CMPE 256 Team 4
Recipe Rating Prediction

By

Ketan Malempati 016695354

Pruthvi Raj Kotigari 016681522

Rahul Raghava Peela 016668275

Dev Savsani 016710447

Introduction

- The abundance of food choices available today can make it difficult for consumers to decide what to eat. To address this problem, food recommendation systems have been developed to help users discover new food items that match their preferences.

- This project focuses on building a food recommendation system that predicts ratings based on user interactions with food items.

Dataset

- The dataset consists of the users, recipes, and their corresponding ratings, and reviews. It also consists of other metadata which is related to the recipes of the food such as recipes description, tags, steps involved, etc.
- Raw interaction dataset has approximately 1.1M rows and user_id, recipe_id, rating, review as the columns.
- Raw recipes dataset has approximately 200k rows and recipe_id, user_id, tags, steps, description, ingredients as the columns.

Modules

Since the dataset is having several textual features, detailed descriptions, and ratings, We would like to use Natural Language Processing to tackle this problem. So we are using text similarity as a base and building, evaluating different algorithms in terms of effectiveness and resources used.

- Preprocessing
- Sentiment-Analysis
- Glove
- TF-IDF
- BM25
- Neural Networks
- FAISS

Pre-processing/EDA

- We are using 2 tables of raw interactions and raw recipes. We have visualised all the columns in these tables and we can see that reviews and descriptions have Nan values. So we have dropped these rows.

- When we look at the bar charts and box plots for the columns we can see that majority of the data are close to each other but there are some outliers.

- When we look at the rating distribution we can see that the majority of the ratings are 5 so this may affect our predictions.

Sentimental Analysis

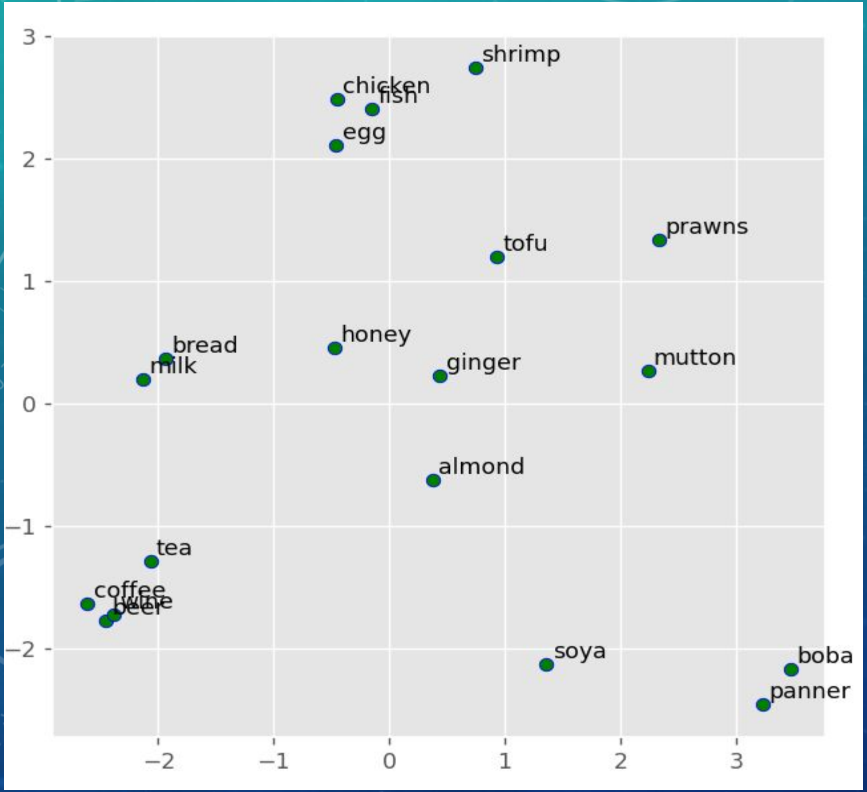
Sentimental Analysis is being used to find the sentiment of the reviews as it should be closely related to the ratings. We are using the Twitter-roBERTa-base model for finding the sentiment. It is trained on 58M tweets and fine-tuned. It finds the probability for each label and returns the name of the label with the highest probability. We then are using this sentiment data for training neural networks.

Glove

- The GloVe is a technique for producing vector representations of words using unsupervised learning. Training is done using corpus-based global word-word co-occurrence statistics, and the resulting representations show off some of the word vector space's fascinating linear substructures.
- The Glove has a lot of features but for this use case, we are going to use the nearest neighbour methodology of the glove where the Euclidean distance (or cosine similarity) between two-word vectors is an efficient method for determining the linguistic or semantic similarity of the words.

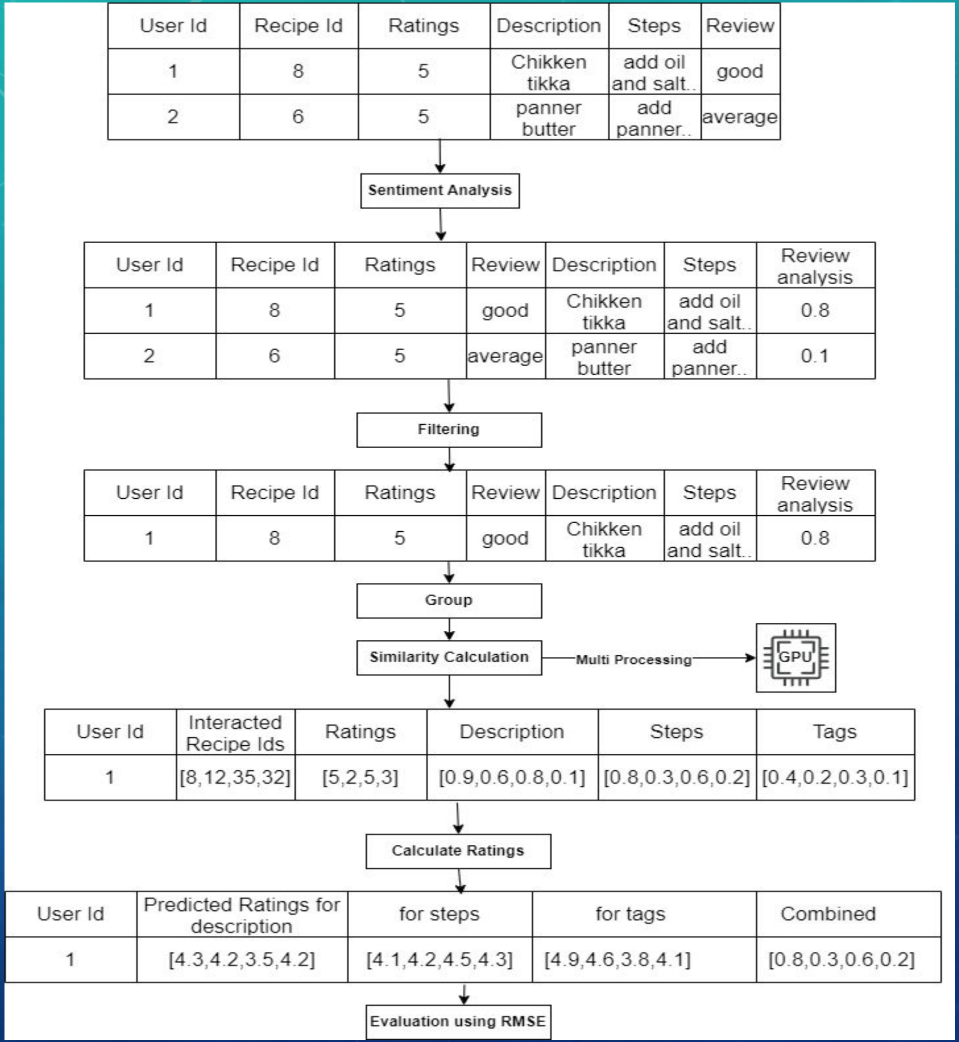
$$PR(x) = \frac{\sum_{i \epsilon RI} (GloveSimilarity(i,x) * rating(i))}{\sum_{i \epsilon RI} GloveSimilarity(i,x)}$$

A scatter plot of different ingredients of recipes using vectors from glove



Model Pipeline

Data flow using any similarity measure (Glove, TF-IDF, BM25, Spacy)



User Id	Recipe Id	Ratings	Description	Steps	Review
1	8	5	Chikken tikka	add oil and salt..	good
2	6	5	panner butter	add panner..	average

User Id	Recipe Id	Ratings	Review	Description	Steps	Review analysis
1	8	5	good	Chikken tikka	add oil and salt..	0.8
2	6	5	average	panner butter	add panner..	0.1

User Id	Recipe Id	Ratings	Review	Description	Steps	Review analysis
1	8	5	good	Chikken tikka	add oil and salt..	0.8

User Id	Interacted Recipe Ids	Ratings	Description	Steps	Tags
1	[8,12,35,32]	[5,2,5,3]	[0.9,0.6,0.8,0.1]	[0.8,0.3,0.6,0.2]	[0.4,0.2,0.3,0.1]

User Id	Predicted Ratings for description	for steps	for tags	Combined
1	[4.3,4.2,3.5,4.2]	[4.1,4.2,4.5,4.3]	[4.9,4.6,3.8,4.1]	[0.8,0.3,0.6,0.2]

TF-IDF

Term Frequency – Inverse Document Frequency (TF-IDF) is a popular statistical technique utilized in natural language processing and information retrieval to assess a term’s significance in a document compared to a group of documents, known as a corpus. The technique employs a text vectorization process to transform words in a text document into numerical values that denote their importance

$$w_{i,j} = tf_{i,j} \times log(\frac{N}{df_i})$$

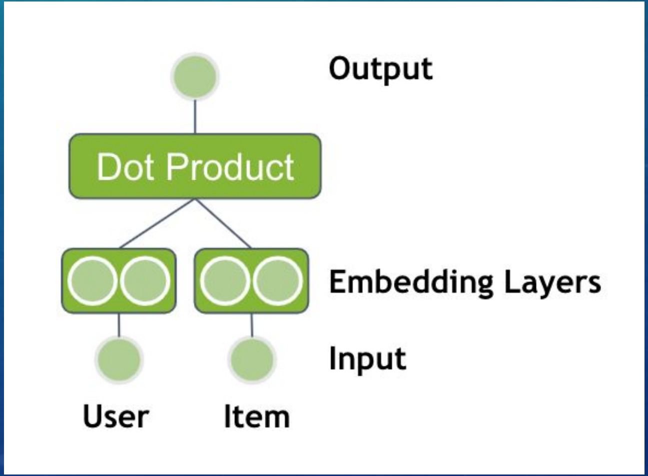
BM25

- BM25 is a simple Python package and can be used to index the data, tweets in our case, based on the search query. It works on the concept of TF-IDF. [3]. Same as TF-IDF it also cannot calculate semantic similarity but is faster than both TF-IDF and Glove.

$$\sum_{i}^{n} IDF(q_i) \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

Neural Networks

- Neural networks are a type of machine learning model inspired by the structure and function of the human brain
- Used embeddings as it captures the underlying semantic meaning of the input in a lower-dimensional space
- Created the used the user_id and recipe_id as an categorical embedding with the description of the recipes which also is an embedding to the data.
- Used output of the sentiment analysis as one more embedding and trained the neural network.

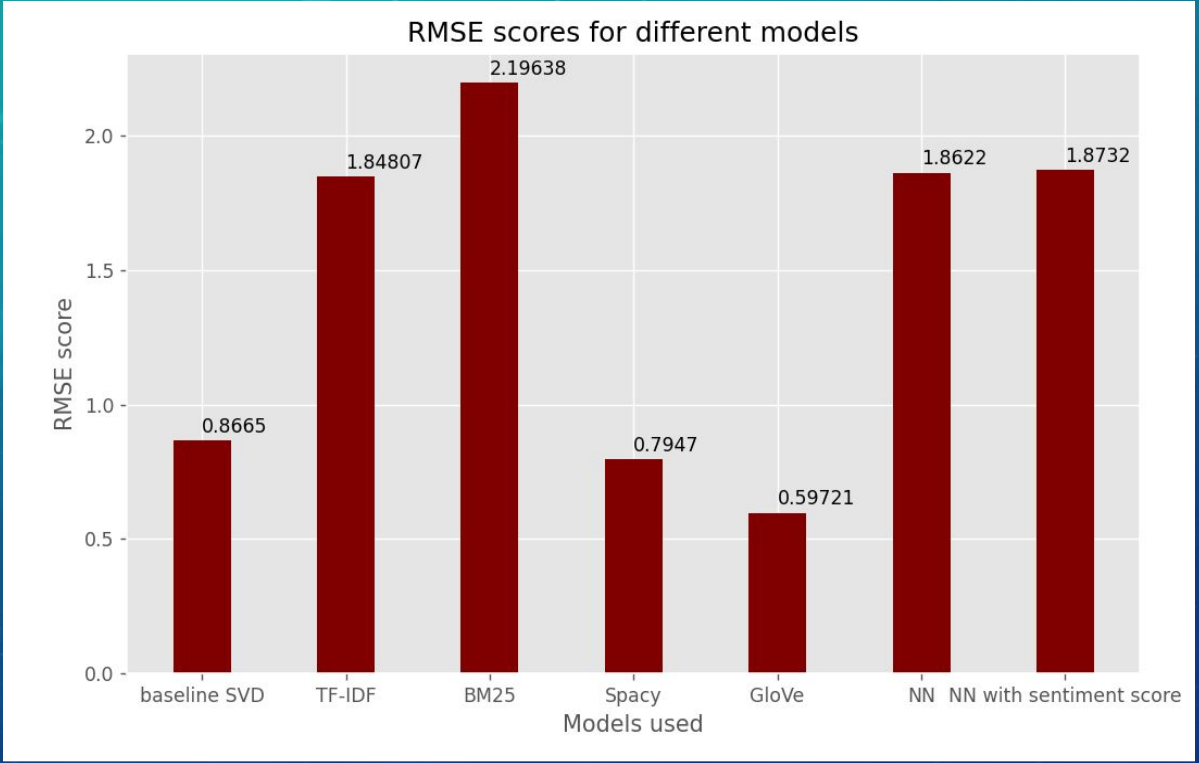


FAISS

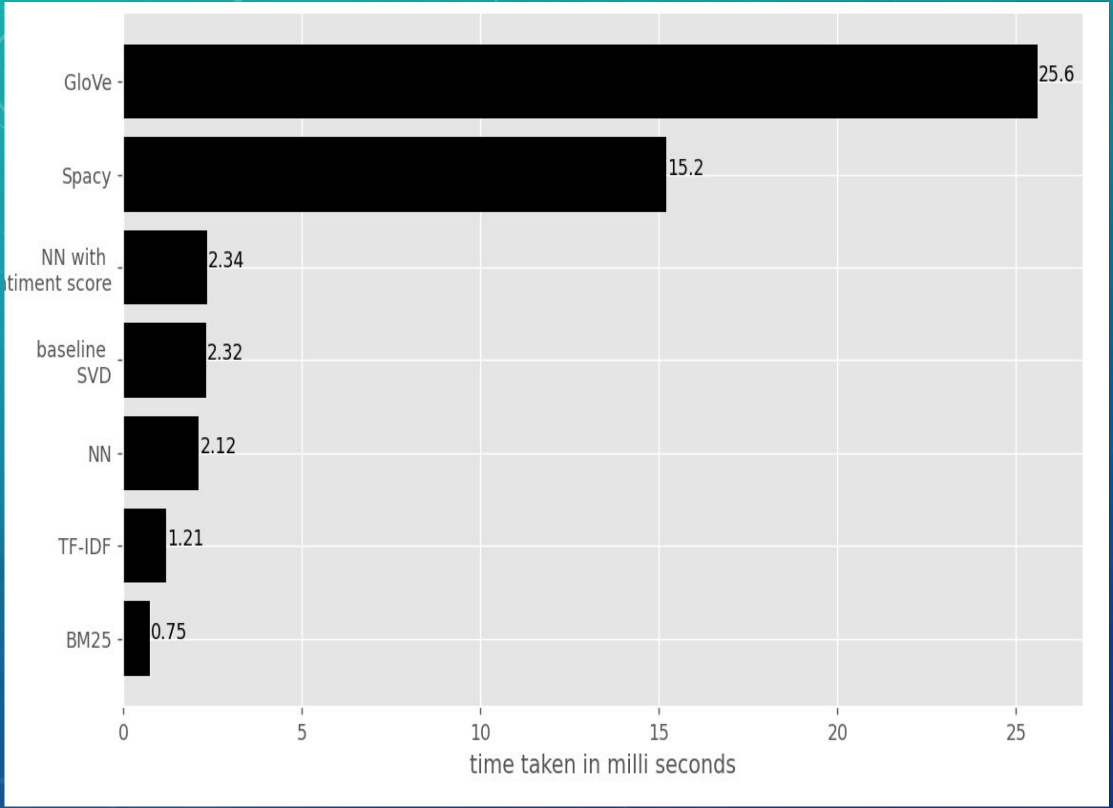
- Facebook AI similarity search (FAISS) is a algorithm developed by Facebook for fast and efficient similarity search specially for large data.
- 8.5x faster than the previous reported state-of-the-art.
- It employs Euclidean distance to identify the list of database objects that are closest to a given query vector, And it utilizes dot product to identify the list of database objects that exhibit the highest similarity with the query vector.
- For food rating recommendation dataset containing 1132198 reviews it took 32 minutes to train the index and 0.4 seconds to recommend the rating and 10 similar vectors.
- In this case we've used sentence transformer model 'paraphrase-distilroberta-base-v1' to convert text to embeddings.

Model Evaluation

We can see that Glove and Spacy as they use semantic scores and we get more information from them. TF-IDF and BM25 give the worst results because they are only based on simple word document similarity.



Time taken to predict ratings in milliseconds for each model



- Model Efficiency is calculated using time taken for prediction for a single user item pair

Conclusion/Future Work

- The main difficulty was to create embeddings and perform sentiment analysis since there are a lot of textual values with an average length of 20 words per row. Handling the imbalance in the data was another difficulty but we tackled it using sentiment analysis.

- we can see that Glove and Spacy as they use semantic scores and we get more information from them. TF-IDF and BM25 give the worst results because they are only based on simple word document similarity.

- In Future we would like to explore on feature Engineering and feature importance/weights so that we can fine tune the methodologies. Use Bagging methods to combine different models we trained.

Thank You