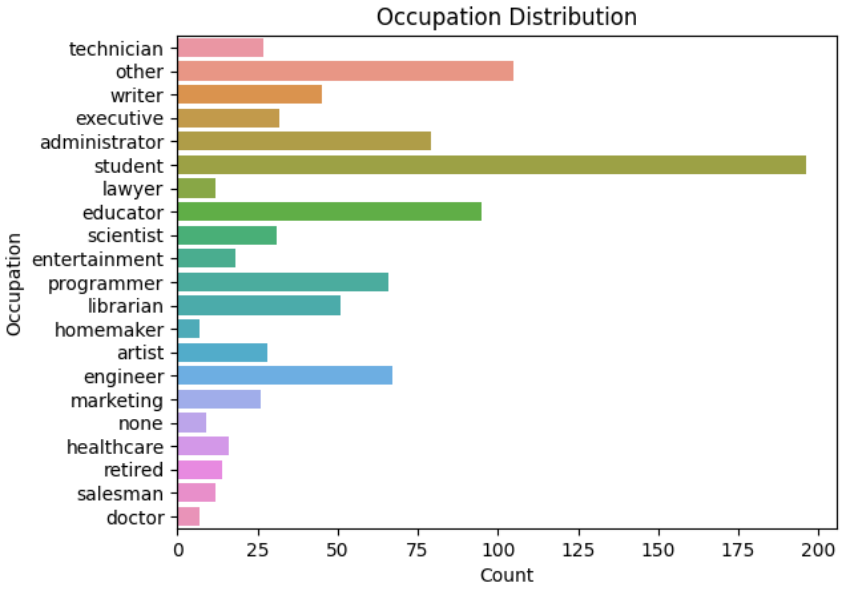

Fig. 3

When analyzing u.item, to can see the missing data (fig. 5), but it doesn't interfere because this data was just deleted because it is not used. It can also be seen that the most popular genres are drama and comedy (fig. 6). Finally, there were a few columns in this table that were removed because they were not needed for model training. During preprocessing, information about movie titles and genre information was taken from this table.

After preprocessing all tables were merged into one table and then split into training, validation and test set.

```
item.isnull().sum()

item_id                0
movie_title            0
release_date           1
video_release_date    1682
IMDb_URL               3
unknown                0
Action                 0
```

Fig. 5

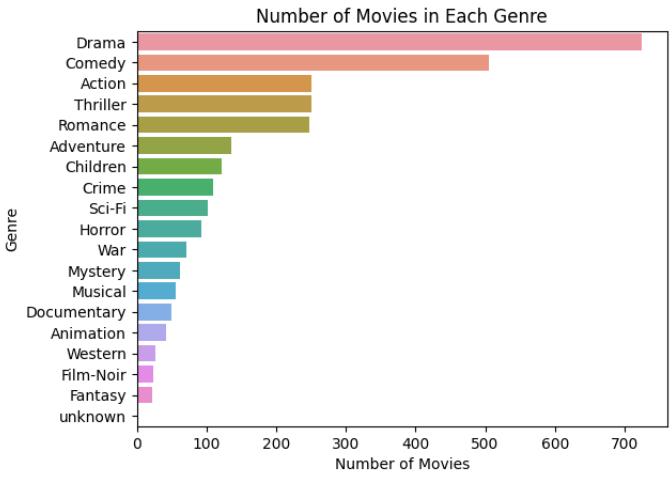

Fig. 6

Model Implementation

Model Architecture

The recommendation model follows a collaborative filtering approach, incorporating both user and item embeddings. The key components of the model include:

Embedding Layers: Separate embedding layers are used for user and item IDs, as well as additional features such as age, gender, occupation, and genres. These embedding layers capture latent representations for users and items.

Dense Layers: After concatenating the embeddings, dense layers are employed to capture higher-level features and interactions.

- Dense(64, activation='relu'): The first dense layer with ReLU activation function.
- Dropout(0.5): Dropout layer to prevent overfitting by randomly setting a fraction of input units to 0 at each update during training.
- Dense(32, activation='relu', kernel_regularizer=l2(0.01)): Second dense layer with ReLU activation and L2 regularization on the kernel weights.
- Dropout(0.5): Another dropout layer.
- Dense(16, activation='relu', kernel_regularizer=l2(0.01)): Third dense layer with ReLU activation and L2 regularization.

Output Layer: The output layer generates a single rating prediction for each user-item pair.

Regularization

Regularization techniques such as dropout and L2 regularization have been incorporated into the model to mitigate overfitting. Dropout layers randomly drop a certain percentage of input units during training, preventing the network from relying too much on specific features. L2 regularization penalizes large weights in the network, promoting simpler models.

Model Advantages and Disadvantages

Model Advantages:

- Non-Linearity: The model incorporates non-linear activation functions (ReLU) in the dense layers, allowing it to capture complex relationships between user and item features.
- Embedding Layers: The use of embedding layers for user and item IDs helps the model learn meaningful representations for users and items, enabling better generalization.
- Dropout: The inclusion of dropout layers aids in preventing overfitting by randomly dropping input units during training, promoting more robust feature learning.
- Regularization: L2 regularization on the kernel weights in certain dense layers helps control the complexity of the model, reducing the risk of overfitting.

Model Disadvantages:

- Simplicity: While simplicity can be an advantage, the model may not capture highly intricate patterns present in user-item interactions. More complex architectures might be necessary for certain scenarios.
- Data Dependency: The model's performance heavily relies on the quality and quantity of training data. Insufficient or biased data may lead to suboptimal predictions.
- Cold Start Problem: The model may face challenges in making accurate predictions for new users or items.
- Interpretability: Neural network models, especially with multiple layers, are often considered as "black-box" models, making it challenging to interpret the learned representations and decision-making processes.

Training Process

The model is trained using the mean squared error (MSE) loss function, which measures the average squared difference between predicted and true ratings. The Adam optimizer is employed for gradient descent, adjusting learning rates adaptively during training.

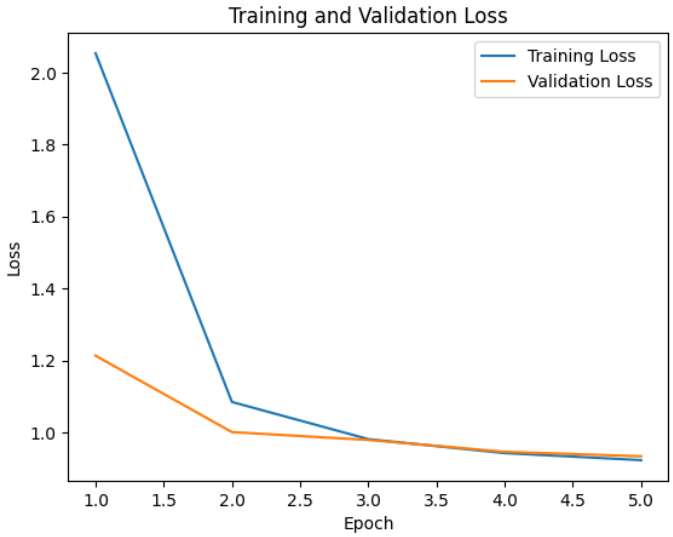

Fig. 7

Evaluation

The model's performance is evaluated on a test dataset using common evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). These metrics provide insights into how well the model generalizes to unseen data and accurately predicts user ratings.

- RMSE measures the average magnitude of the errors between predicted and actual values.
- MAE measures the average absolute difference between predicted and actual values.

```
Root Mean Squared Error (MSE): 0.9514706368879132
Mean Absolute Error (MAE): 0.7532620705604554
```

Fig. 8 (evaluation of model)

Results

The model predicts the k best movies and outputs them. Below are examples for some users from the test set:

	item_id	movie_title
49	50	Star Wars (1977)
126	127	Godfather, The (1972)
301	302	L.A. Confidential (1997)
312	313	Titanic (1997)
315	316	As Good As It Gets (1997)

Fig. 9 (user_id = 26)

	item_id	movie_title
312	313	Titanic (1997)
317	318	Schindler's List (1993)
356	357	One Flew Over the Cuckoo's Nest (1975)
426	427	To Kill a Mockingbird (1962)
602	603	Rear Window (1954)

Fig. 10 (user_id = 815)

	item_id	movie_title
49	50	Star Wars (1977)
99	100	Fargo (1996)
312	313	Titanic (1997)
314	315	Apt Pupil (1998)
474	475	Trainspotting (1996)

Fig. 11 (user_id = 432)