JOKE RECOMMENDATION SYSTEM

CMPE 256 PROJECT

BY

AMRUTA DHONDAGE

NAVNEET JAIN

SURBHI JAIN

GUIDED BY:-

DR. MAGDALINI ERINAKI

WHY WE NEED?

- CATEGORIZATION OF JOKES
- RECOMMENDATION OF JOKES FOR A USER
- FINDING TARGET AUDIENCE FOR A NEW JOKE

WHAT DATASET IS USED?

- Jester (UC Berkley)
- jester_dataset_1_1.zip: (3.9MB) Data from 24,983 users who have rated 36
- or more jokes, a matrix with dimensions 24983 X 101
- • Ratings are real values ranging from -10.00 to +10.00 (the value "99"
- corresponds to "null" = "not rated").
- • The first column gives the number of jokes rated by that user. The next 100
- columns give the ratings for jokes 01 - 100.
- • The text of the jokes can be downloaded here:
- jester_dataset_1_joke_texts.zip (92KB) / jester_dataset_2.zip (7.7MB)

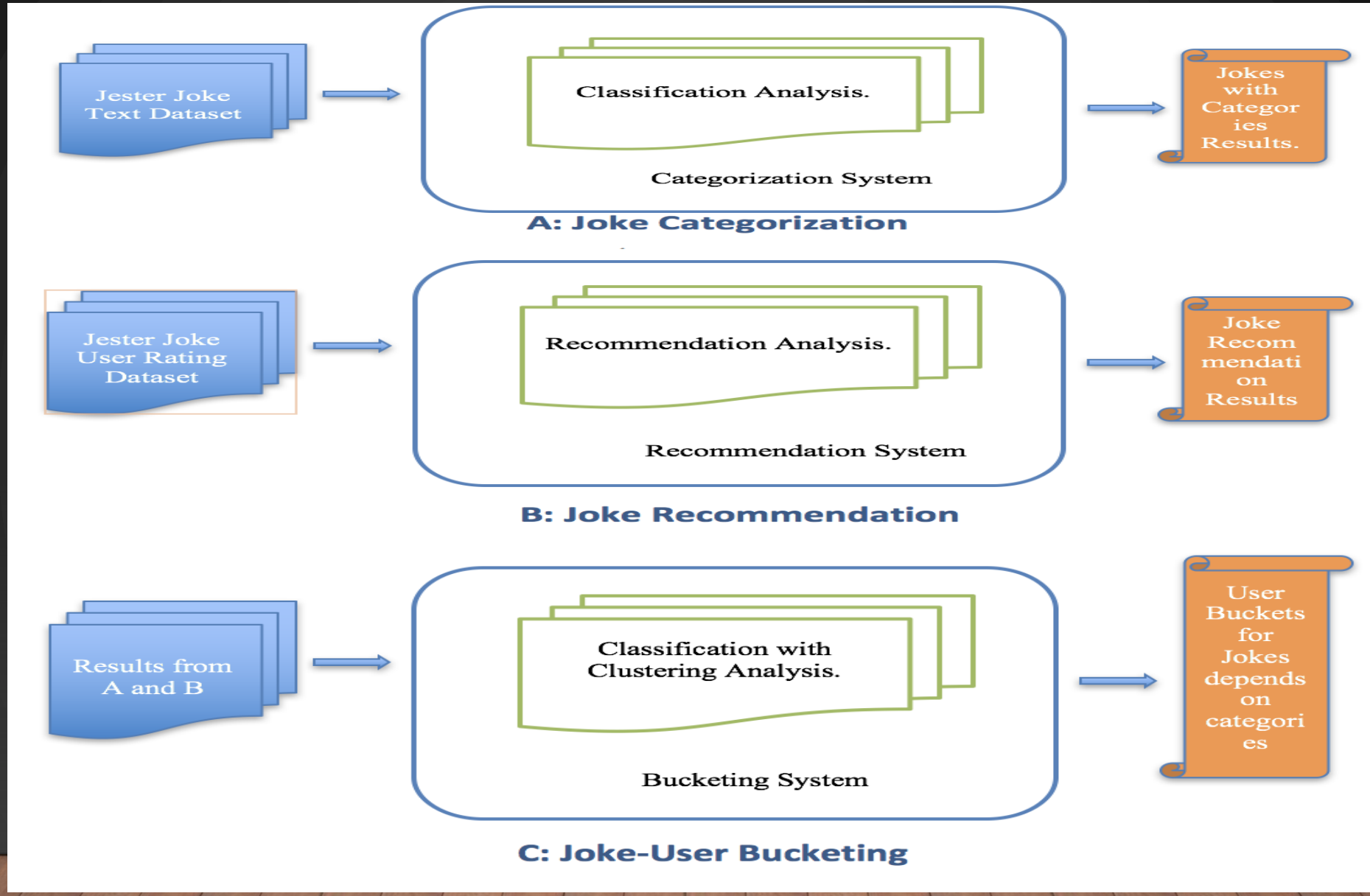
WHAT IS THE OBJECTIVE?

- Build a recommendation system which looks for user's preferences
- Perform categorization of jokes
- Mapping the users to nearest category of jokes which user might be most interested in.
- Finding the target audience for each new joke in the market. Like this new jokes will only be sent to the audience who are most likely to read them in comparison to sending to every user.

DATA PREPROCESSING

- The dataset set contained two types of data :- 1) User-Joke Ratings 2) Joke Text.
- From User-Joke rating dataset we formatted data into userid-jokeid-rating matrix. This dataset is then fed to recommendation system.
- For Joke-User Bucketing, we needed to have categorization for joke and user likes and dislikes for specific category. Data is processed for having a userid-jokecategory-jokeid-rating structure.
- Joke text dataset was raw text dataset. We processed it into jokeid and joketext structure. Joketext had many unwanted words and characters. We processed the text for following conditions to get the clean joke text.
  - Remove special characters such as html tags and obtain only words
  - Remove stop words

SYSTEM DESIGN



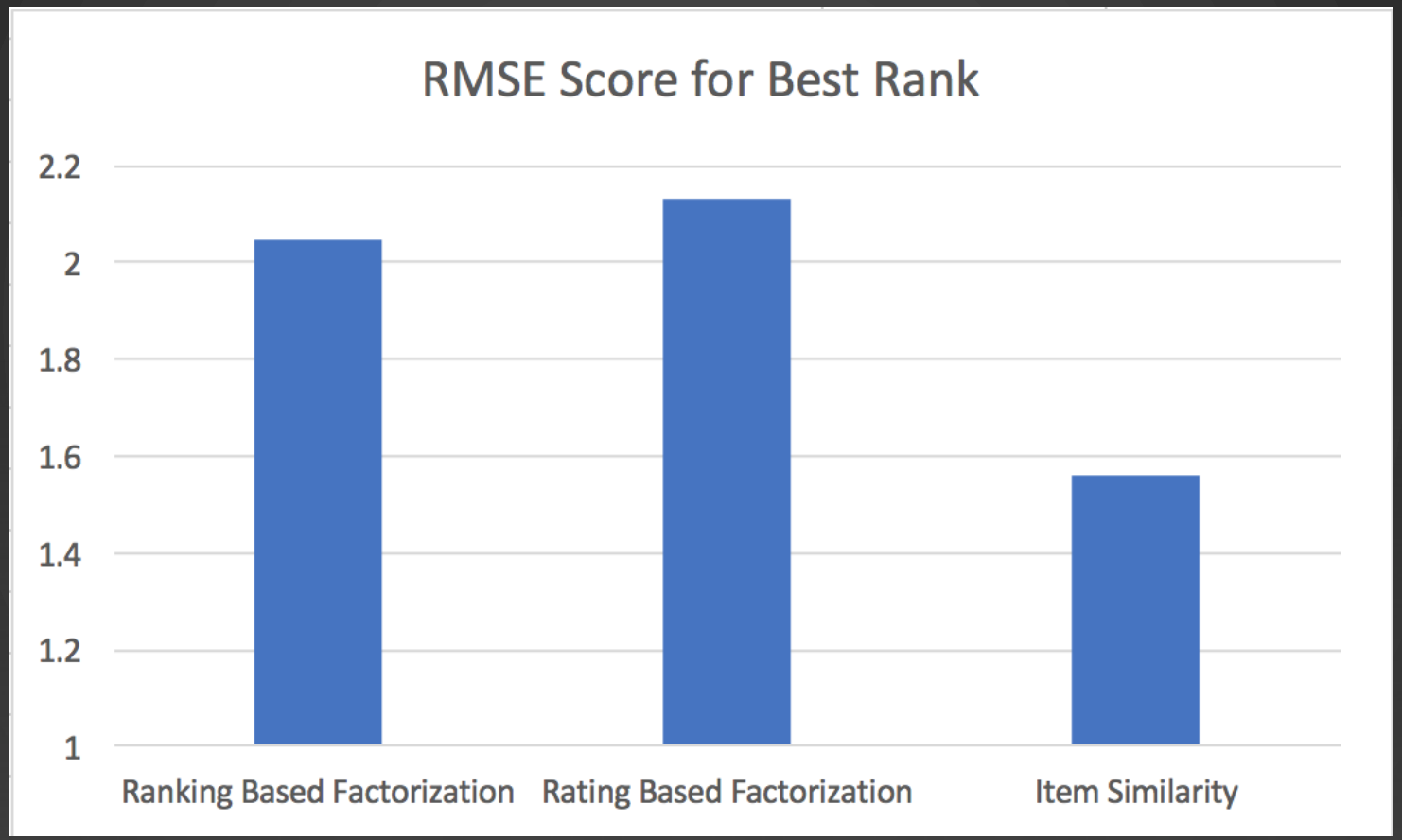
RECOMMENDATION SYSTEM

- Train all below three models to obtain the best model. All of these models take user-item rating matrix :-
  1) Ranking Factorization
  2)Rating Factorization
  3) Item Based Factorization
- Steps:
  - Split the data in train and validation using Graphlab library.
  - Define all these models.
  - Train these models for various latent factors: Factors required in factorization.
  - Optimizing for RMSE.
  - Plot various models to obtain which model and factor performed the best.

ALGORITHMS

- **Ranking Factorization** :- A RankingFactorizationRecommender learns latent factors for each user and item and uses them to rank recommended items according to the likelihood of observing those (user, item) pairs. RankingFactorizationRecommender contains a number of options that tailor to a variety of datasets and evaluation metrics, making this one of the most powerful models in the GraphLab Create recommender toolkit.
- **Rating Factorization** :- Create a FactorizationRecommender that learns latent factors for each user and item and uses them to make rating predictions. This includes both standard matrix factorization as well as factorization machines mode.
- **Item Based Factorization** :- This model first computes the similarity between items using the observations of users who have interacted with both items. Given a similarity between item i and j, S(i,j)S, it scores an item j for user u using a weighted average of the user's previous observations I(u).

EVALUATION OF THE RECOMMENDATION SYSTEM

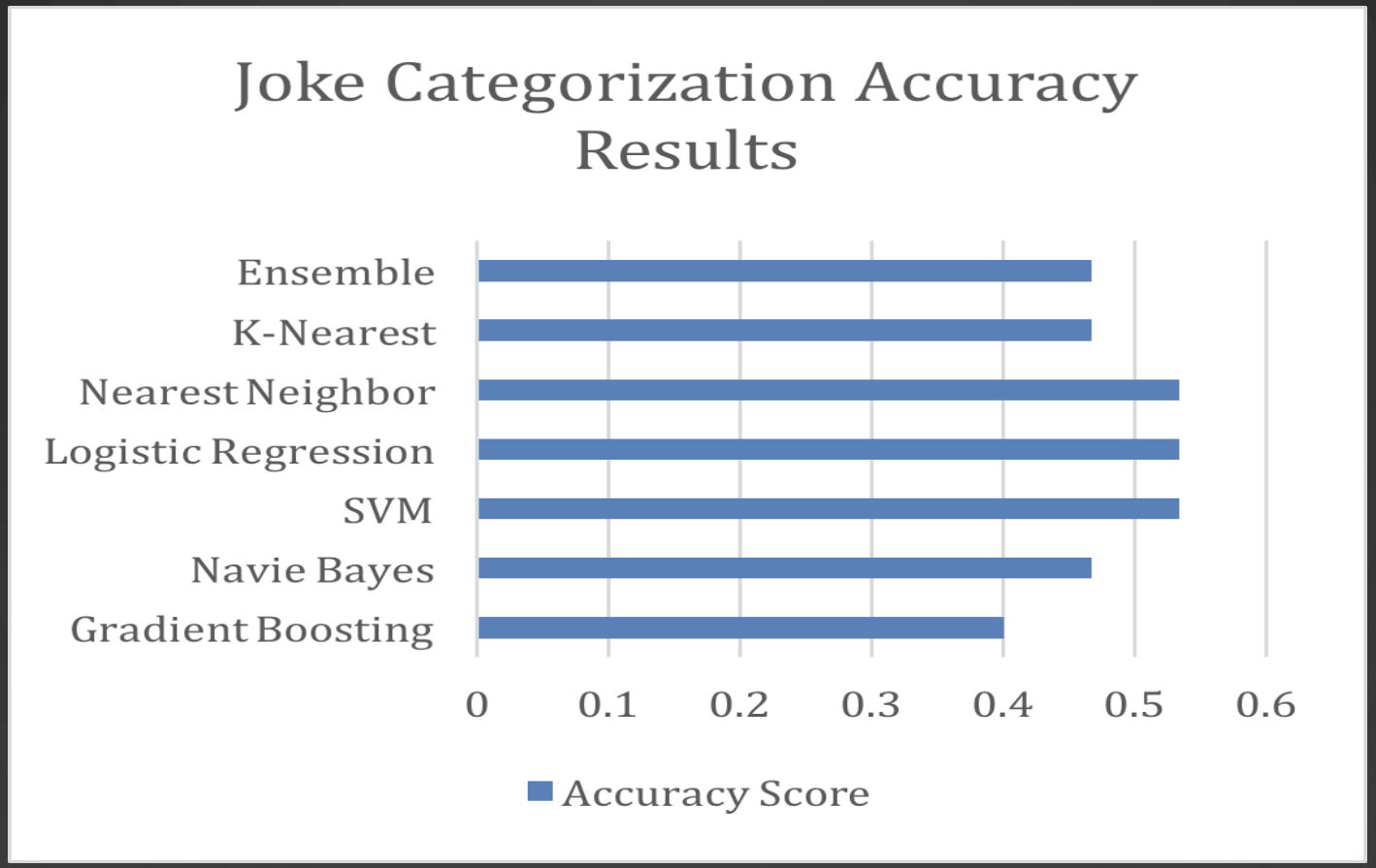


JOKE CATEGORIZATION

- This is the Code for predicting the category of joke from the following categories.We have pre-labelled the jokes data (150 jokes) into the mentioned classes. Following is the distribution of the categories:-

Category	Percentage
1. Animals	5.33
2. Technology	12.66
3. Doctor	4.67
4. Man	7.33
5. Politics	10
6. Relationship	8.68
7. Religion	0.67
8. School	0.67
9. Food	5.33
10. Others	38.66

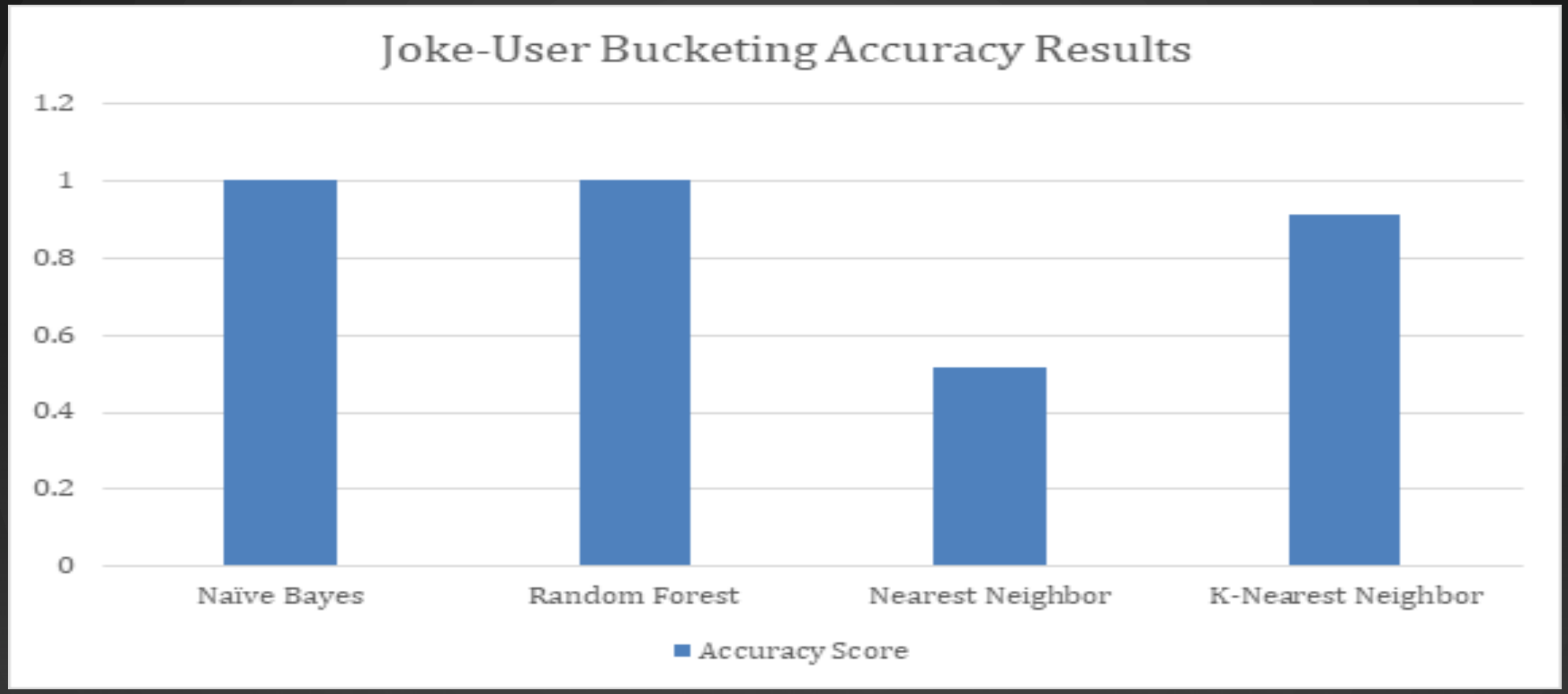
EVALUATION OF JOKE CATEGORIZATION



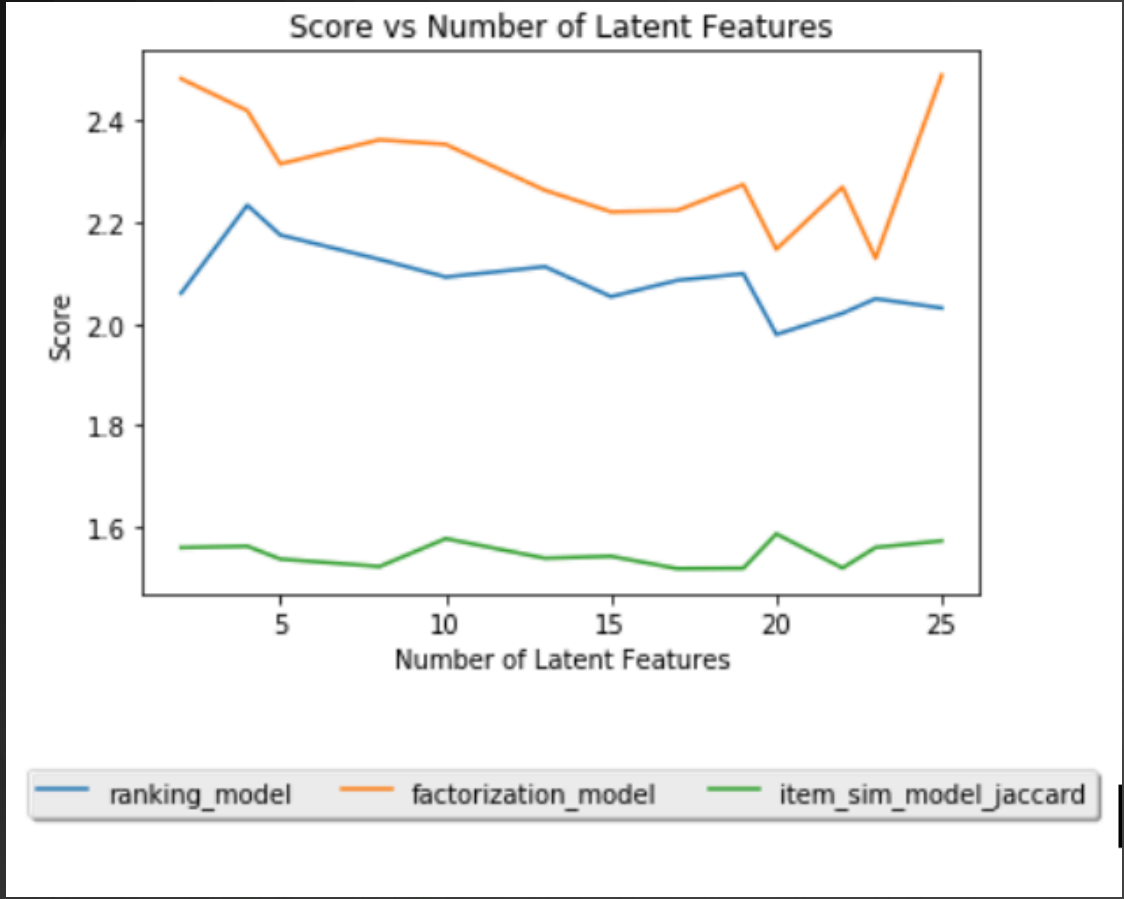
JOKE BUCKETING

- **Steps:**
  - Obtain the Joke categorized dataset
  - Retrieve jokes for the specified category
  - Classify/cluster user likes and dislikes for the specific jokes under specified category.
  - Show users liking the jokes belonging to specified category.
- **Training:**
  - Dataset is divided into 90% training and 10% testing data.
  - User likes and dislikes are decided from ratings.
  - Model is trained for user likes and dislikes.
  - Classification in 0:dislike, 1 :like is done using algorithms mentioned.
  - From the results and categorization output, joke-category to user bucketing is done.

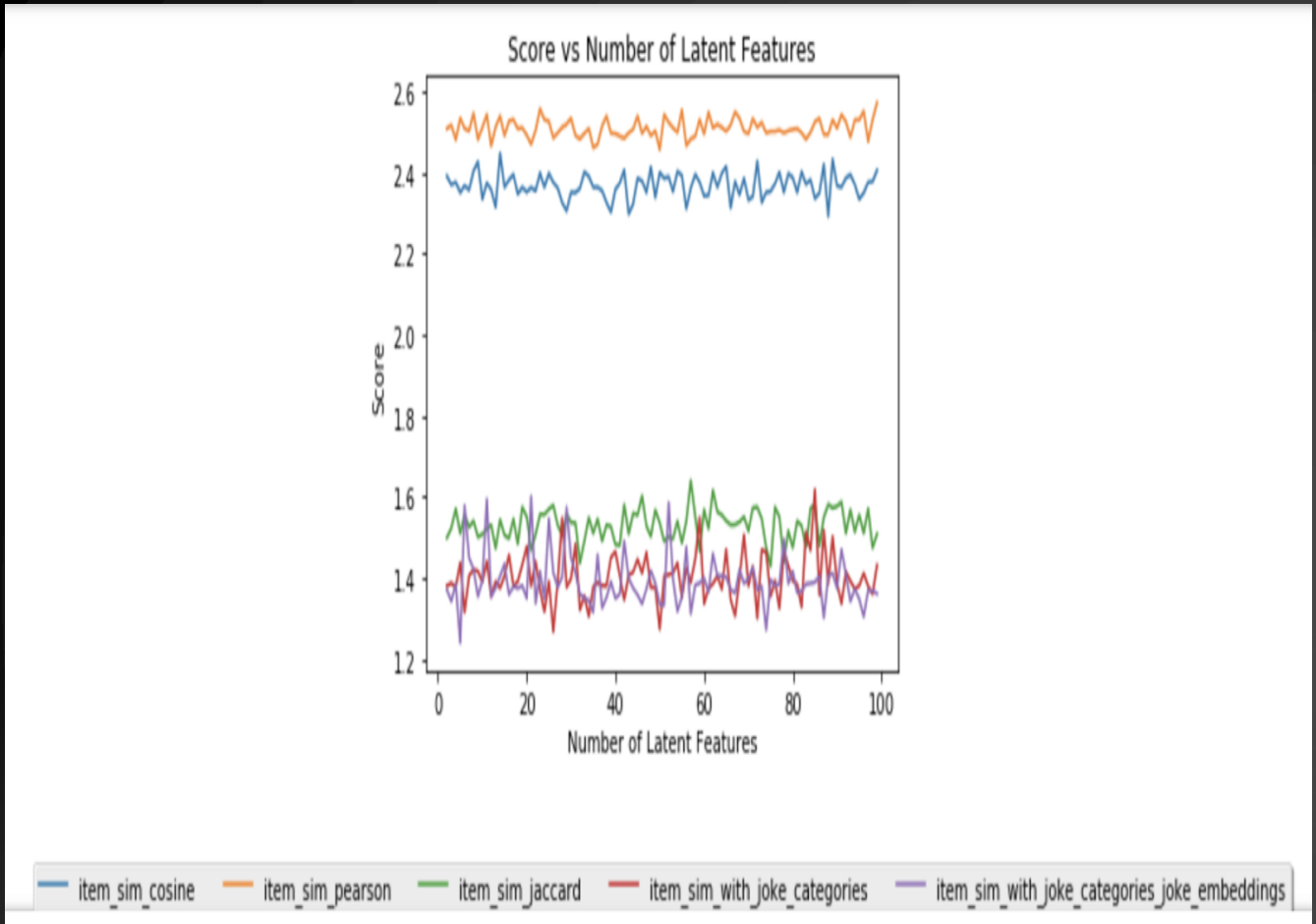
EVALUATION OF JOKE BUCKETING



ANALYSIS RESULT



From the figure mentioned, it is clear that Item Similarity gives least RMSE. Therefore, we further improved Item Similarity based recommendation with joke level features such as joke category and Joke Glove vectors.



We tried Item similarity based recommendation in five settings:

1.Item Similarity using Pearson
2. Item Similarity using Cosine
3. Item Similarity using Jaccard
4. Item Similarity using Jaccard and Joke Categories
5. Item Similarity using Jaccard and Joke Categories and Joke level Glove Features.

We observed that as we add more item level information, the model further improves and does much better compared to the baseline models. Hence, the model with Joke categories and Joke level Glove Features gave us the best result.

CONCLUSIONS

- We build a Joke Recommendation system using Jester dataset, which has over 1 million ratings by around 80,000 users on 150 jokes. We explored this dataset and performed a thorough analysis on various kinds of recommendation systems for this dataset.
- **Difficulties faced**
  - Joke level features, since there are only 150 jokes and hard to derives word level unigram.n-gram level features.
- **Things that worked**
  - GloVe word factorization worked.
- **Things that didn't work well**
  - SVM Algorithm for Joke-User bucketing halted the system.
- **Future Work**
  - Add item level information in Ranking and Rating based Recommendation System to see if it improves there as well in those cases as well just the way it improved item level.