Movie Recommendation Systems

Ashish Podduturi (ap1822)

Ushasee Das (ud29)

Mahesh Reddy Annapureddy (ma1700)

Karthik Chava (kc1157)

PROBLEM STATEMENT

Movie Recommendation:-

- Top N recommendations for each user :-
  - Input :- userId
  - Output :- Top N movies based on descending order of predicted ratings
- Evaluation :-
  - Prediction Accuracy
    - RMSE
    - MAE
  - Relevance
    - Precision
    - Recall
    - F-measure

About our Data

- The Dataset was obtained from Movielens site which is part of Grouplens Research.
- There were one million entries in the movie ratings dataset.
    - It contained ratings from 6040 users.
    - About 3706 movies out of total 3883 movies available.
    - Training dataset to Test dataset ratio was 0.8 to 0.2.

Exploratory Data Analysis

- Analysed on 1000209 X 23 data
- No redundancies
- No missing values
- Contains data authenticity and no incorrect values.
- Correlation among columns is performed using pearson method.
- All the data features are independent and positively skewed and none of the columns are normally distributed.

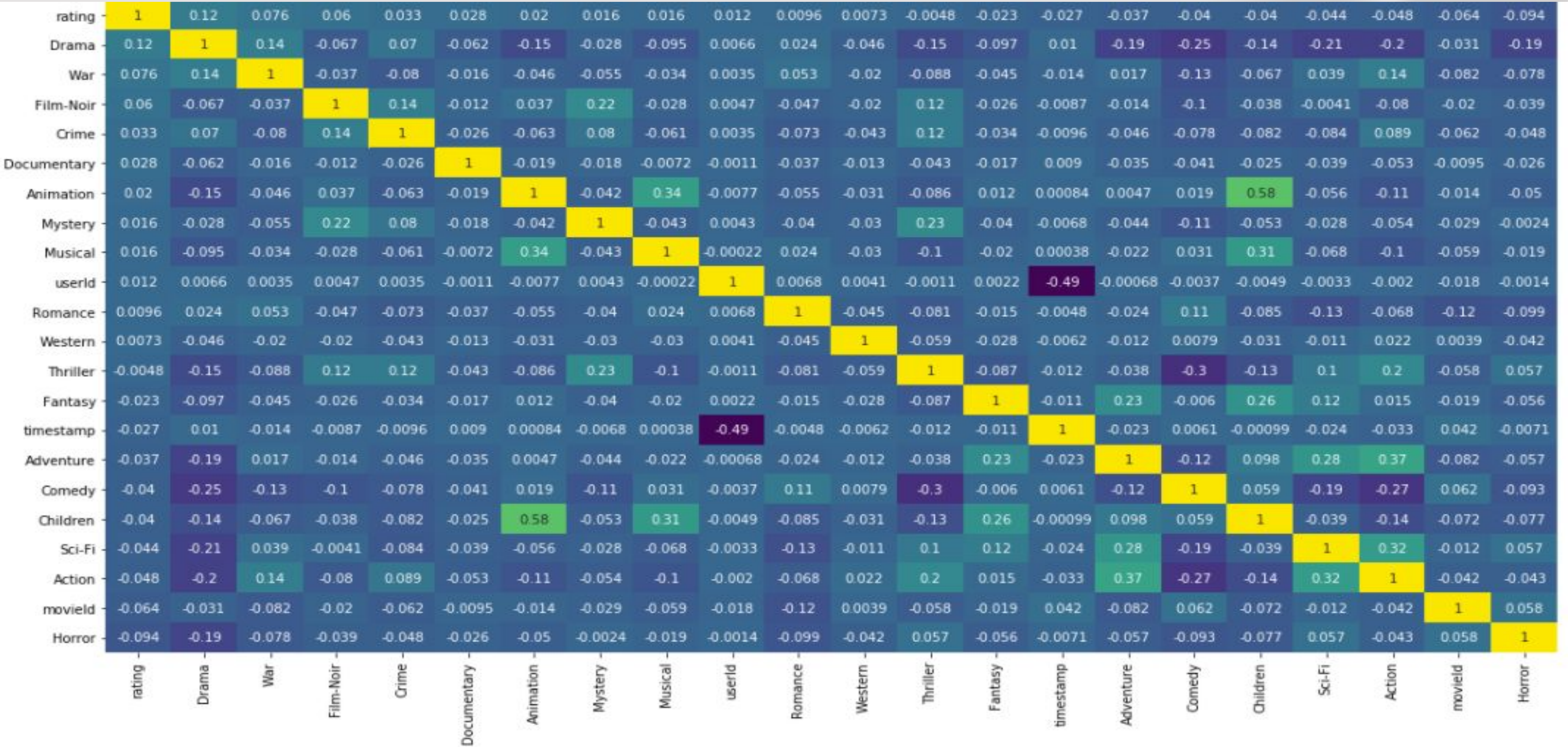
Information of the data:-

```
Information about the columns and datframe:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000209 entries, 0 to 1000208
Data columns (total 23 columns):
 #   Column       Non-Null Count    Dtype
---  ------       --------------    -----
 0   userId       1000209 non-null  int64
 1   movieId      1000209 non-null  int64
 2   rating       1000209 non-null  int64
 3   timestamp    1000209 non-null  int64
 4   Action       1000209 non-null  float64
 5   Adventure    1000209 non-null  float64
 6   Animation    1000209 non-null  float64
 7   Children     1000209 non-null  float64
 8   Comedy       1000209 non-null  float64
 9   Crime        1000209 non-null  float64
 10  Documentary  1000209 non-null  float64
 11  Drama        1000209 non-null  float64
 12  Fantasy      1000209 non-null  float64
 13  Film-Noir    1000209 non-null  float64
 14  Horror       1000209 non-null  float64
 15  Musical      1000209 non-null  float64
 16  Mystery      1000209 non-null  float64
 17  Romance      1000209 non-null  float64
 18  Sci-Fi       1000209 non-null  float64
 19  Thriller     1000209 non-null  float64
 20  War          1000209 non-null  float64
 21  Western      1000209 non-null  float64
 22  title        1000209 non-null  object
dtypes: float64(18), int64(4), object(1)
memory usage: 183.1+ MB
```

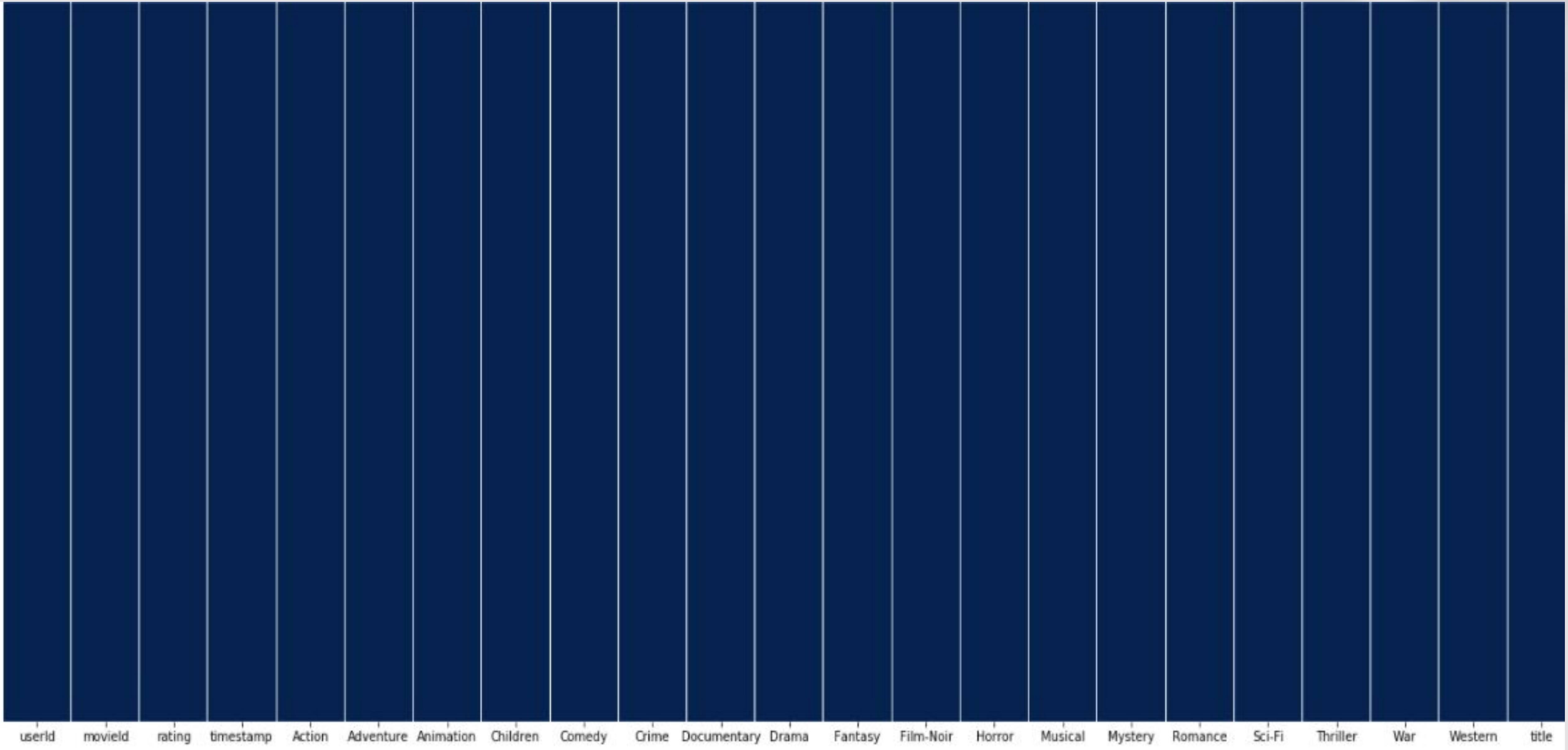
Data description (statistics):-

	count	mean	std	min	25%	50%	75%	max
userId	1000209.0	3.024512e+03	1.728413e+03	1.0	1506.0	3070.0	4476.0	6.040000e+03
movieId	1000209.0	1.865540e+03	1.096041e+03	1.0	1030.0	1835.0	2770.0	3.952000e+03
rating	1000209.0	3.581564e+00	1.117102e+00	1.0	3.0	4.0	4.0	5.000000e+00
timestamp	1000209.0	9.722437e+08	1.215256e+07	956703932.0	965302637.0	973018006.0	975220939.0	1.046455e+09
Action	1000209.0	2.574032e-01	4.372036e-01	0.0	0.0	0.0	1.0	1.000000e+00
Adventure	1000209.0	1.339250e-01	3.405719e-01	0.0	0.0	0.0	0.0	1.000000e+00
Animation	1000209.0	4.328395e-02	2.034957e-01	0.0	0.0	0.0	0.0	1.000000e+00
Children	1000209.0	7.217092e-02	2.587708e-01	0.0	0.0	0.0	0.0	1.000000e+00
Comedy	1000209.0	3.565055e-01	4.789672e-01	0.0	0.0	0.0	1.0	1.000000e+00
Crime	1000209.0	7.952438e-02	2.705556e-01	0.0	0.0	0.0	0.0	1.000000e+00
Documentary	1000209.0	7.908347e-03	8.857659e-02	0.0	0.0	0.0	0.0	1.000000e+00
Drama	1000209.0	3.544549e-01	4.783481e-01	0.0	0.0	0.0	1.0	1.000000e+00
Fantasy	1000209.0	3.629341e-02	1.870194e-01	0.0	0.0	0.0	0.0	1.000000e+00
Film-Noir	1000209.0	1.825718e-02	1.338801e-01	0.0	0.0	0.0	0.0	1.000000e+00
Horror	1000209.0	7.637004e-02	2.655894e-01	0.0	0.0	0.0	0.0	1.000000e+00
Musical	1000209.0	4.152432e-02	1.994996e-01	0.0	0.0	0.0	0.0	1.000000e+00
Mystery	1000209.0	4.016960e-02	1.963569e-01	0.0	0.0	0.0	0.0	1.000000e+00
Romance	1000209.0	1.474922e-01	3.545960e-01	0.0	0.0	0.0	0.0	1.000000e+00
Sci-Fi	1000209.0	1.572611e-01	3.640470e-01	0.0	0.0	0.0	0.0	1.000000e+00
Thriller	1000209.0	1.896404e-01	3.920166e-01	0.0	0.0	0.0	0.0	1.000000e+00
War	1000209.0	6.851268e-02	2.526237e-01	0.0	0.0	0.0	0.0	1.000000e+00
Western	1000209.0	2.067868e-02	1.423063e-01	0.0	0.0	0.0	0.0	1.000000e+00

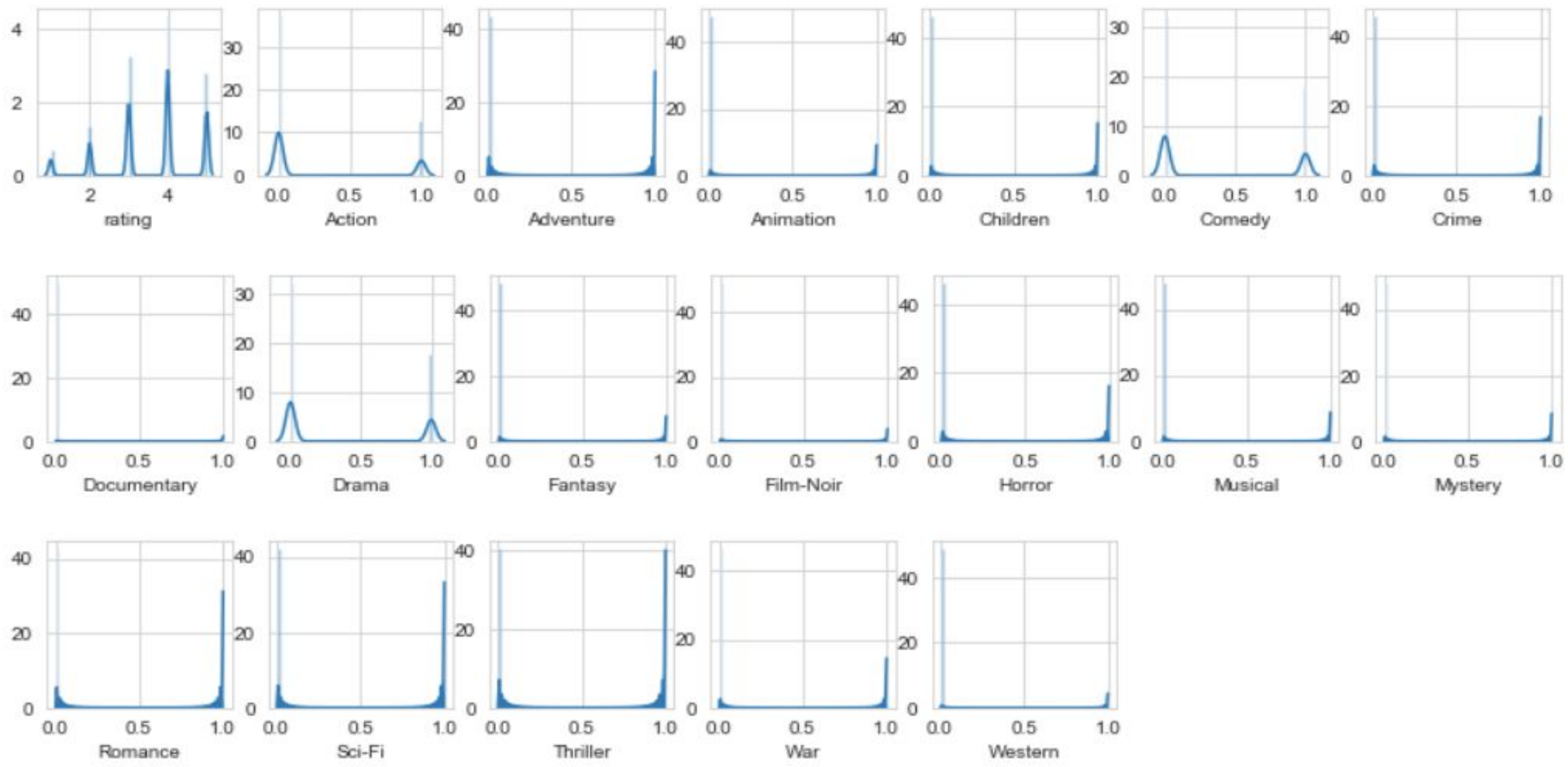
Correlation among data (pearson):-



Missing values heatmap



Data distribution :-



We can observe that none of the columns are normally distributed and all the variables are independent and positively skewed

APPROACH

Singular Vector Decomposition (SVD):-

- Singular Value Decomposition is a collaborative recommendation engine technique for decomposing a matrix into three matrices which yield more information concerning the matrix data

  A=UΣVT , where

  U is an m×m orthogonal matrix

  Σ is an diagonal m×n matrix

  V is an n×n orthogonal matrix

- Used for dimensionality reduction, noise reduction and also compression
- More stable than eigen decomposition.

Non-Negative Matrix Factorization (NMF):-

- It discovers latent factors in utility matrix.
- Maps users and movies to a k-dimensional concept space.
- Intuitively, Clustering the columns of the utility matrix
- Defined as $X \approx WH$ where
  - X is $n \times p$, W is $n \times r$, H is $r \times p$, $r \leq p$
- W gives the cluster centroids, i.e., the kth column gives the cluster centroid of kth cluster.
- This matrix factorization can be used for example for dimensionality reduction, source separation, and topic extraction

KNN

- The KNN algorithm, another collaborative filtering algorithm, is based on a simple premise, that similar things are close to each other
- It captures this idea of similarity by calculating cosine distances.
- The smaller the distance the more likely items are to be similar to one another.
- Thus, by finding the closest training samples to a point, it can predict the label for these based on cosine distances.
- Used KNNBasic method from surprise package in our model.

CoClustering

- Co-Clustering is a collaborative recommendation technique that given a matrix A seeks to cluster rows of A and columns of A at the same time.
- Simultaneous clustering along the rows and columns of the utility matrix.
- Each user and item assigned to cluster and co-cluster.
- Final rating depends on the average rating of the user cluster and the movie cluster

Basically, users and items are assigned some clusters C_u, C_i, and some co-clusters C_{ui}.

The prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i}),$$

Alternating Least Squares (ALS) :-

- Alternative Least Squares (ALS) is a matrix factorization algorithm that runs in parallel fashion and is built for large scale collaborative filtering problems .
- ALS trains by minimizing two loss functions alternatively.
    - It first fixes the user matrix and runs gradient descent with item matrix.
    - it fixes the item matrix and runs gradient descent with user matrix.
- ALS scales very well and does well with sparse datasets

Deep Neural Net

- We used collaborative filtering using Deep learning.
- The methods converts each user and item into embeddings which are then concatenated into one feature matrix.
- The feature matrix gets passed through neural network model.

```
Network(
  (embedding_m): Embedding(3707, 8)
  (embedding_u): Embedding(6041, 5)
  (lin1): Linear(in_features=13, out_features=200, bias=True)
  (lin2): Linear(in_features=200, out_features=80, bias=True)
  (lin3): Linear(in_features=80, out_features=50, bias=True)
  (lin4): Linear(in_features=50, out_features=20, bias=True)
  (lin5): Linear(in_features=20, out_features=1, bias=True)
  (relu): ReLU()
  (dropout): Dropout(p=0.1, inplace=False)
)
```

Ensemble method

- Our Ensemble method is a combination of all the other models we employ and based upon the idea of the wisdom of crowds.
- By taking into account many different models, and using their mean results, we can minimize error and hypothetically provide a perfect well-balanced result.
- Reduce Noise and avoid overfitting
- This model is built by taking the mean of the ratings predicting by each of the individual models.

Ensemble Rating = Mean(Ratings from different recommendation model)

RESULTS

1 MILLION DATASET

MODEL	PRECISION	RECALL	F-MEASURE	MAE	RMSE
SVD	0.683	0.683	0.683	0.684	0.872
NMF	0.671	0.671	0.671	0.722	0.915
KNN	0.677	0.677	0.677	0.705	0.894
CoClustering	0.676	0.676	0.676	0.718	0.916
Deep Neural Net	0.674	0.674	0.674	0.703	0.905
ALS	0.651	0.651	0.651	0.681	0.872
Ensemble	0.68	0.68	0.68	0.68	0.87

CONCLUSION

CONCLUSION

- Ensemble method can balance the bias variance trade-off and provided better results then base learner methods.

- We can use Ensemble method to combine different algorithms and methods like collaborative filtering, Content based filtering, neural networks etc.

FUTURE WORK

FUTURE WORK

- Instead of simple average of ratings, we can use weighted average of the rating to improve the performance of Ensemble model.

- If we use rank based method in our base model there is a chance of model performance improvement.

REFERENCES

- https://medium.com/the-andelaway/foundations-of-machine-learning-singular-value-decomposition-svd-162ac796c27d
- "Non-negative Matrix Factorization." Stanford. http://statweb.stanford.edu/~tibs/sta306bfiles/nnmf.pdf.
- https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm6a6e71d01761
- http://mlwiki.org/index.php/Co-Clustering
- https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1
- https://towardsdatascience.com/ensemble-machine-learning-wisdom-of-the-crowd-56df1c24e2f5#:~:text=In%20short%2C%20you%20heavily%20research,the%20process%20of%20decision%20making
- https://grouplens.org/datasets/movielens/

Thank You