Ensemble based Movie Recommendation System and Evaluation

Sanket Badhe (smb535)
Abhishek Bhatt (ab2083)
Siva Harshini Dev Bonam (sdb202)
Soumya Anthareddy (sa1607)

Instructor:
Prof. Yongfeng Zhang

DATASET

MovieLens https://grouplens.org/datasets/movielens/

- 1M
  - Users - 6000
  - Movies - 4000
  - Ratings - 1 million
  - Train set - 800193 ratings
  - Test set - 200016 ratings

- Full Latest(27M)
  - Users - 280,000
  - Movies - 58,000
  - Ratings - 27,000,000
  - Train set - 22207659 ratings
  - Test set - 5545785 ratings

Exploratory Data Analysis (EDA)

- Analysed on 27753444 x 24 data
- No missing values
- No redundancies
- Assumes data authenticity and so no incorrect values.
- Correlations calculated using Pearson correlation method.
  - Children-Animation (0.64)
  - Action-Adventure (0.33)
  - Musical-Animation (0.32)
  - Action-SciFi (0.32)
  - Children-Musical (0.31)
  - Thriller-Mystery (0.3)
  - ...
  - Comedy-Drama (-0.27)
  - Comedy-Thriller (-0.34)
- None of columns appears to be normally distributed. All are independent variables positively skewed.
- Action, Adventure, Crime, Fantasy and War are 5 significant genres that have ratings.

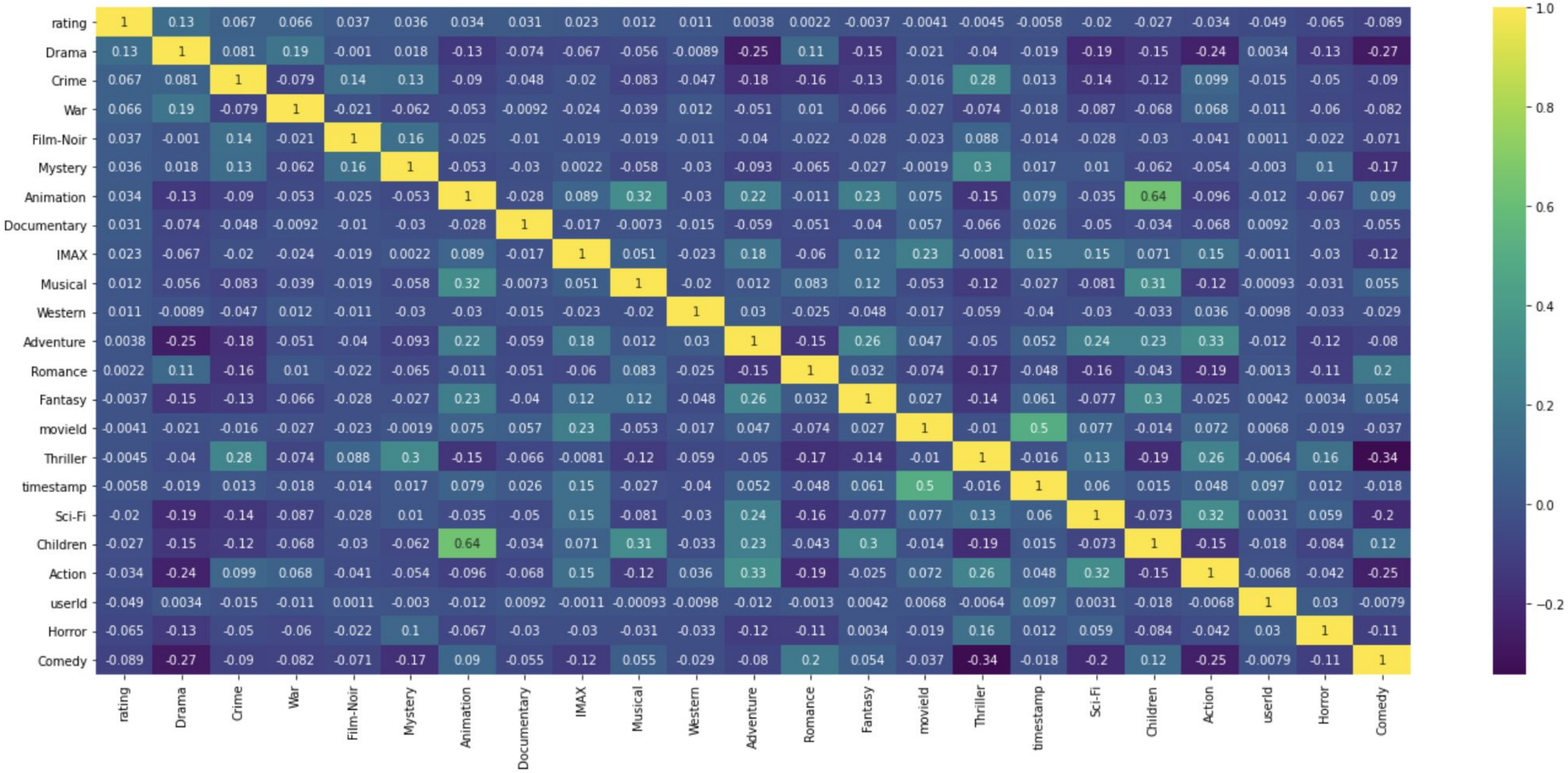
Number of ratings in each bucket

Ratings	Count
4.0	7394710
3.0	5515668
5.0	4071135
3.5	3404360
4.5	2373550
2.0	1850627
2.5	1373419
1.0	886233
0.5	442388
1.5	441354

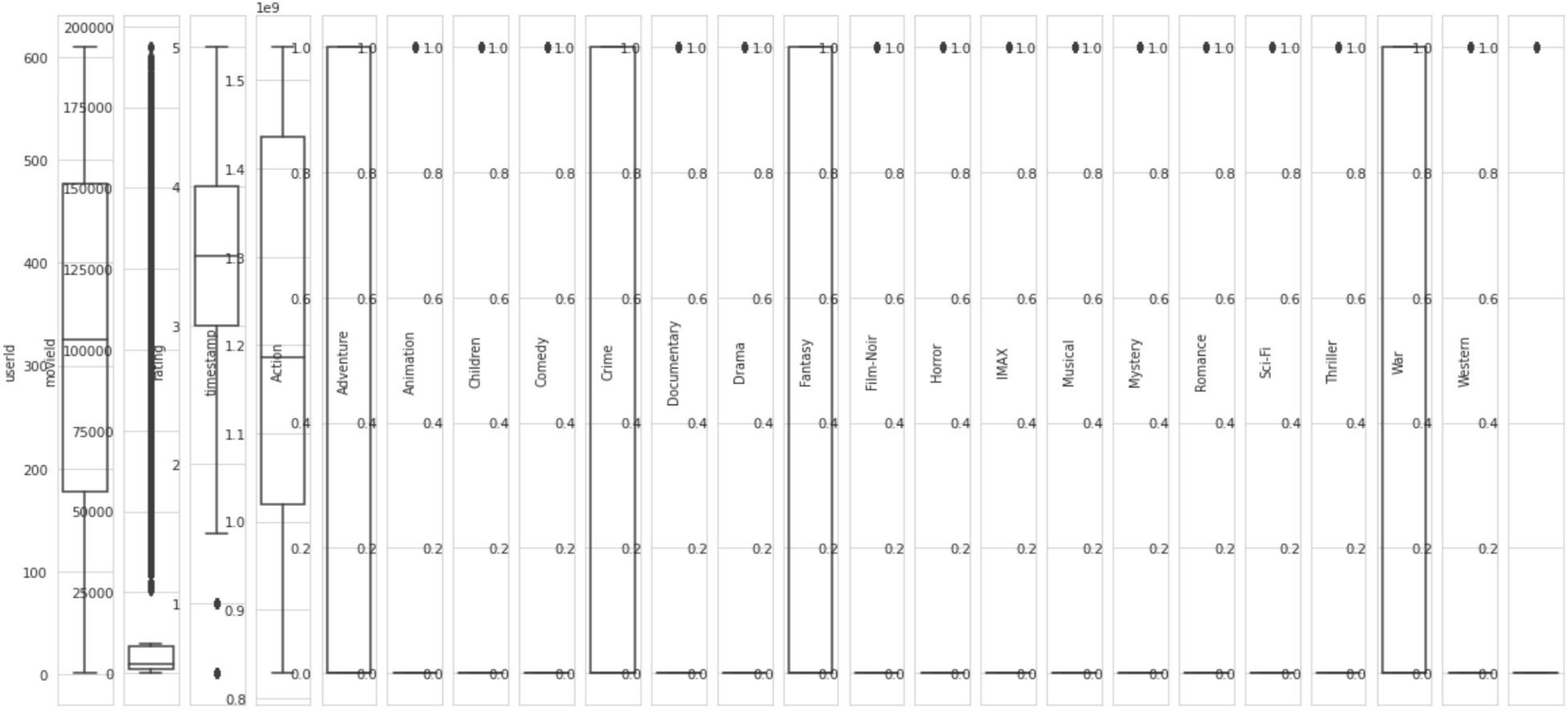
Top 10 movies based on the total number of ratings

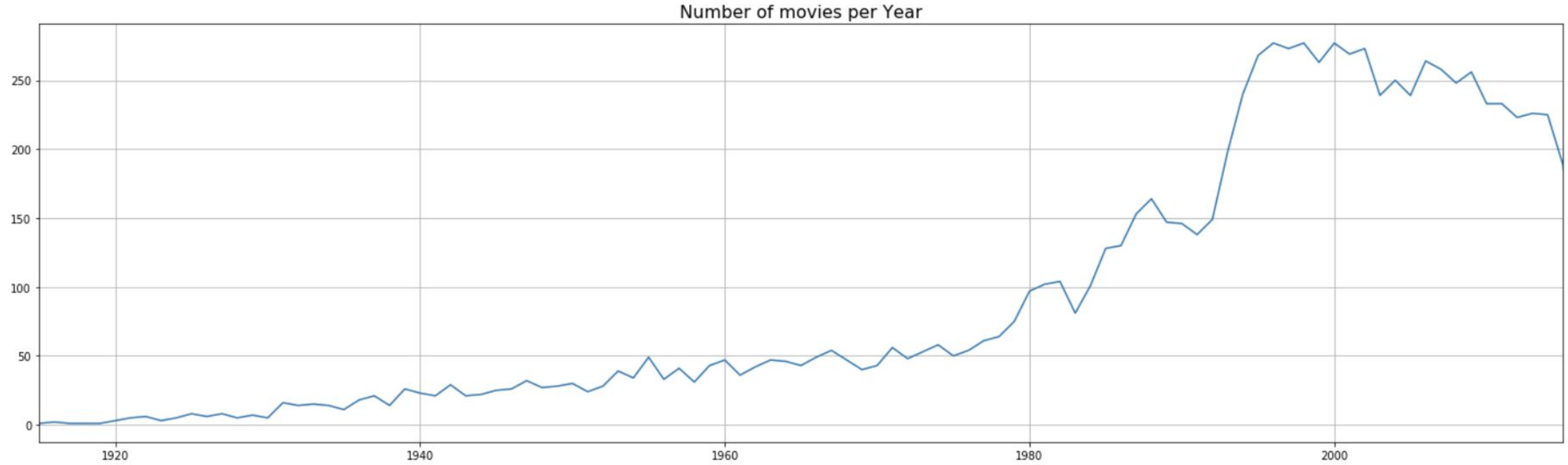
Movie title	Count
Shawshank Redemption, The (1994)	97999
Forrest Gump (1994)	97040
Pulp Fiction (1994)	92406
Silence of the Lambs, The (1991)	87899
Matrix, The (1999)	84545
Star Wars: Episode IV - A New Hope (1977)	81815
Jurassic Park (1993)	76451
Schindler's List (1993)	71516
Braveheart (1995)	68803
Toy Story (1995)	68469

Pearson correlation of the ratings.

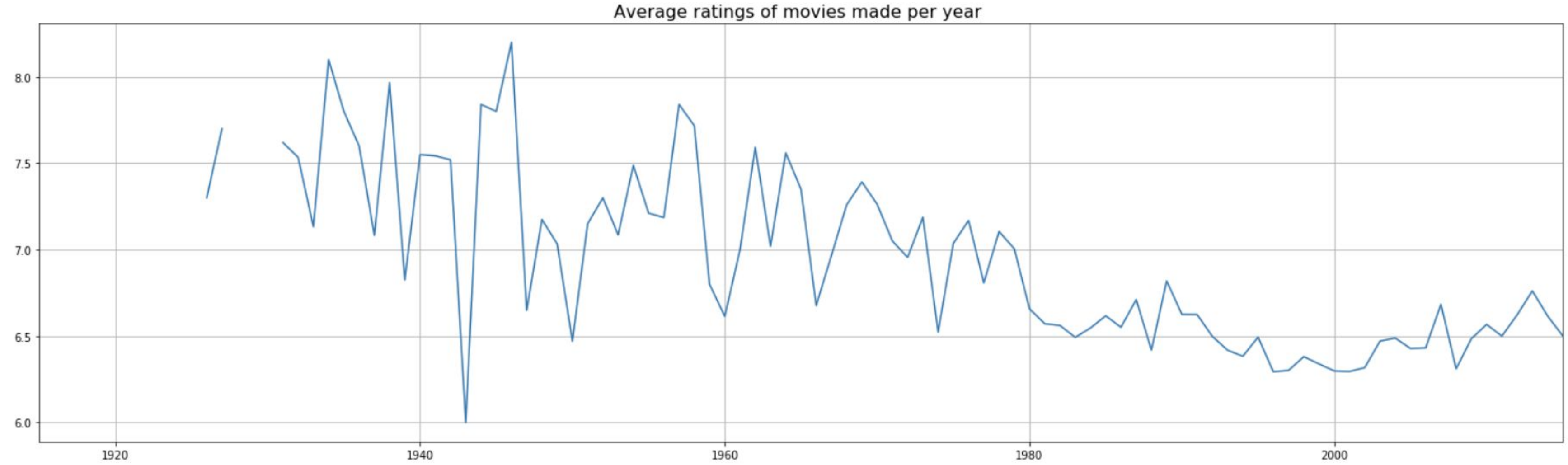


Significant genres that contribute to good ratings.

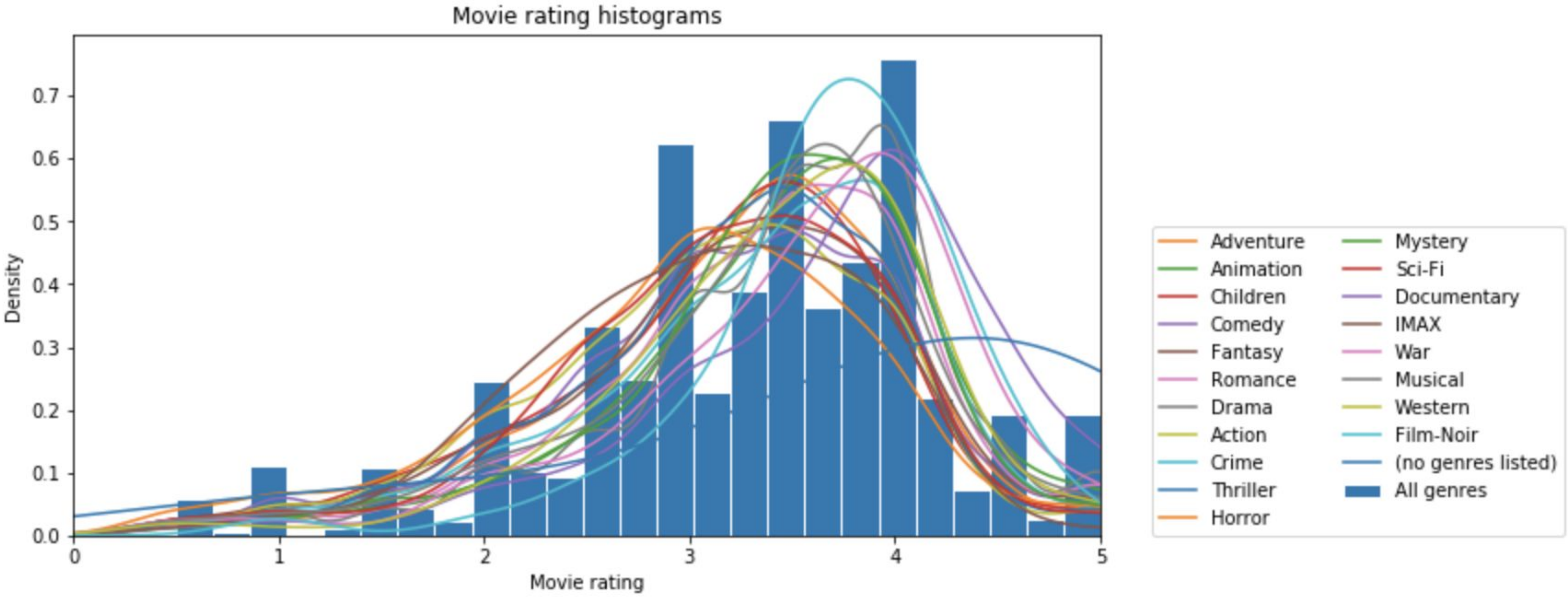




Number of movies per Year
0
50
100
150
200
250
1920
1940
1960
1980
2000



Average ratings of movies made per year
6.0
6.5
7.0
7.5
8.0
1920
1940
1960
1980
2000


Movie rating histograms
Density
Movie rating
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
1
2
3
4
5
Adventure
Animation
Children
Comedy
Fantasy
Romance
Drama
Action
Crime
Thriller
Horror
Mystery
Sci-Fi
Documentary
IMAX
War
Musical
Western
Film-Noir
(no genres listed)
All genres

PROBLEM STATEMENT

Movie Recommendations

- Top N recommendations for each user
  - Input : User ID
  - Output : Top N movies based on decreasing order of predicted ratings

- Evaluation
  - Relevance
    - Precision
    - Recall
    - F Measure
  - Prediction Accuracy
    - RMSE
    - MAE

APPROACH

Ensemble Learning

- Ensemble learning is an approach that combines different predictive models in order to improve the results obtained by individual models.
- Good balance between bias and variance
- Reduce Noise and avoid overfitting
- We can create hybrid model of Collaborative as well as content based approach.

Ensemble Rating = Mean(Ratings from different recommendation model)

Content Based Approach

- list of genres for a movie as content.
- created features and vectorized each movie using TF-IDF.
- Found out cosine similarity between movies user has already rated and movies we need to be recommended
- For predicting rating for the movie i (movie_i), we computed the weighted average rating with all the movies that the user has rated(movie_u)
- Restrict just with the movie with positive cosine similarity.
- If no positive cosine similarity movie available we consider mean of movie_u as the rating for movie_i

Collaborative Filtering - Neural Network

- Used the two embedding layers to represent the users and the movies as our input.
- The interactions (rating matrix) is the target variable.
- Merge both the layer using concatenation layer.
- We then try to minimize the loss via back-propagation and optimize the embedding layers

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
Movie (InputLayer)              (None, 1)            0
__________________________________________________________________________________________________
User (InputLayer)               (None, 1)            0
__________________________________________________________________________________________________
Movie-Embedding (Embedding)     (None, 1, 8)         431120      Movie[0][0]
__________________________________________________________________________________________________
User-Embedding (Embedding)      (None, 1, 5)         1416145     User[0][0]
__________________________________________________________________________________________________
FlattenMovies (Flatten)         (None, 8)            0           Movie-Embedding[0][0]
__________________________________________________________________________________________________
FlattenUsers (Flatten)          (None, 5)            0           User-Embedding[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 8)            0           FlattenMovies[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 5)            0           FlattenUsers[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 13)           0           dropout_1[0][0]
                                                                 dropout_2[0][0]
__________________________________________________________________________________________________
FullyConnected-1 (Dense)        (None, 100)          1400        concatenate_1[0][0]
__________________________________________________________________________________________________
FullyConnected-2 (Dense)        (None, 50)           5050        FullyConnected-1[0][0]
__________________________________________________________________________________________________
FullyConnected-3 (Dense)        (None, 20)           1020        FullyConnected-2[0][0]
__________________________________________________________________________________________________
Activation (Dense)              (None, 1)            21          FullyConnected-3[0][0]
==================================================================================================
Total params: 1,854,756
Trainable params: 1,854,756
Non-trainable params: 0
```

Collaborative Filtering - SVD and ALS

- Singular Vector Decomposition (SVD):
    - a matrix decomposition technique
    - used for dimensionality reduction, noise reduction and also compression.
    - popular because any real matrix can be decomposed -> more stable than eigen decomposition.
    - A real matrix is decomposed into three matrices, say U, Sigma, V^T,
        - where U and V are composed of orthonormal columns
        - Sigma is a diagonal matrix with singular values, which help us deduce the most significant correlated properties.
- Alternating Least Squares(ALS) :
    - a matrix factorization technique.
    - runs in a parallel fashion,
    - 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.
    - simple, scalable and works on even sparse data.

Collaborative Filtering - NMF

- Discovers latent factors in utility matrix
- Maps users and movies to a k-dimensional concept space
- Intuitively : Clustering the columns of the utility matrix
- H gives the cluster membership, i.e., if H kj ≥ H ij for all ik, this suggests that the input data vj belongs to kth cluster
- W gives the cluster centroids, i.e., the kth column gives the cluster centroid of kth cluster

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{bmatrix} \quad W = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_k \end{bmatrix} \quad H = \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_k \end{bmatrix}$$

$$x_i = \underbrace{[\boxed{w_{i1}}\ \boxed{w_{i2}}\ \dots \boxed{w_{ik}}]}_{w_i:\ \text{weights}} \times \overbrace{\begin{bmatrix} \boxed{h_1} \\ \boxed{h_2} \\ \dots \\ \boxed{h_k} \end{bmatrix}}^{\text{components}} = \sum_{j=1}^{k} w_{ij} \times h_i$$

Collaborative Filtering - Slope One

- Simple Linear predictor, improved efficiency relative to Linear Regression
- Precompute the average difference between the ratings of one item and another for users who rated both

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

$$\hat{r}_{ui} = \mu_u + \frac{1}{|R_i(u)|} \sum_{j \in R_i(u)} \text{dev}(i, j),$$

where $R_i(u)$ is the set of relevant items, i.e. the set of items j rated by u that also have at least one common user with i. $\text{dev}_{(}i, j)$ is defined as the average difference between the ratings of i and those of j:

$$\text{dev}(i, j) = \frac{1}{|U_{ij}|} \sum_{u \in U_{ij}} r_{ui} - r_{uj}$$

Collaborative Filtering - KNN with Baseline

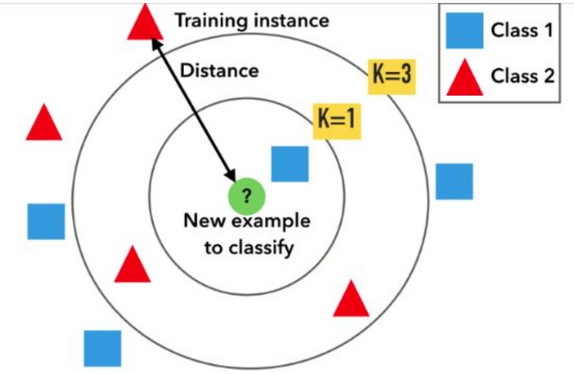
- KNN relies on item feature similarity
- Returns top K nearest neighbor movies based on Pearson Correlation "distance" between the target movie and every other movie
- Baseline estimate added to KNN to model for biases in rating

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

$$\hat{r}_{ui} = b_{ui} + \frac{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - b_{vi})}{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v)}$$

or

$$\hat{r}_{ui} = b_{ui} + \frac{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j) \cdot (r_{uj} - b_{uj})}{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j)}$$



Collaborative Filtering - CoClustering

- 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 user cluster and movie cluster

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

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

RESULTS

1M DATASET

Model	RMSE	MAE	PRESICION	RECALL	FMeasure
NeuralNet	0.95	0.71	0.67	0.67	0.67
ContentBased	0.94	0.71	0.66	0.66	0.66
SVD	0.87	0.68	0.68	0.68	0.68
NMF	0.91	0.72	0.66	0.66	0.66
CoClustering	0.91	0.71	0.67	0.67	0.67
ALS	1.25	1.16	0.62	0.62	0.62
KNN	0.89	0.70	0.67	0.67	0.67
Ensemble	0.87	0.68	0.68	0.68	0.68

Full Latest(27M) Dataset

Model	RMSE	MAE	Precision	Recall	FMeasure
NeuralNet	0.90	0.66	0.83	0.83	0.83
SVD	0.78	0.59	0.85	0.85	0.85
NMF	0.87	0.67	0.82	0.82	0.82
CoClustering	0.89	0.69	0.83	0.83	0.83
ALS	1.29	1.01	0.77	0.77	0.77
Ensemble	0.80	0.61	0.84	0.84	0.84

CONCLUSION

CONCLUSION

- Ensemble techniques balanced the bias variance trade-off and provide better result then base learner.

- Using ensemble learning we can combine different types of algorithms such as Content based, Collaborative filtering etc.

FUTURE POSSIBILITIES

- We can improve our model by using weighted average of the rating rather than taking simple average.

- Our result might improve more if we incorporate learning to rank based approached in base model.

REFERENCES

1. N. S. Altman. 1992. An Introduction to Kernel and Nearest-Neighbor Non-parametric Regression.e American Statistician46, 3 (1992), 175–185. hp://www.jstor.org/stable/2685209
2. Luis M. [de Campos], Juan M. Fern ́andez-Luna, Juan F. Huete, and Miguel A.Rueda-Morales. 2010. Combining content-based and collaborative recommen-dations: A hybrid approach based on Bayesian networks.International Journalof Approximate Reasoning51, 7 (2010), 785 – 799.DOI:hp://dx.doi.org/hps://doi.org/10.1016/j.ijar.2010.04.001
3. Chris Ding, Xiaofeng He, and Horst D. Simon. 2005. On the equivalence ofnonnegative matrix factorization and spectral clustering. Inin SIAM International Conference on Data Mining.
4. Arthur Fortes and Marcelo Manzato. 2014. Ensemble Learning in RecommenderSystems: Combining Multiple User Interactions for Ranking Personalization.WebMedia 2014 - Proceedings of the 20th Brazilian Symposium on Multimedia andthe Web(11 2014), 47–54.DOI:hp://dx.doi.org/10.1145/2664551.2664556
5. Simon Funk. 2006. Netix Update: Try is at Home. (2006). hps://sier.org/~simon/journal/20061211.html
6. Salma Ghoneim. 2019.Accuracy, Recall, Precision, F-ScoreSpeci-city, which to optimize on?(2019).hps://towardsdatascience.com/accuracy-recall-precision-f-score-specicity-which-to-optimize-on-867d3f11124
7. [7]G ́erard Govaert and Mohamed Nadif. 2008. Block clustering with Bernoullimixture models: Comparison of dierent approaches.Computational Statistics &Data Analysis52, 6 (February 2008), 3233–3245. hps://ideas.repec.org/a/eee/csdana/v52y2008i6p3233-3245.html
8. GroupLens. 2019. MovieLens. (2019). hps://grouplens.org/datasets/movielens/

[... Please refer the project report for the complete set of references]

THANK YOU

Q&A