# Joke Recommendation

Team 25

Amrithaa Seshadri, Bowen Sun, Pengpeng Sun, Savitha Srinivasan, Wei-Ru Lin

# Motivation

- ❑ Smart assistants like Siri, Google, and Alexa are pushing to recommend jokes based on user preferences
- ❑ Humor can be an essential tool to reduce stress/tension
- ❑ Understanding humor makes computers more ‘user friendly’

# Why is it difficult?

- ❑ What is a ‘good’ joke?
- ❑ Difficult to formalize humor
- ❑ Highly subjective

# Dataset

- Jester dataset for collaborative filtering research, AUTOLab, UC Berkeley
- 4.1 million ratings of 100 jokes
- 73,421 users, rating scale: -10.00 to 10.00

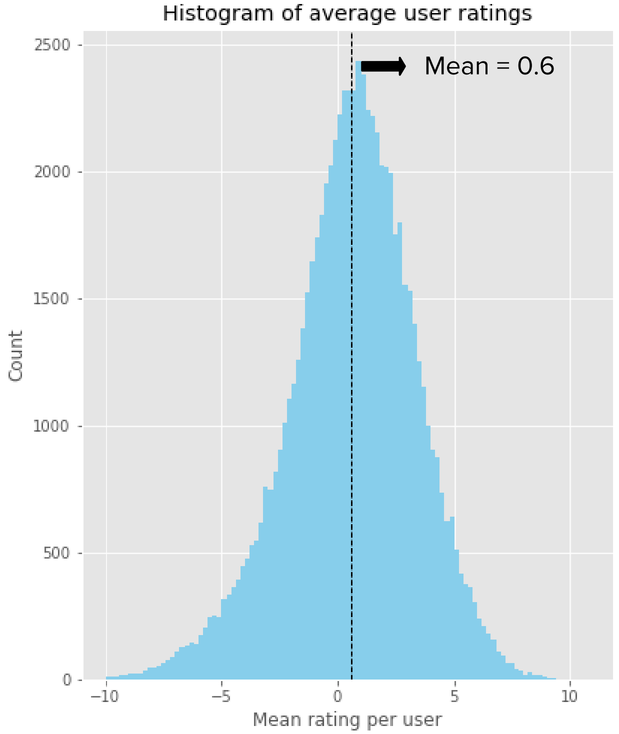
| User ID | joke1 | joke2 | joke3 | joke4 | joke5 | joke6 | joke7 | joke8 | joke9 | joke10 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -7.82 | 8.79 | -9.66 | -8.16 | -7.52 | -8.50 | -9.85 | 4.17 | -8.98 | -4.76 | ... |
| 1 | 4.08 | -0.29 | 6.36 | 4.37 | -2.38 | -9.66 | -0.73 | -5.34 | 8.88 | 9.22 | ... |
| 2 | NaN | NaN | NaN | NaN | 9.03 | 9.27 | 9.03 | 9.27 | NaN | NaN | ... |
| 3 | NaN | 8.35 | NaN | NaN | 1.80 | 8.16 | -2.82 | 6.21 | NaN | 1.84 | ... |
| 4 | 8.50 | 4.61 | -4.17 | -5.39 | 1.36 | 1.60 | 7.04 | 4.61 | -0.44 | 5.73 | ... |
| 5 | -6.17 | -3.54 | 0.44 | -8.50 | -7.09 | -4.32 | -8.69 | -0.87 | -6.65 | -1.80 | ... |

Snippet of cleaned data

A horse walks into a bar. The bartender asks: "So, why the long face?"

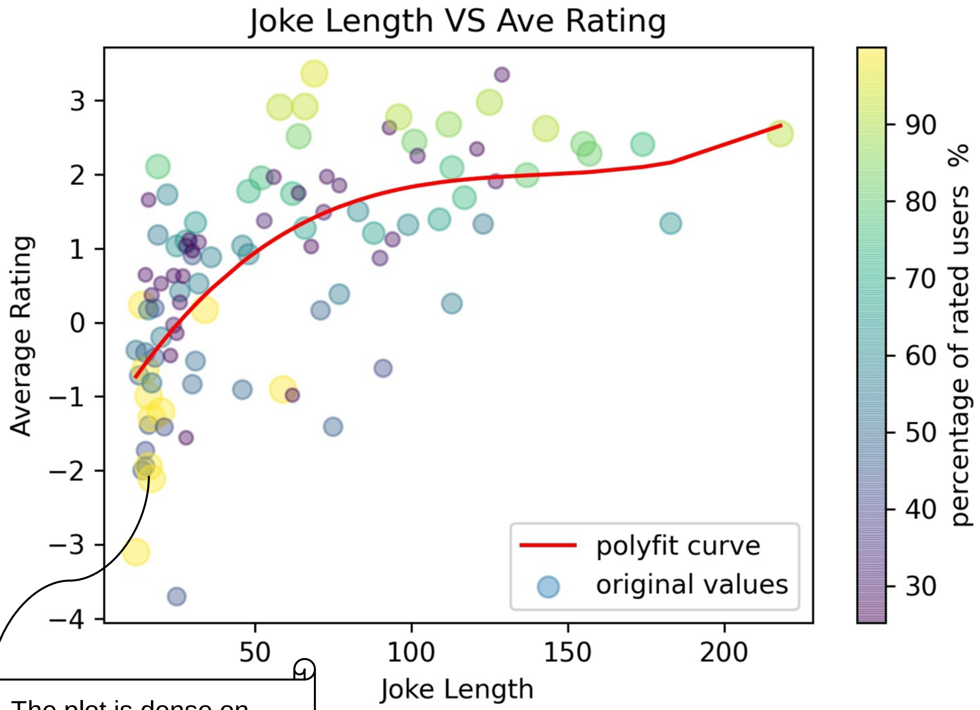
Sample joke

# Insights from the Data


User average ratings are not zero centered - need to be normalized

# Insights from the Data




The plot is dense on the left, indicating that most jokes are short!

Bubble plot showing the effect of the length of a joke on its rating

Size of bubbles represent the number of users who rated the joke

# Content Information from Data

- Jester was primarily developed for collaborative filtering research
- It therefore consists of numerical ratings only, useful for user-user recommendations
- To explore content-based recommendations
  - We extracted keywords based on TF-IDF scores
  - Categorized jokes into one of 6 genres


Word cloud of text from jokes

# Term Frequency-Inverse Document Frequency (TF-IDF)

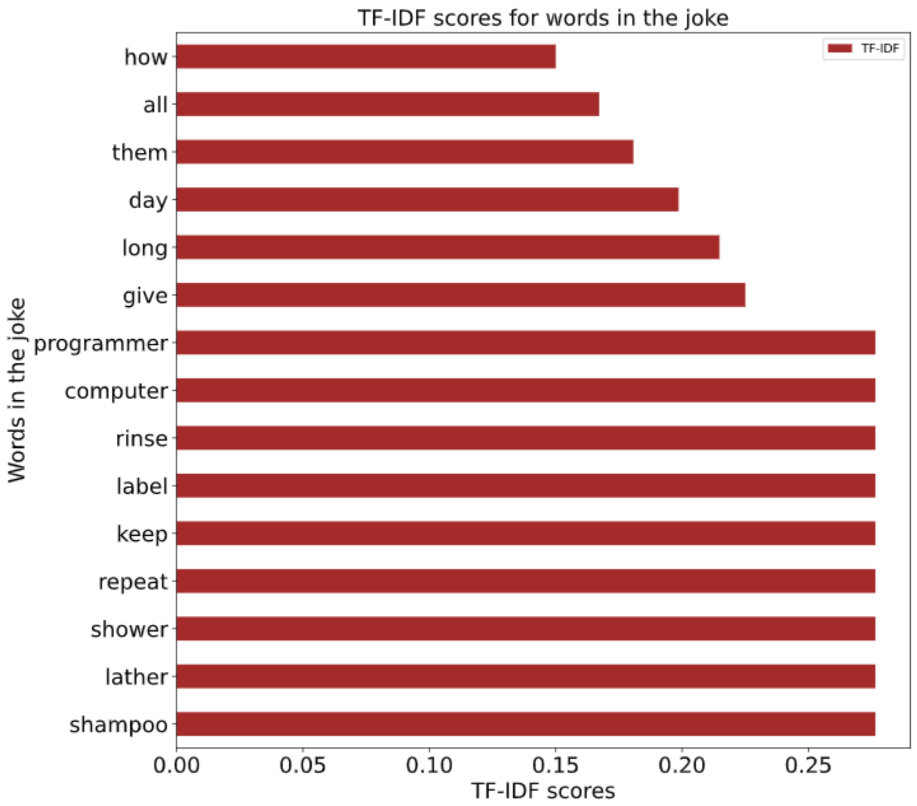
- ❑ Statistical measure that reflects the ‘relevance’ of a word to a document
- ❑ TF: Frequency of the word in the document
- ❑ IDF: Inverse frequency of the word across a set of documents
- ❑ TF-IDF = TF * IDF

# TF-IDF Scores for a Joke

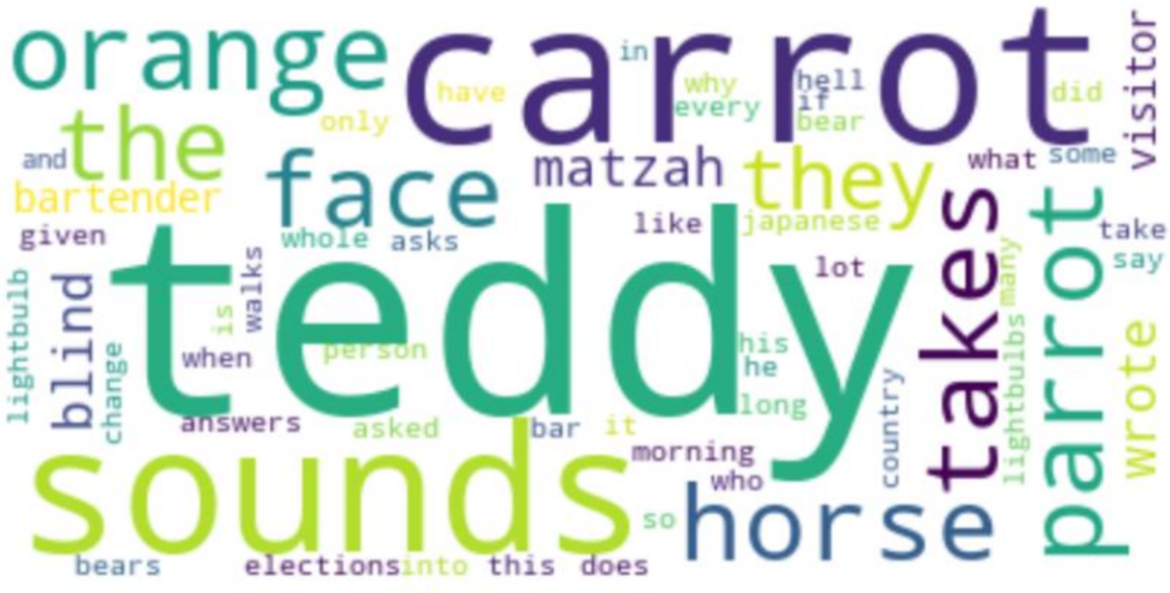
Sample TF-IDF scores for a joke:

Joke 81:

Q: How do you keep a computer programmer in the shower all day long?
A: Give them a shampoo with a label that says "rinse, lather, repeat."

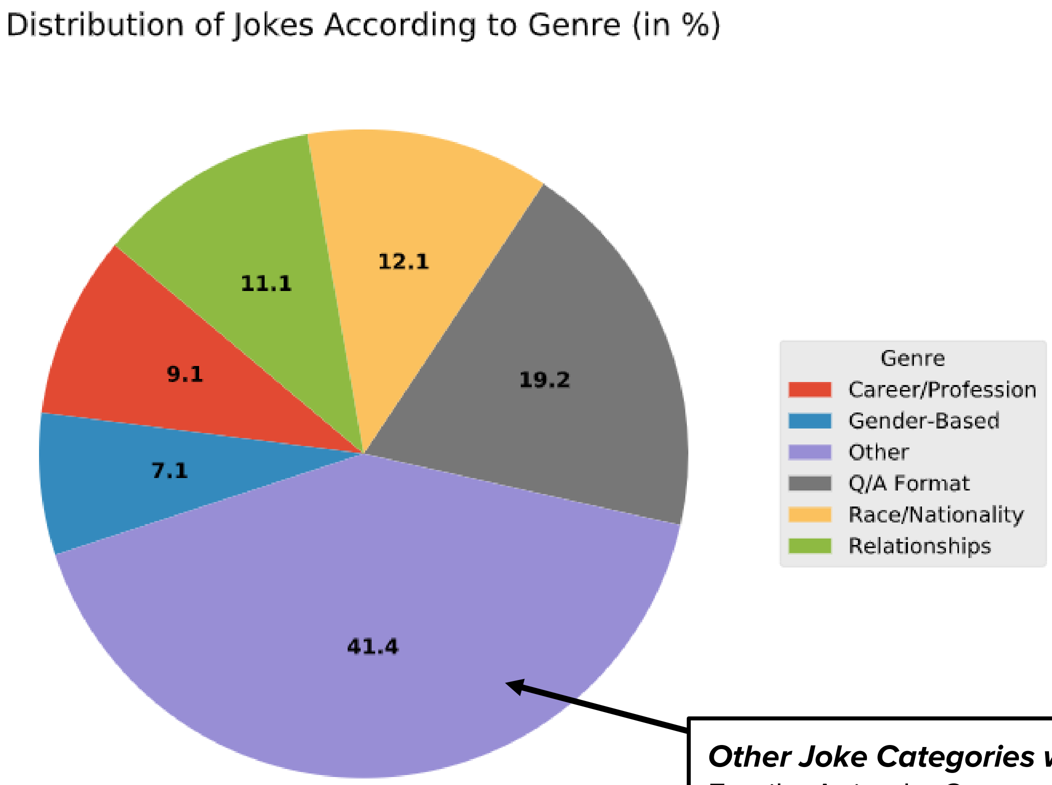


# Keywords in the Five Highest Rated Jokes




Word cloud of keywords in five highest rated jokes

# Content Information from Data

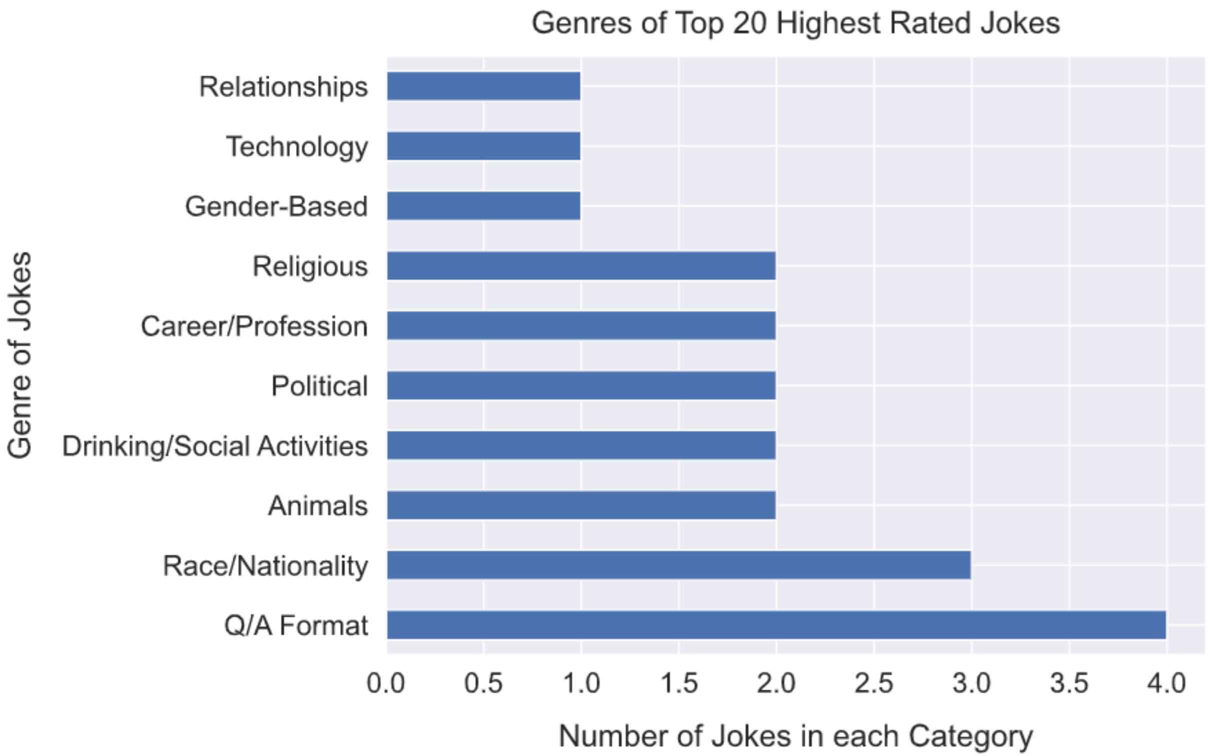



**TOP GENRES**

- Q/A FORMAT
- Race/ Nationality
- Relationships

***Other Joke Categories with 1-2 jokes each:***
Family, Animals, Current Affairs, Politics etc

# Content Information from Data

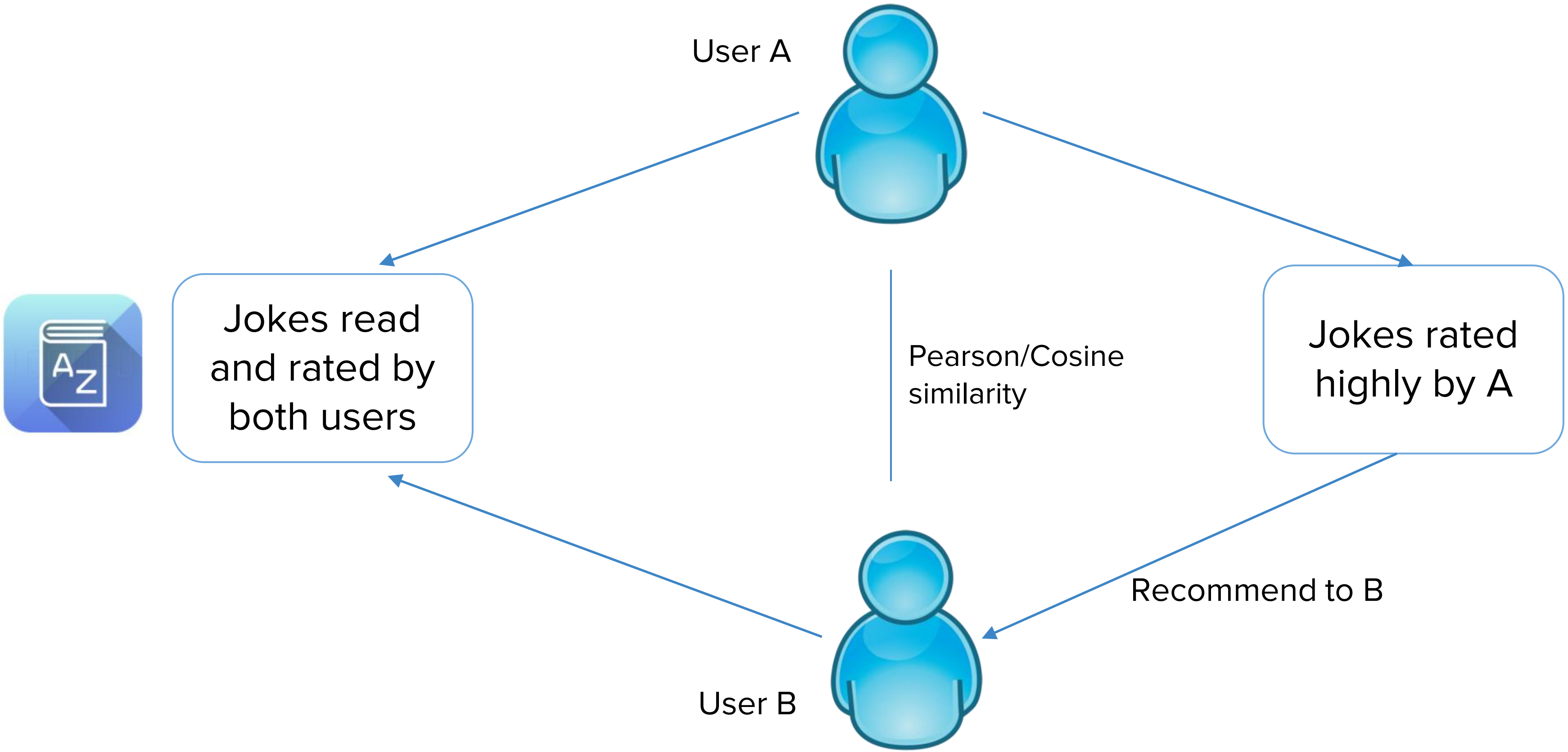



JOKES ON "RACE/NATIONALITY" ARE LIKED BY MOST USERS

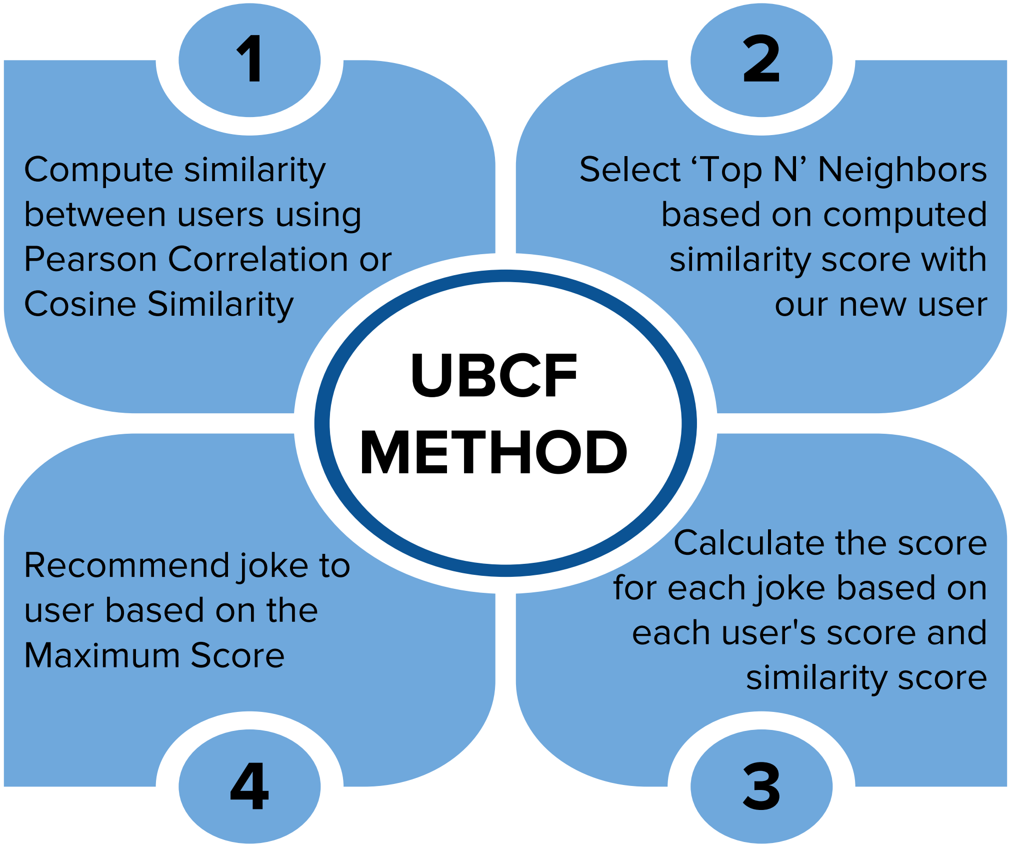
JOKES IN Q/A FORMAT ARE ALSO HIGHLY RATED

# Let’s Start Recommending Jokes!

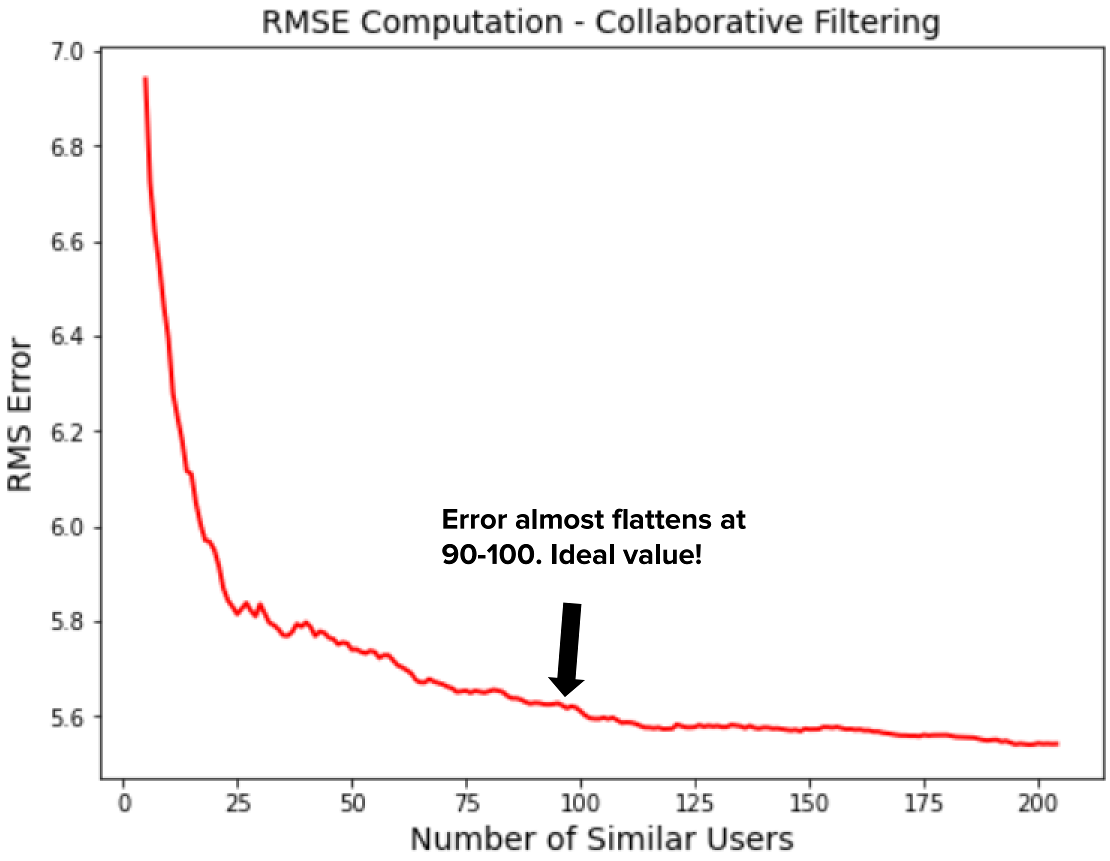
# User-based Collaborative Filtering (UBCF)



# User-based Collaborative Filtering (UBCF)



# UBCF - Finding Optimum Number of Similar Users

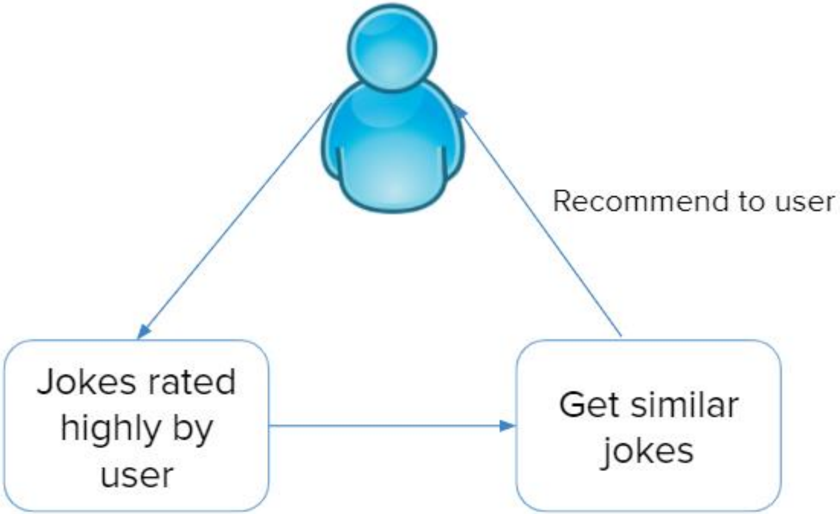



Finding 'optimum' value of similar users to use for Collaborative Filtering

Calculating RMS Error by removing part of the dataset and computing the error between the actual ranking and the calculated ranking.

**"Optimal" Number of Similar Users = 90-100**

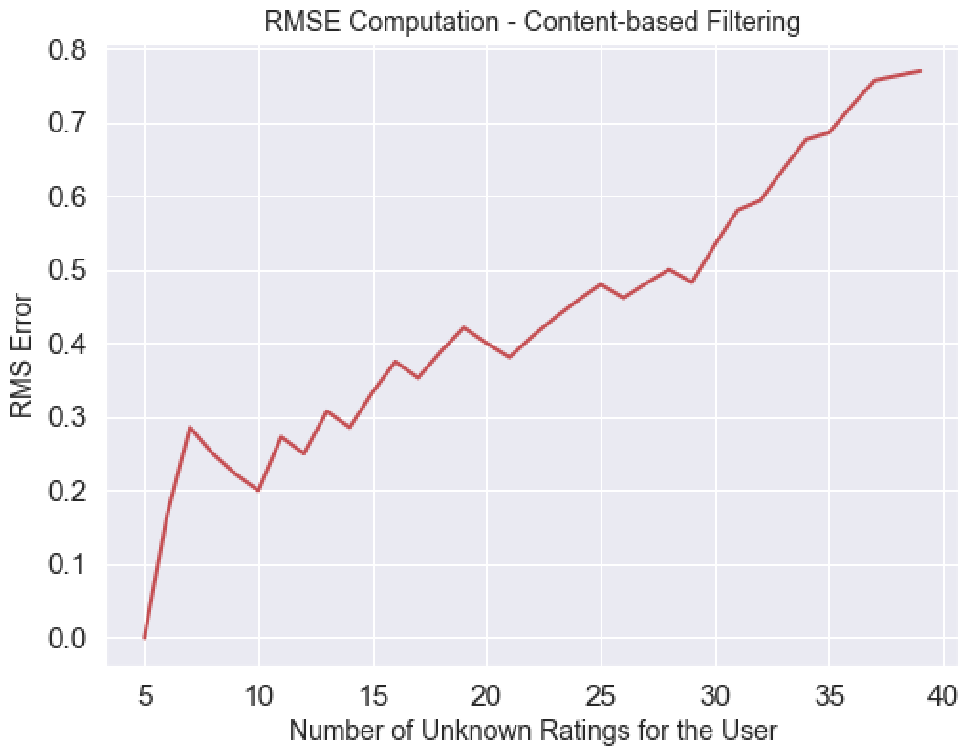
# Content-based Filtering



# Content-based Filtering

1. Find top N key words using TF-IDF scores for the jokes in the dataset
2. Create a profile of the key words of the jokes rated by our user who needs recommendations
3. Compute Cosine Similarity between the unseen jokes and the rated jokes
4. Recommend new jokes based on distance between user profile and joke database

# Content-based Filtering: RMSE Computation




Calculating RMS Error by removing part of the dataset and computing the error between the actual ranking and the calculated ranking.

**The RMS Error is increasing with the number of unknowns, as expected**

# Results

# Ok Google, Tell Me a Joke...

FOR A NEW USER who hasn't rated any of the jokes

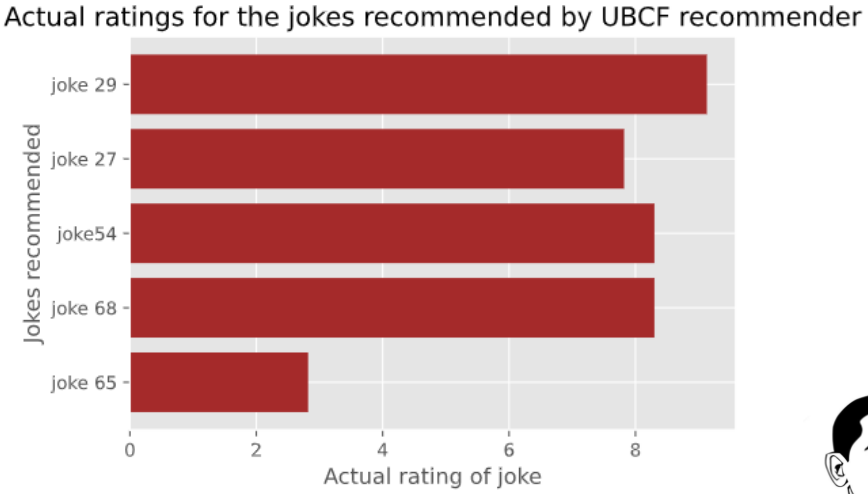
Our Recommender will randomly display one of the top 20 highest rated jokes!

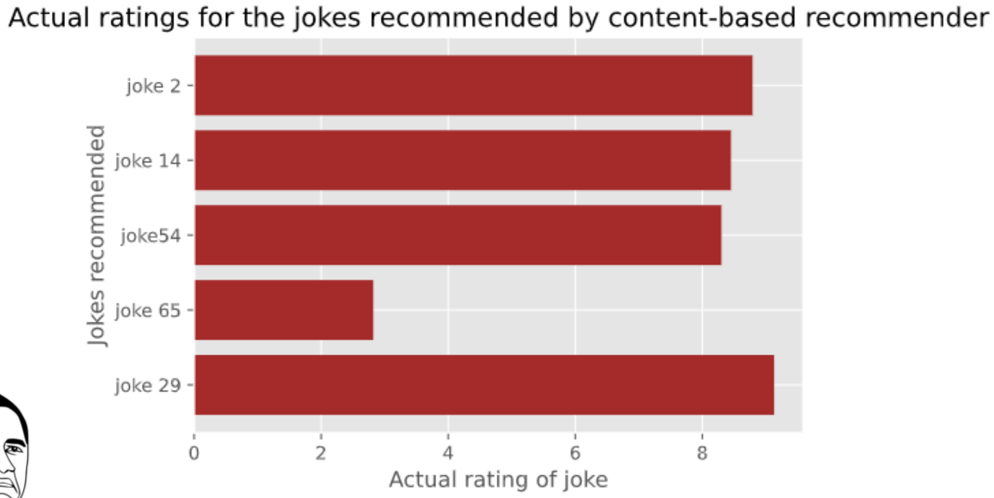
**Son:** "Mom, I got a role in a school play. I play a husband"

**Mother:** "Go back and tell them you want a speaking role"

# Results: UBCF vs Content-based

Sample results for a new user who has rated few jokes using UBCF and content-based recommenders:




# Conclusions and Limitations

- **Performance of UBCF and Content-based recommender is comparable, but since humor is subjective, a more effective content-based recommender would be ideal.** **Small Dataset:** Insufficient number of jokes (100) to build a good content-based recommender

- **Data Sparsity:** Missing ratings in the dataset can lead to inaccurate recommendations

# Future Scope

- ❑ Find datasets with more jokes and ratings to build a good content-based recommender system.
- ❑ Implement other neighbor selection techniques to improve the efficiency of UBCF
- ❑ Effectively combine multiple recommendation schemes to create the best one

THANK YOU!

QUESTIONS?