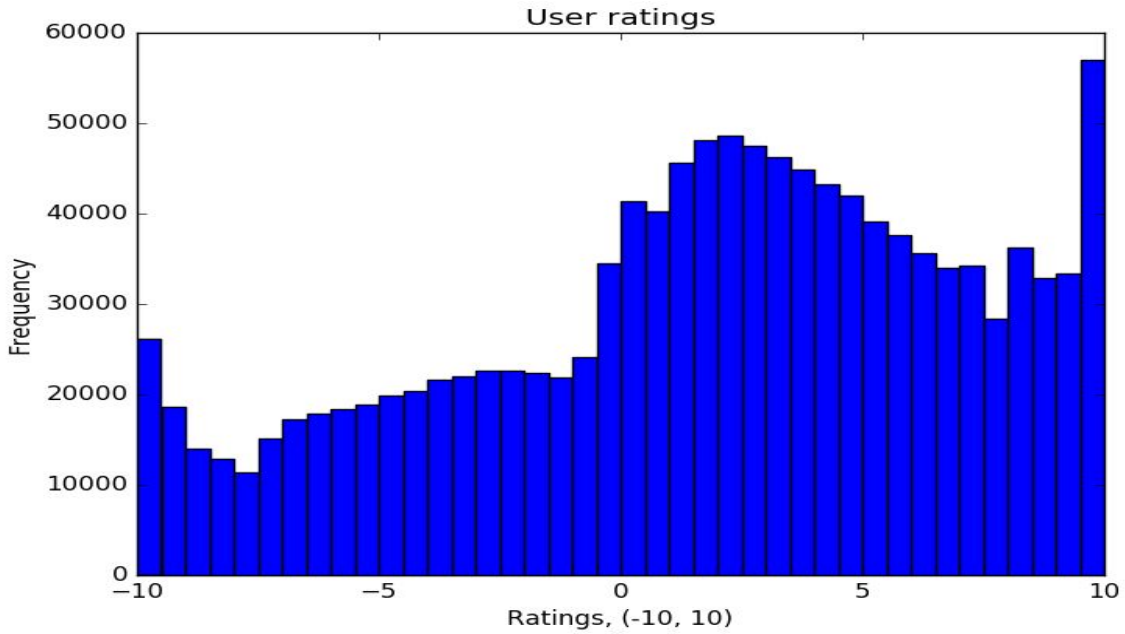
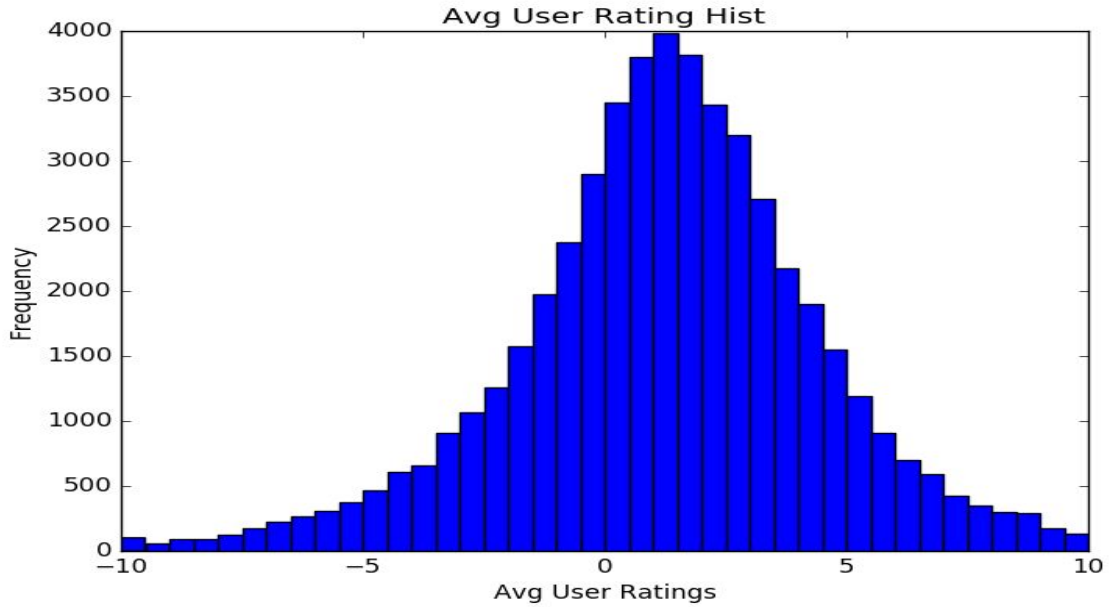
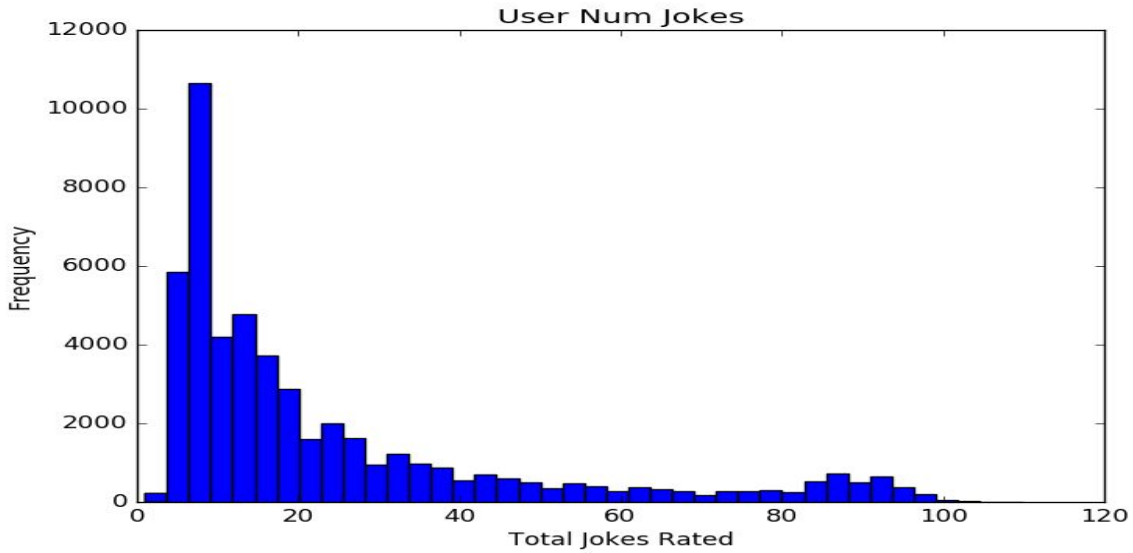
Steps and Findings

EDA:

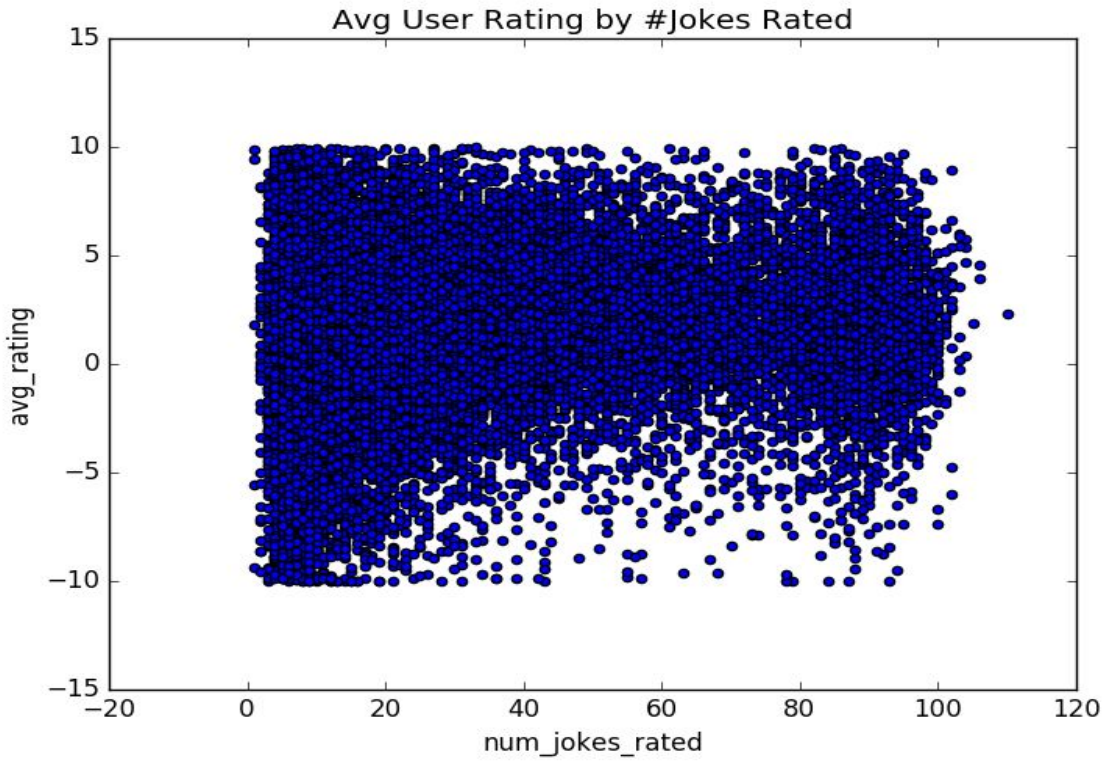
- Total 50181 unique user_id
- Total 141 unique joke_id
- Ratings are between -10 to 10, and the average rating is 1.700, so it looks like people generally like the jokes and rate them positively.







Average number of jokes rated: 24.03
Mode number of jokes rated: 7 jokes
Average rating by user: 1.22
Average rating by user +100 jokes: 1.3296554487179488



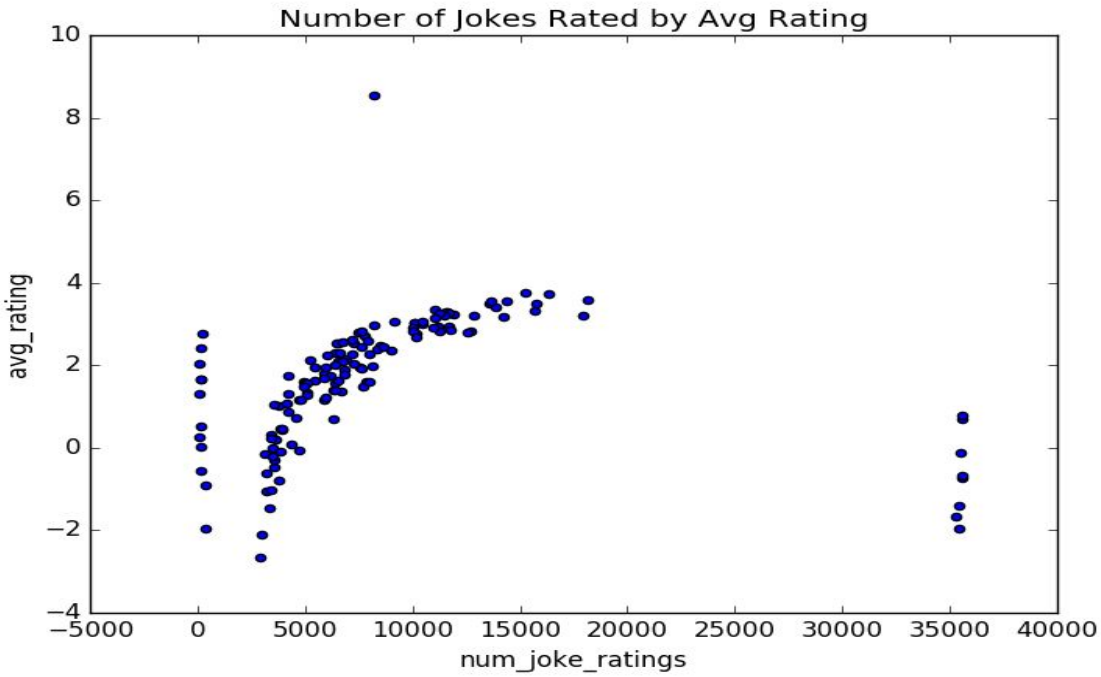
No one rated all of the jokes. The max number of jokes rated is 110 jokes, and there doesn't seem to be much of a trend in terms of people rating a lot of jokes, but rating them low - there are a few outliers in the bottom right corner where people rated 80 (for example) or more jokes and also rated them all low (-10).
With higher number of jokes rated, the average user ratings trended to average out and converge around the total joke average (1.77).

Lowest scored 5 jokes and number of ratings given:

joke_id	avg_rating	num_joke_ratings
142	-2.654188	2894
125	-2.108662	2982
6	-1.960360	406
8	-1.943980	35424
17	-1.657345	35290

Highest scored 5 jokes and number of ratings given:

joke_id	avg_rating	num_joke_ratings
1	8.529491	8206
106	3.750912	15274
54	3.721022	16311
90	3.593007	18131
36	3.561569	14359



Model:

Factorization machine to model using linear_side_features=True performs better in RMSE with longer training time

Search for optimal number of features and lambda (generalization term)

```
best params by rmse:
   {'item_id': 'joke_id',
    'linear_regularization': 1e-07,
    'max_iterations': 50,
    'num_factors': 8,
    'regularization': 1e-09,
    'side_data_factorization': False,
    'solver': 'als',
    'target': 'rating',
    'user_id': 'user_id'}
```

The latent features are:

TOP JOKES FOR FACTOR 1:

joke_id:1
joke_id:106
joke_id: 54
joke_id: 67
joke_id: 78

TOP JOKES FOR FACTOR 2:

joke_id: 33
joke_id:142
joke_id: 26
joke_id: 46
joke_id: 89

TOP JOKES FOR FACTOR 3:

joke_id: 13
joke_id: 57
joke_id: 54
joke_id: 90
joke_id: 58

TOP JOKES FOR FACTOR 4:

joke_id: 9

joke_id: 124
joke_id: 5
joke_id: 76
joke_id: 139

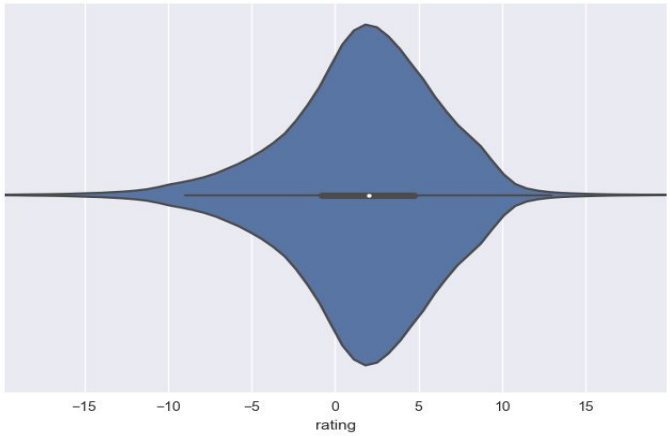
What are the top topics for these factors?

Prediction:

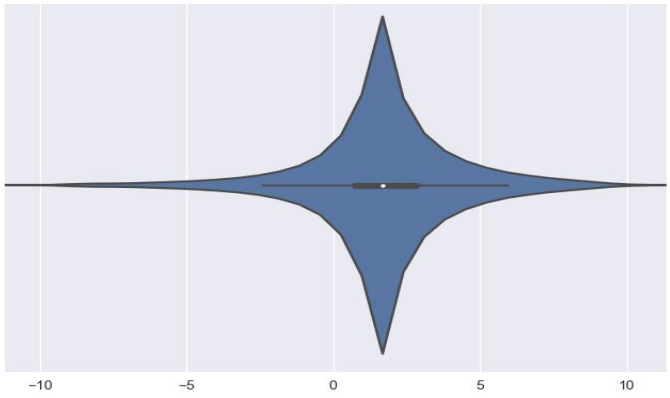
Below is the average rating of the top seven jokes for each user. It is interesting that from joke rank 5 and below, the rating is negative.

rank	
1	3.611376
2	2.086527
3	1.165749
4	0.488488
5	-0.045136
6	-0.496875
7	-0.893967

Below is the violin plot for test ratings. The majority ratings fall between -5 to 8 and the curve is slightly skewed to the left.



Below violin plot is our prediction of all unique users and jokes combination. In general it has similar result compare to the test data. However, most ratings are between 0 to 3.



Next steps:

Topic model jokes for top factors to understand the topics and which groups of users who liked/disliked these jokes the most.

Conduct A/B or multi-bandwidth test to verify the impact of the recommender and choose the optimal factorization approach (ranked vs non ranked) and collaborative filtering vs content based on small groups of test/control users and based on their reaction, we can keep tweaking our recommendation engine.

Even better, we can create a self-correcting engine to stratify/select audience for A/B testing and gather inputs from end users to auto update/correct our recommendation engine. Also, assigning topics to all jokes and randomly feed jokes from each category and ask the end users to rate the jokes will give us additional info for topic modelling and better predicting end users' interests.

Other demographic user data such as age, gender, nationality etc might help us find more insights and trend about users and best jokes to display.ik

In addition, we can ask unstructured inputs from end users such as reviews/feedbacks in text and conduct sentiment analysis and NLP.