

Recommender System ML Design

There are 610 unique users and 9742 unique movies with 100836 ratings. The ratings range between 0.5 and 5.

Feature Engineering

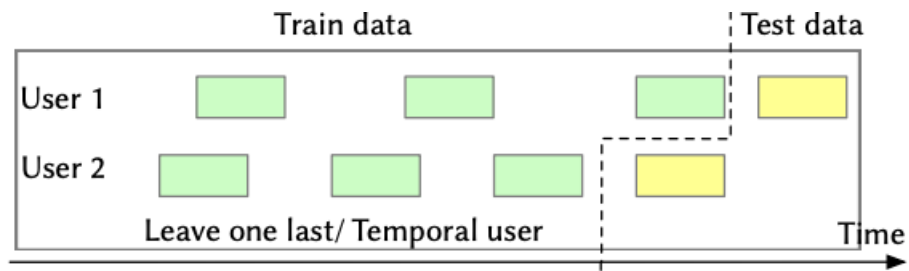
- **Movie Features** (439 features)
 - Release year, month. Days since release
 - Concatenate various Texts (Title, Actor Names, overview) to Text Embeddings using sentence transformers. The text encoder used is Sentence Transformer with embedding size 368
 - Average movie rating
 - Median movie rating
 - Movie genre dummy vectors
 - Spoken language
 - vote_average
 - vote_count
 - runtime
 - popularity
- **User features** (21 features)
 - Average user rating
 - Median user rating
 - (Normalized) number of times User watch a genre

Normalize ratings

The user ratings are skewed toward positive ratings, it would be better to scale the ratings between 0.5 and 1.

Train / Validation / Test Split

To create train and validation and test split, leave one our temporal splitting was used. For each use the latest rating is used in the Test split and the second latest is used for Validation split



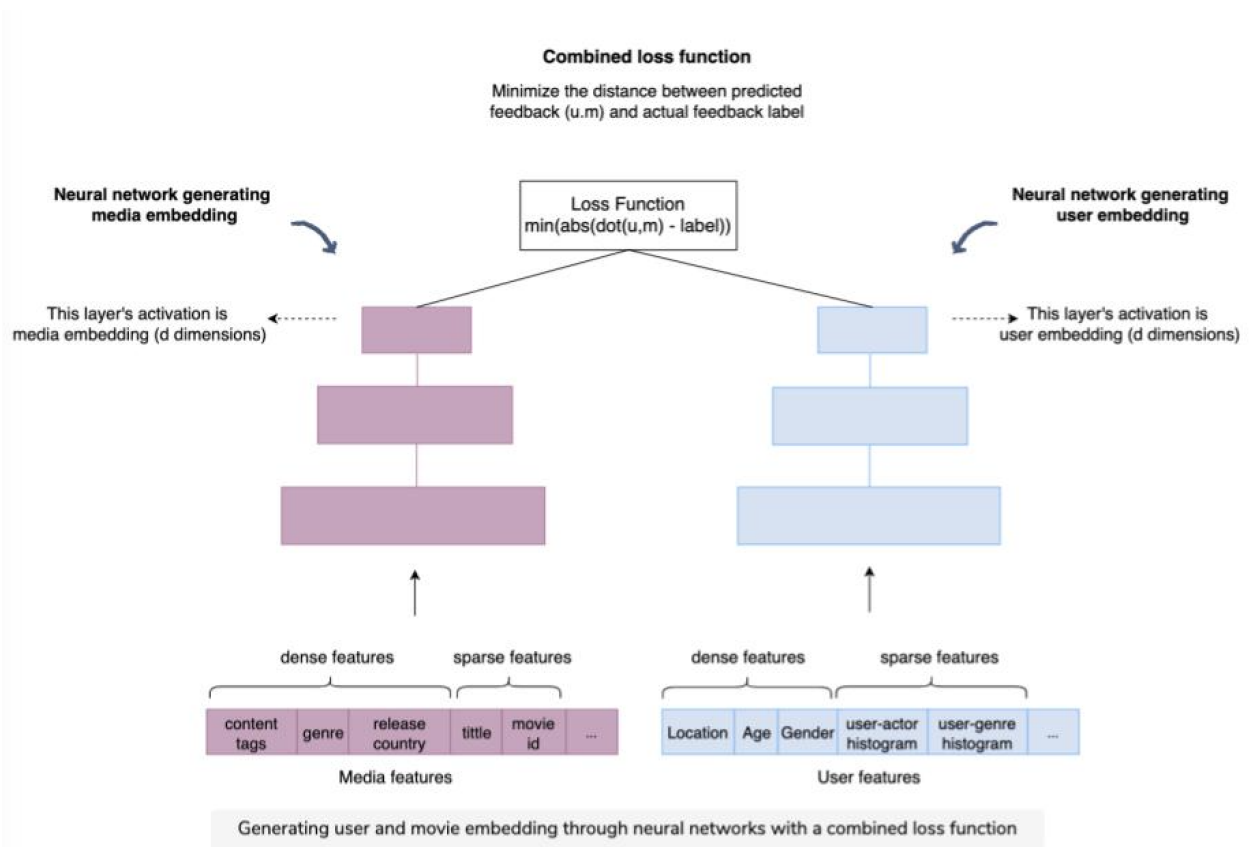
Partition	# of samples
Train	99616
Validation	610
Test	610

Loss function and Metric

- Loss: mean_absolute_error
- Metric: mean_absolute_percentage_error

Model Design

Used a neural network to predict the normalized rating. User features are sent into user tower and movie tower. Finally take the dot product of the user activations and media activations.



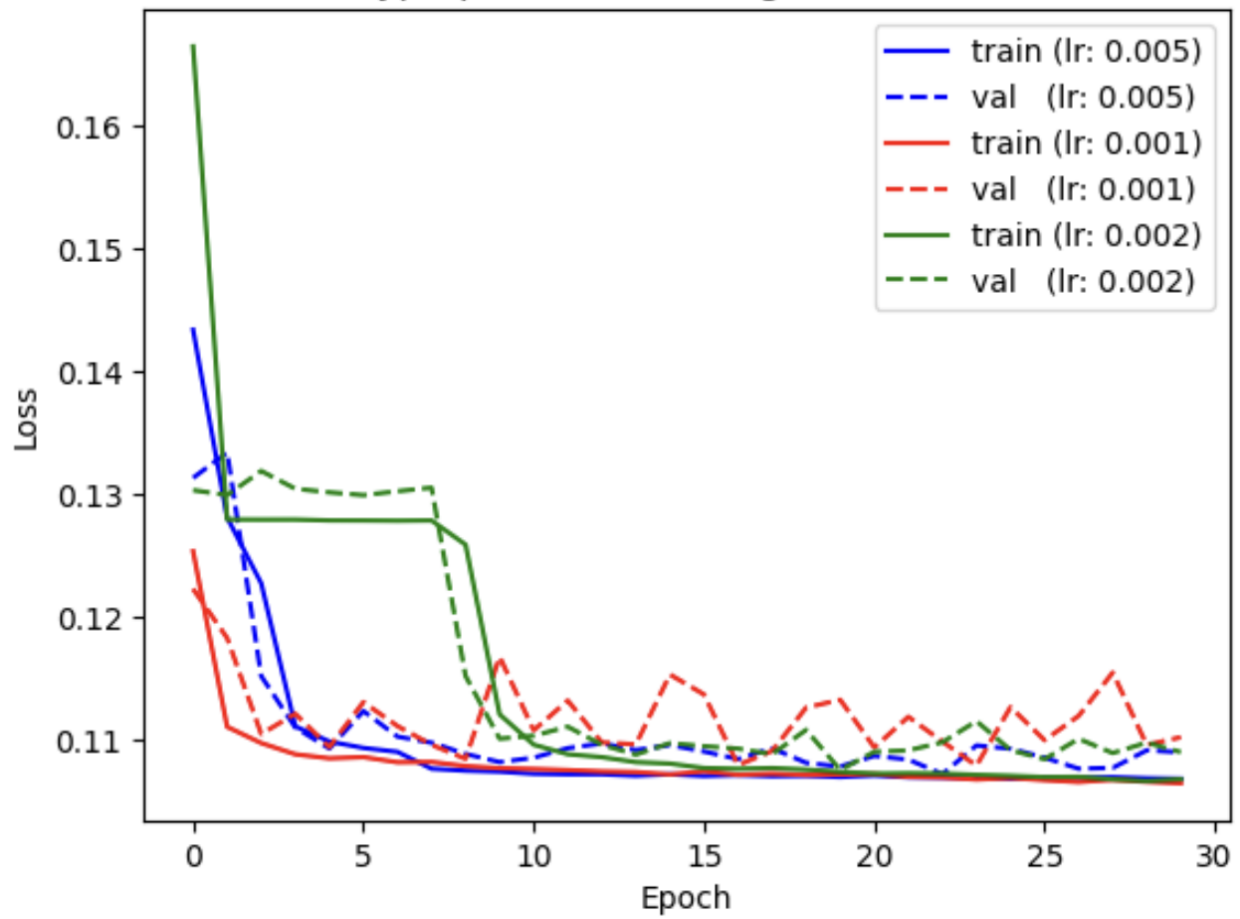
Model Summary

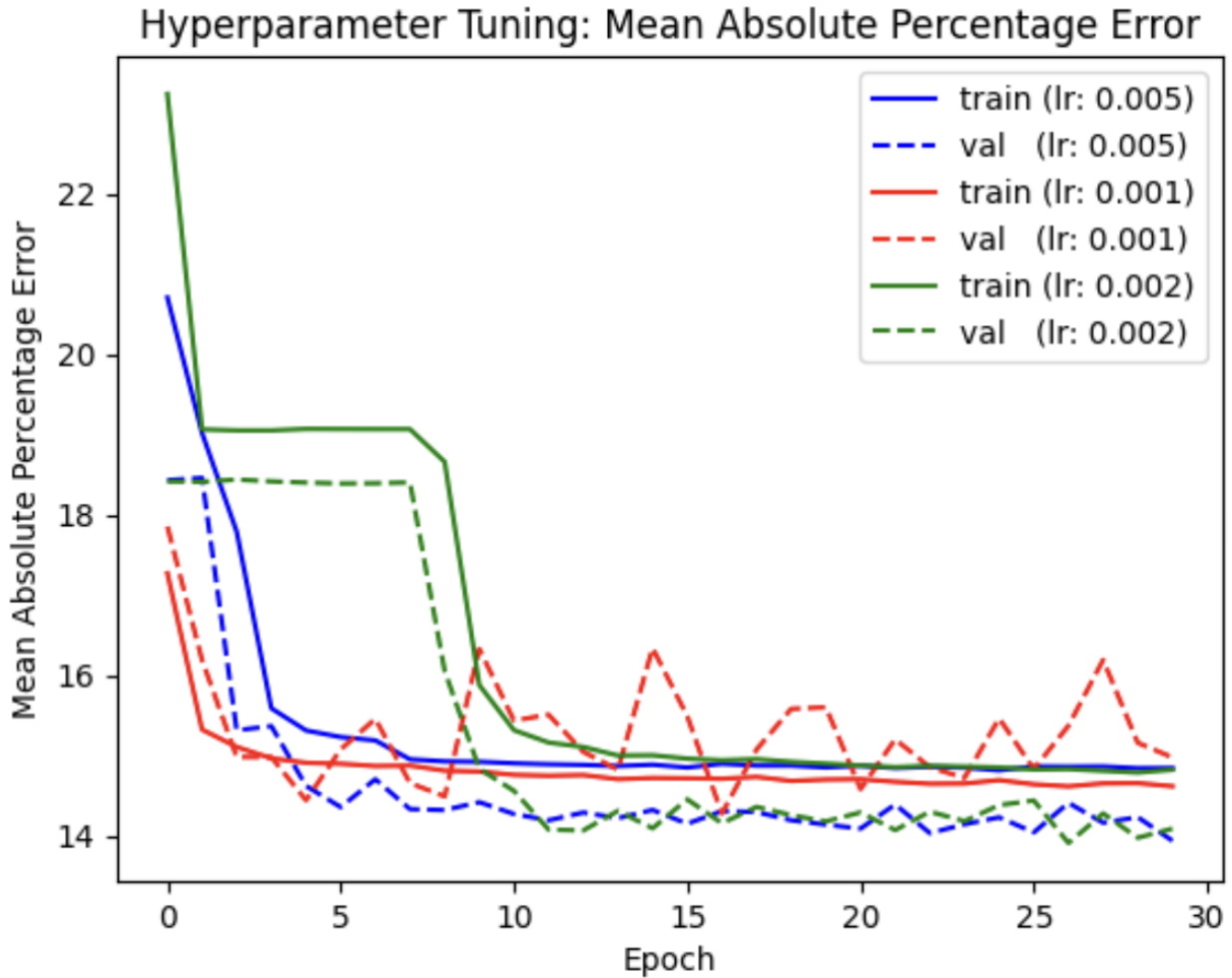
Layer (type)	Output Shape	Param #
movie_layer1 (Dense)	multiple	167600
movie_layer2 (Dense)	multiple	120300
movie_layer3 (Dense)	multiple	60200
movie_layer4 (Dense)	multiple	20100
dropout_45 (Dropout)	multiple	0
user_layer1 (Dense)	multiple	8800
user_layer2 (Dense)	multiple	120300
user_layer3 (Dense)	multiple	60200
user_layer4 (Dense)	multiple	20100
dropout_46 (Dropout)	multiple	0
dot_product (Dot)	multiple	0
y_out (Dense)	multiple	2
Total params: 577602 (2.20 MB)		
Trainable params: 577602 (2.20 MB)		
Non-trainable params: 0 (0.00 Byte)		

Training Curves and Model Performance

The model learning curves are not perfect at this time and need some work to improve the model.

Hyperparameter Tuning: Loss Curves





The train and validation partition are used to tune hyper-parameters (learning rate). Bases on the graphs above used learning rate 0.005. After finalizing the hyper-parameters, the performance is noted on the test set as seen below.

Partition	Mean Absolute Error Loss	Mean Absolute Percentage Error
Train	0.1282	19.0320
Val	0.1296	18.49
Test	0.133	19.3

Final Model Training

Finally model is trained on train and validation splits combined and validated against test set to save the best checkpoint.

Model Recommendation for UserId

- Compute model predictions for the input user id and all the movies that the user has not watch before.

- recommend the movies with the highest prediction scores

Future Work

- Scrape IMDB data. Features like movie director should be impact the user rating.
- Incorporate collaborative filtering technique into the model architecture.