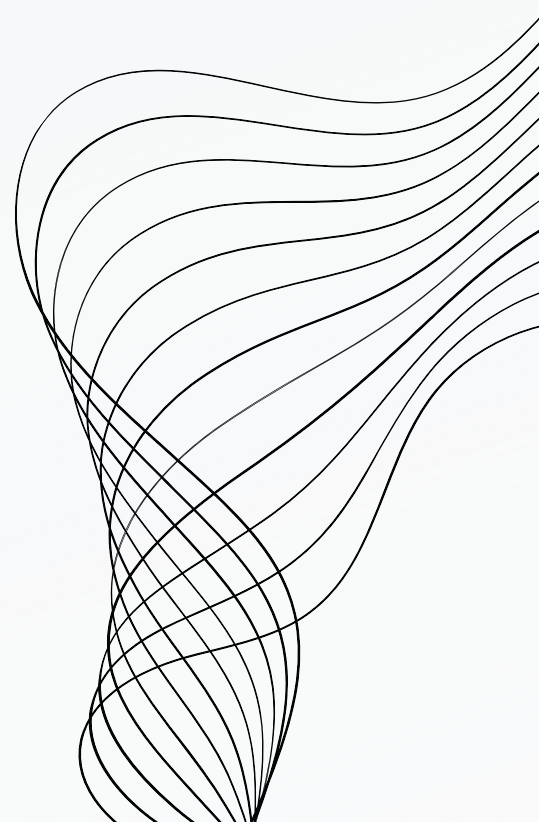




MACHINE LEARNING - GROUP 27

# **JOKE RECOMMENDER SYSTEM**

LAKSHYA - 2021262  
ZUBAIDA FATIMA - 2021221  
AHMED HANOON - 2021006  
SUMIT BHAGAT - MT22302  
NITISH KUMAR - MT23129



# PROBLEM STATEMENT

Individuals have varying senses of humour, and it can be challenging to propose a novel system that can recommend a funny joke to people. In this project, a model would be proposed that can rate jokes provided to it.

This system will implement classical machine-learning techniques to build a joke rating system. Lastly, the performance would be evaluated by predicting the accuracy of ratings of 20 new jokes

## ABOUT THE DATASET

In this project, we use the Jester Dataset (<https://eigentaste.berkeley.edu/dataset/>) to create the Joke Recommender Systems using a hybrid approach that uses both Collaborative Filtering and Content-Based Filtering.

The Jester dataset containing 6.5 million anonymous ratings of jokes. Focused on supporting algorithms like Eigentaste, the datasets offer three versions, featuring 4.1 million ratings from 73,421 users. For this project, we use Dataset 1

Spanning a diverse range of user preferences, the dataset includes ratings on a scale from -10.00 to +10.00. The datasets play a pivotal role in advancing the filtering algorithms and refining recommendation systems. The data matrix dimensions vary based on the number of jokes rated by users. It captures user ratings, with null values marked as "99" denoting unrated jokes. The datasets are further distinguished by the density of ratings for specific jokes, offering insights into universal user preferences.

# EDA

## Data Loading and Cleaning:

### 1. Jokes Data Cleaning

- Dataset contains 73,418 users who rated 100 jokes. Unrated jokes are given 99.00 value.
- Jokes contain stop words, HTML tags, they are removed, each word tokenized and jokes are lemmatized.
- Handling Missing Values: Impute missing values with median of columns

### 2. User Data Cleaning

- a. There is null values present in the data which can introduce bias, removing it is necessary for accuracy
- b. We cluster the jokes by using TF-IDF and applying dimensionality reduction via NMF

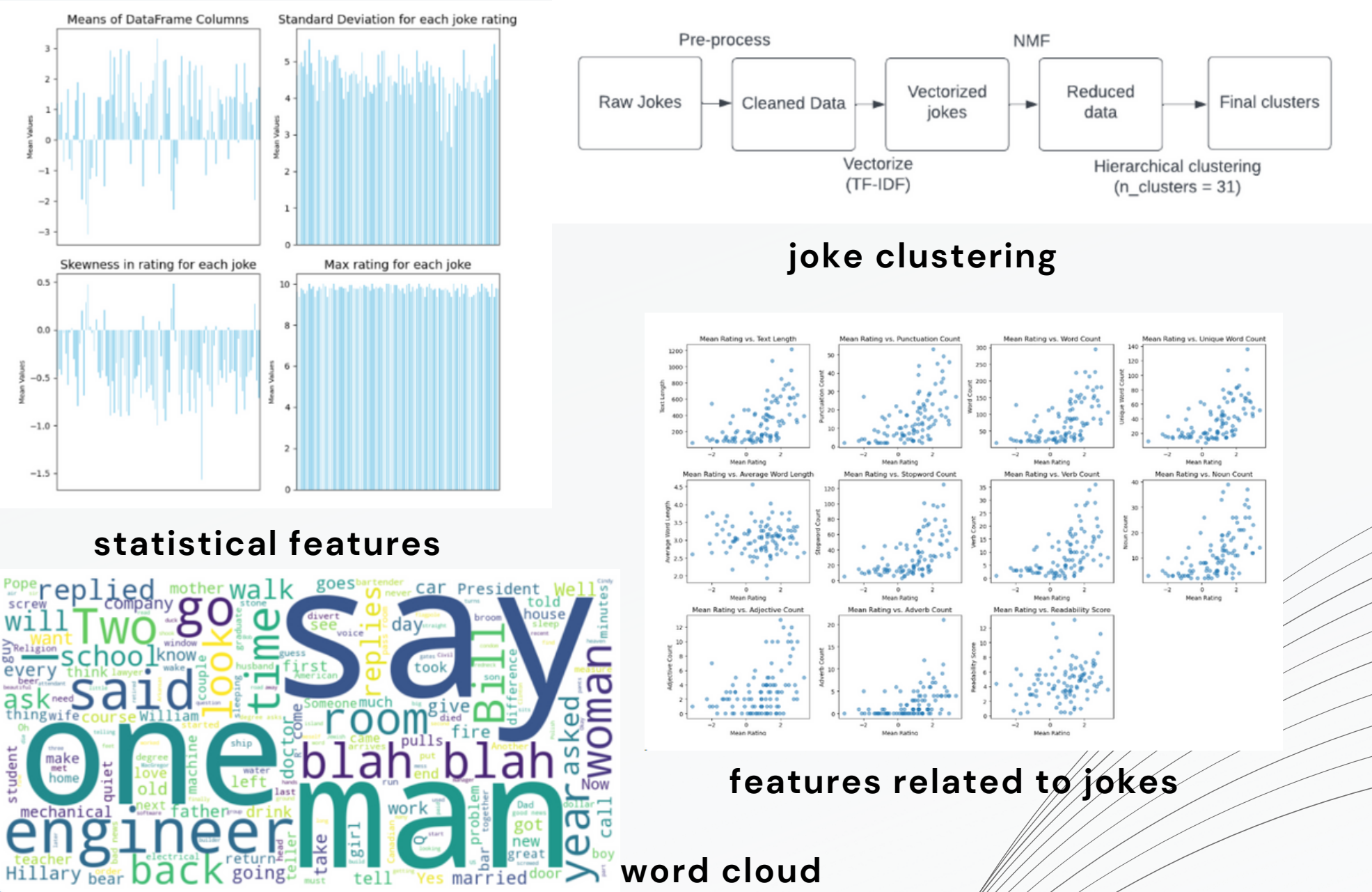
## Data Visualization:

We plotted the following figures to represent our data better

- Plot Missing Values:** Presentation of missing values in the data set.
- Correlation Heat map:** Heat map depicting the correlation of joke ratings.
- Punctuation vs. Mean Rating:** Investigating the association of punctuation number for jokes and average scores.
- Low-rated joke Word Cloud:** Construct a word map illustrating frequently-used words in low rated jokes
- Scatter Plot:** Conduct a scatterplot analysis of joke lengths against average rating and also observe the Silhouette Score
- Top and Bottom Jokes:** Show the best and worst ranked jokes according to their average rating. skewness provides important context for our recommendation system

## Pattern Finding:

- Joke Ratings:** Distribution of mean ratings across comedic patterns.
- Correlation Analysis:** Determined, correlations among jokes in order to discover possible configurations in a matrix format.
- Punctuation Impact:** Examines what effect punctuations bring on jokes' mean ratings.
- Plot Statistical Features:** Plot mean, SD, skew, and max rating for every joke.
- Analyze 75th Percentile:** Plot the 75th percentile for every joke and use this distribution to explain how people are distributed by their rating.





# METHODOLOGY

Two methodologies considered: **User-Based Rating** and **Average Ratings** of jokes.

In user-based rating, individual user predictions for a new joke are generated, while in the Average Ratings approach, the average rating of a new joke reflects its overall likeliness among all users

## 1. User-Rating Data Pre-Processing

### **Average Joke Ratings model:**

- Combined three Jester Dataset-1 rating datasets into one.
- Replaced missing values (99.0) with NaN.

### **User-Based Joke Ratings model:**

- Data formatting similar to Average Joke Ratings
- Applied Singular Value Decomposition (SVD) for matrix factorization and clipped predicted matrix between -10 and 10.

## 2. Feature Engineering

### **Average Joke Ratings model:**

- Extracted jokes, cleaned by removing hyphens.
- Tokenized, removed stop words, and lemmatized.
- Calculated TF-IDF values.
- Applied Part-Of-Speech Tagging.
- Extracted features: irony detection, sentiment analysis, structural features, humour features, POS distributions, lexical features, and stylistic features.

### **User-Based Joke Ratings model:**

- Phrases and Phraser models are created to identify bigrams and trigram
- Prepared Dictionary and Corpus for LDA and NMF topic modelling and trained it for topic extraction.
- Conducted sentiment analysis to extract features. This process is repeated for every joke.

# METHODOLOGY

## 3. Models

### Average Joke Ratings model

A few different models were tested to obtain the best possible squared value, they are :

- Ada Boost regressor
- SVR with different parameters ( $c=1$  and  $c=2$ )
- Ridge regressor and Bayesian ridge regressor

For each of them, R-squared value is obtained using K-Folds validation.

### User-Based Joke Ratings model:

The user-based method involves no specific model training. The test set jokes undergo transformation, and their representation is compared to the representation of jokes in the training set using cosine similarity.

## 4. Test

### Average Joke ratings model

- The model's fit was assessed using K-Fold cross-validation with  $K=3$  and an average R-squared score. In K-Fold with  $K=3$ , the data is divided into three equally sized folds.
- The model undergoes training three times, each iteration using two folds as the training set and one fold as the testing set.
- R-squared scores are computed for each fold, and the average performance is then calculated

### User-Based Joke ratings model

- The similar jokes are chosen based on content features. Cosine similarities serve as weights, and weighted average ratings are computed for each user in the training set.
- This approach assigns more weight to ratings from users whose preferences closely align with the current test joke. The resulting ratings offer a prediction for the target joke's rating, determined by the average or weighted average rating of the selected similar jokes.

# RESULTS

After careful analysis, the average joke ratings model emerged as the preferred choice over the user-based ratings model. This decision is substantiated by higher R-square scores and a stronger alignment with the problem statement. The detailed results of both models are documented below for reference.

## Average Joke Ratings model

Model	Fold 1	Fold 2	Fold 3	Average R2 score
Ada Boost	0.172	0.255	0.471	0.299
Ridge	0.285	0.254	0.280	0.273
Bayesian Ridge	0.341	0.282	0.430	0.351
SVR, c=1	0.352	0.248	0.437	0.346
SVR, c=2	0.359	0.285	0.418	0.354

The SVR with C=2 yielded the highest R-square value, achieving an average R-squared score of **0.354**. R-squared serves as a measure of the model's goodness of fit, with a higher value indicating a better fit to the data.

In this context, the R-squared score of **0.354** signifies that the model, on average, predicts 35 percent of the relationship between independent and dependent variables.

## User-Based Joke Ratings model

The User-Based model exhibited comparatively poorer performance in testing due to the sparsity of the user rating dataset, containing multiple NaN values that impact predictions. The R-squared value obtained with LDA was approximately **0.053**, and for NMF, it was **0.034**



# COMPARISON WITH EXISTING ANALYSIS

- Existing analysis focuses on recommending seen jokes from the rated set, not suitable for suggesting rating entirely new jokes.
- Hence, we opted for a Content Based Filtering approach that extracts features from joke text for new joke recommendations.
- User-based ratings with cosine similarities showed weak accuracy, lower than the existing analysis.
- Improved accuracy was observed while using the average rating method, closely aligning with existing analysis results.

## MODEL COMPLEXITY

- Complexity encompasses model architecture, parameters, and algorithms, extending to preprocessing, feature engineering, and metric selection.
- Achieving an optimal level involves an iterative process, blending domain expertise with algorithmic understanding.
- Analysis of ML model complexity during training involves steps like preprocessing, lemmatization, and feature extraction.
- Training complexity:  $O(M * (N + M))$ , where  $M$  is the number of jokes and  $N$  is the average length.
- Testing complexity:  $O(M * N)$  during preprocessing,  $O(M)$  for SVR prediction.

# CONCLUSION

This project focuses on constructing a recommendation system tailored to suggest ratings for newly introduced jokes.

Leveraging Natural Language Processing (NLP) and associated concepts such as sentiment analysis, topic modeling, Word to Vector using TF-IDF, and Bag of Words forms the foundation of our recommendation system.

By extracting various features from raw texts, we aim to enhance the model's ability to generalize effectively across a spectrum of jokes. Subsequently, diverse algorithms are systematically applied, and Support Vector Regression (SVR) with a parameter setting of  $C = 2$  emerges as the most suitable choice based on performance evaluations.

The report concludes with a thorough analysis of model complexity, refining our understanding of the system's intricacies and optimizing its performance. Our approach ensures the robustness and efficacy of our recommendation system for rating newly introduced jokes.