# RecipeWiz

Recipes Recommendation System

**Project Report**
**Presented to**
**CMPE 256**
**Fall 2022**


By
Team OdeToCode

Sanjana Kothari
Soumyendra Shrivastava
Chirudeep Gorle


December 18, 2022

# Table of Contents

# Project Description

Food has been around long before even humans walked this planet. It is the source of energy for all living beings that have evolved over centuries. With the birth of cultures, new and varying ways of preparing food emerged so much so that we now have 40+ cuisines around the world. And who wouldn't like to try them all?

With the advent of the internet, all information is now available at your fingertips, so much so that one can access recipes from across the globe. "RecipeWiz" is an engine that can help users find a subset of recipes available on Food.com. However, it is just not a recipe collection but offers a range of recommendations based on several search criteria pertaining to recipes as well as factors like user ratings and reviews. Our goal is to offer users a wide range of options based on different criteria like similar ingredients, similar nutritional value, similar methods of preparation, etc., by leveraging the powerful techniques of item-item collaborative filtering. When a user searches for a recipe, multiple carousels with recommended recipes will be displayed and each carousel shows recipes similar to the searched one. Clicking on a recipe will show all the details of that recipe like ingredients, preparation steps, nutritional information, etc.

# Project Requirements

**High-Level Features**

1. Landing Page
    - Search bar to search by recipe name, ingredients, etc.
    - Carousel of popular recipes (high rating and most number of reviews)
    - Carousel of newly added recipes
2. Search results
    - Recipe that has an exact match with the name, if any
    - Multiple carousels showing similar recipes based on different criteria like similar ingredients, similar nutritional value, similar ingredients and method of preparation, etc.
3. Recipe details
    - Pop-up that shows the details of the selected recipe like the description, date of upload of recipe, the ingredients, and steps to prepare the item, etc.
    - It will also include the ratings and reviews from users on Food.com

**Dataset**

The dataset used covers recipe information and user interaction information from Food.com, consisting of 180K+ recipes and 700K+ recipe reviews over 18 years of user interactions and uploads on Food.com. Multiple characteristics of each recipe include ingredients, method of preparation, cooking time, nutritional value, among others. As for the user-specific information, the dataset offers user ratings and reviews for the recipes along with the recipes that a user has interacted with. All this is available as CSV files on Kaggle.

**Project Deliverables**

- A website that shows the user top recipes recently added recipes and popular recipes available from Food.com.
- A search option to search for recipes by entering ingredients/name of recipe/other keywords, which would then show the user the most similar recipes based on the input criteria.
- Recipes along with a complete description, steps, ingredients, and user reviews and ratings as obtained on Food.com.

**Technology Stack**

| | |
|---|---|
| Frontend | React |
| Backend | Express.js, Node.js, and MongoDB |
| Recommendations | Python |

# KDD Process

KDD, Knowledge Discovery in Databases, involves several steps that help in identifying hidden patterns and meaningful information in large, complex datasets. It is widely used for data mining tasks. The process involves the below 7 steps:

1. **Data Cleaning**
    - Data Types present - numerical, string/text, and date
      Different cleaning techniques are applied to the different columns. For textual data - Firstly, the columns containing strings in the form of lists are converted to plain comma-separated strings. The nutrition column is formatted and expanded to add more columns that represent the various nutrition values. For date columns, the data type is changed from object to date for the easier and correct processing of date columns.
    - Rows containing NaN are dropped (NaN only in review and description) for models that use these features to recommend.
    - Columns like minutes, n_steps, and calories are seen to have outliers. The outliers are removed using the interquartile range. Points lying on either side of (Q1-1.5*IQR) and (Q3+1.5*IQR) are removed.
    - Recipes that have just one review which is from the author itself are removed as that might cause bias.

2. **Data Integration**
   As a part of the recipe recommendation, we have two datasets where one gives the details about the recipes themselves and the other gives the ratings and reviews for the recipes. For our recommendation engine, we combine the two datasets on the recipe_id so that we have each row giving details about the recipe and the ratings and reviews given to that recipe. Due to this, we get 'n' rows for each recipe where 'n' is the number of reviews for that recipe.

   RAW_recipes.csv contains the following features and 231637 records.
    - name -> recipe name
    - id -> recipe ID
    - minutes -> minutes to prepare
    - contributor_id -> ID of recipe contributor
    - submitted -> date recipe was submitted

- tags -> tags for recipes
- nutrition -> nutritional values of the recipe
- n_steps -> number of steps in the recipe
- steps -> steps of preparation
- description -> description of the recipe as given by the creator
- n_ingredients -> number of ingredients in the recipe
- ingredients -> list of ingredients used in the recipe

RAW_interactions.csv contains the following features and 1132367 records.

- user_id -> ID of reviewer
- recipe_id -> ID of recipe which is reviewed
- date -> date recipe was reviewed
- rating -> Rating given to recipe by user
- review -> Review written by user for the recipe

https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions

3. **Data Selection**

   Data selection involves selecting features that are relevant to the task of data mining. After joining the above two datasets on the recipe_id, we keep only those columns that will be used later to make recommendations based on different criteria. Other columns like user_id, review, contributor_id, submitted, and some intermediate columns created for exploratory data analysis purposes are dropped.

4. **Data Transformation**

   The recommendations are made using different statistical methods and content-based item-item recommendation filters. The data transformation for each technique differs as it is based on the model being used. Broadly, the following transformations are used.

   - Imputing missing ratings with the median.
   - New features like the cuisine and type of meal - veg/ non-veg and sweet/savory are derived from the ingredients and tags column respectively.
   - Using the nutrition values, recipes are classified as healthy or unhealthy.

- In textual columns like ingredients, steps, etc. - Natural Language Processing techniques of lemmatization, tokenization, and stop word removal are applied. Along with that unnecessary numbers, punctuations, and newline characters are also removed.

5. **Data Mining**

RecipeWiz shows the different kinds of recommendations based on the searched criteria - the search can be by recipe name, ingredients, cuisine, etc. These recommendations are based on different parameters and are developed using different methods, both memory-based and model-based. Since our data contains textual data, it is first converted into numeric data using techniques like TFIDF vectorizer, Word2Vec, and Sentence Transformer as machine learning models operate only on numeric data. Once embeddings are obtained, we provide similar recipe recommendations using below mentioned techniques.

- Memory-based techniques use statistical methods like cosine similarity and correlation to suggest similar recipes. These methods are very primitive and do not take into account the semantic similarity between data. It is purely mathematical and hence forms the base models.
- Model-based methods utilize more sophisticated techniques like clustering and deep learning. These methods, among others, have the ability to understand the vector space and the underlying patterns more closely, therefore providing highly similar recipes to the searched one. These recommendations are more likely to be viewed and tried due to their closeness to the searched criteria.

6. **Pattern Evaluation**

The recipe recommendations made by RecipeWiz can be evaluated under two broad categories - recommendation-centric and business-oriented metrics.

Recommendation-centric metrics:

- Diversity in recommendations offered - the average dissimilarity between all pairs of items in the result set.
- Coverage - the ability of the recommender system to recommend all items from a train set to users. The measure lies in the ability of the system to bring unexpectedness to the results.

Business-oriented metrics:

- Click-through rates - measurement of how many users click on the recommendations.
- User behavior and engagement - By showing more relevant recommendations, user engagement is estimated to increase. This achieves the goal of driving up business performance and profit.

## 7. Knowledge Presentation

As the final deliverable, RecipeWiz is a website that allows search functionality and then recommends similar recipes. These recipes are displayed as carousels on the website that the user can scroll through and open. On opening, these recipes will open up as separate articles with all details about the recipe.

# Feature Engineering

**Missing values**

Rating column contains 0 which represents that the recipe was not rated by the user who reviewed it. The missing ratings are imputed by the median of the same recipe rating.

**Feature splitting**

Nutrition column consists of a list of values that are expanded into 7 numerical features - calories, fat, sodium, saturat_fat, carbohydrates, protein, and sugar.

**Transforming textual data**

Tags, ingredients and steps are given as lists which are converted to strings, and punctuations are removed. This is done for embedding purposes.
Embeddings are done using different techniques like TF-IDF, Word2Vec, Sentence Transformation, etc. to suit different requirements and models.

**Feature Creation and encodings**

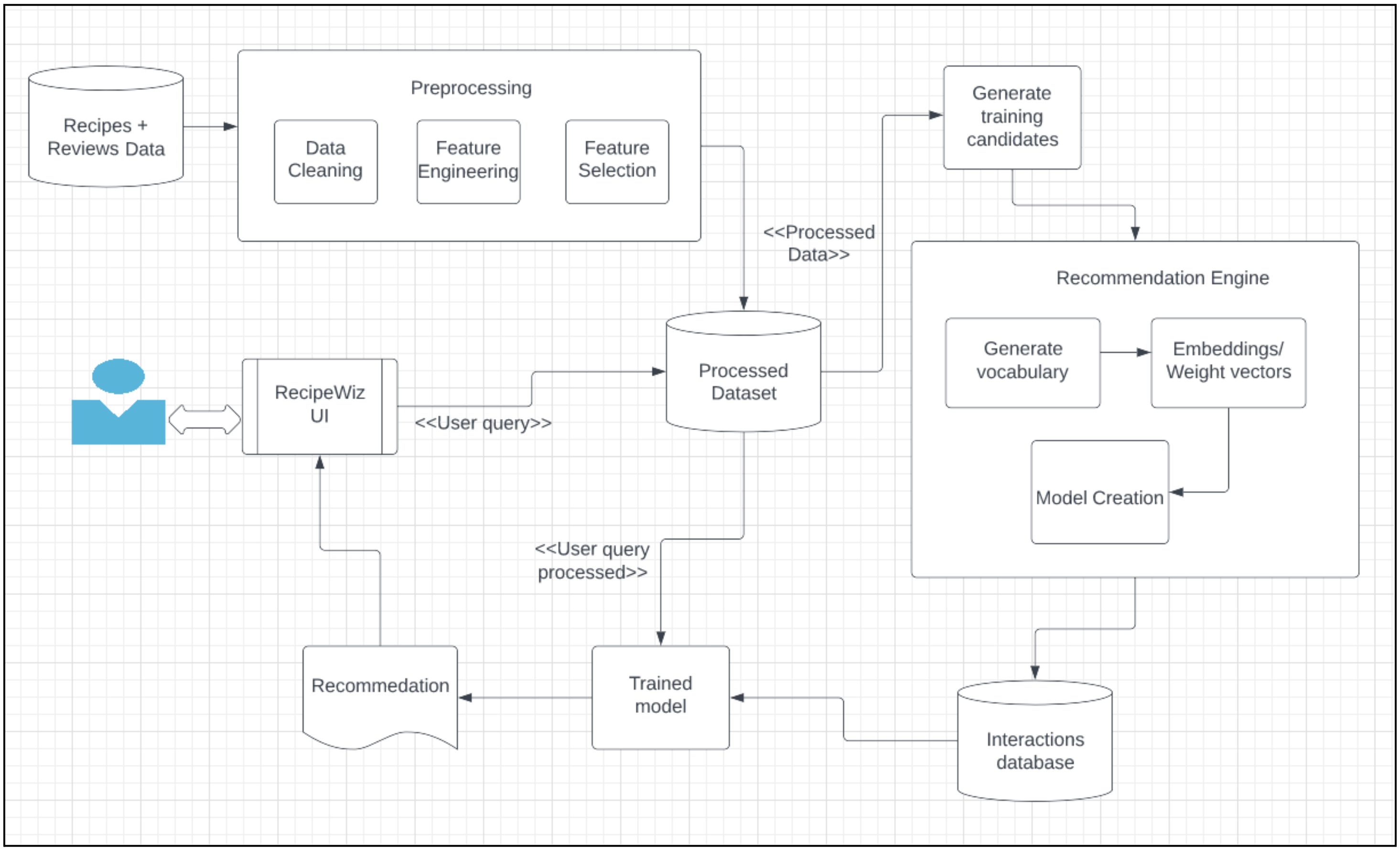
4 new features are derived. These are:

- Veg/ Non-veg derived from the ingredients column
- Sweet/ Savory and Cuisine derived from the tags
- Healthy/ Unhealthy derived from the nutrition values.

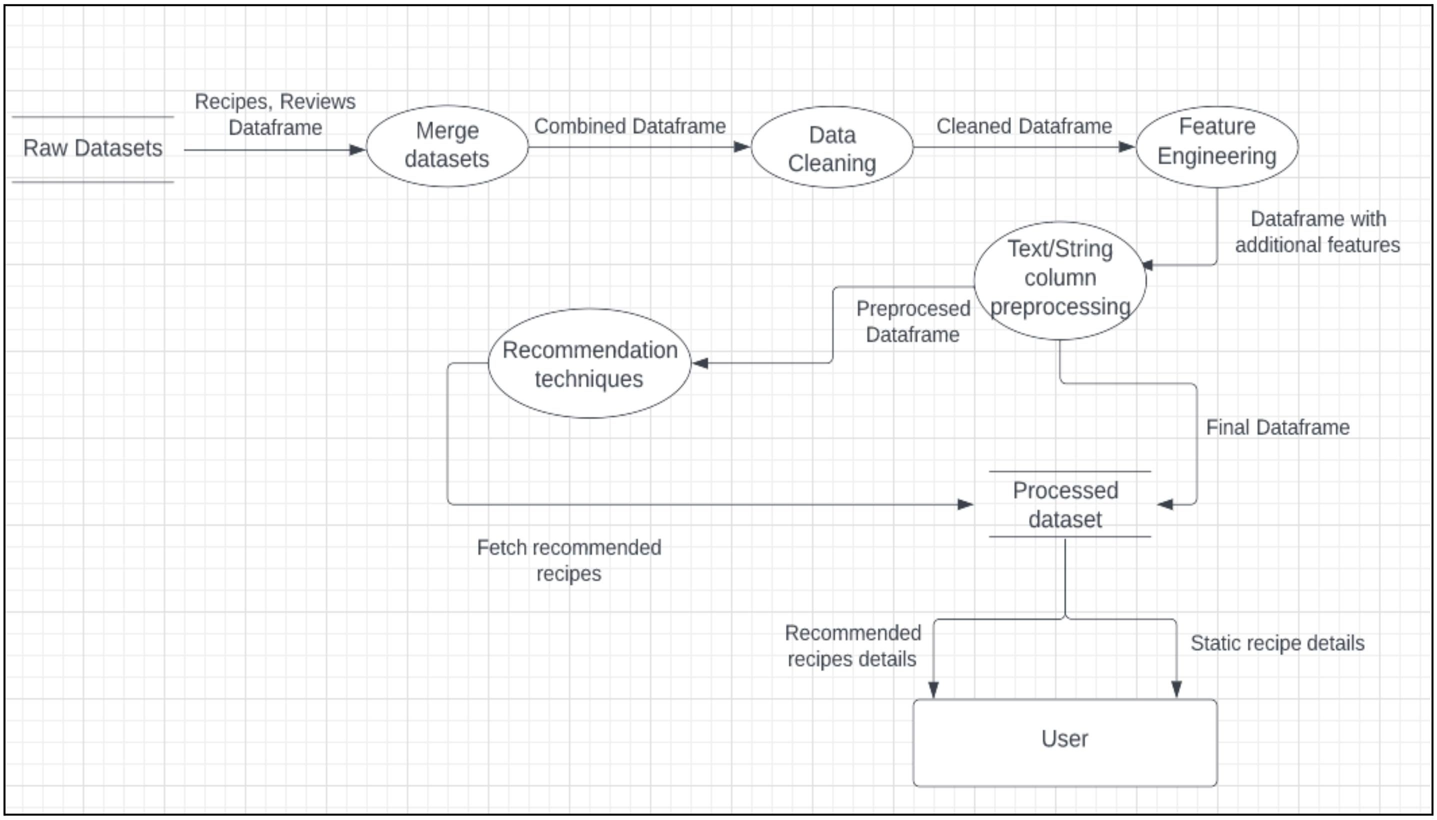
All these columns are finally one-hot encoded for input to the models.

Additional textual columns are added which are derived from columns like ingredients, steps, cuisine, etc. so that these can be embedded, allowing users to search by describing the recipe, etc. These columns are processed using NLP techniques like lemmatization, tokenization, and stop-word removal. Finally, they are embedded using several techniques like Word2Vec, Sentence Transformer, etc.

# High-level Architecture Design

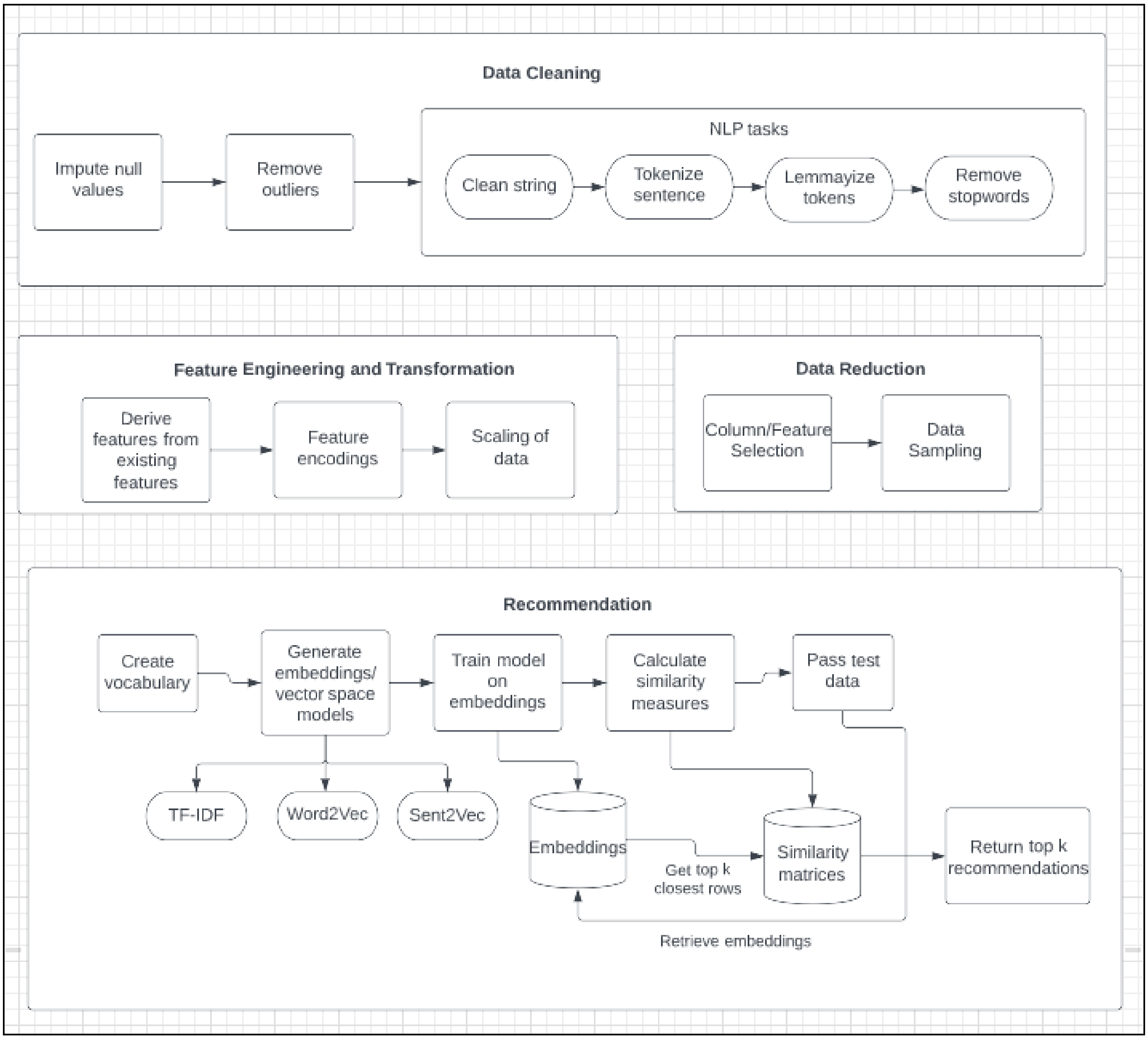


# Dataflow Diagram

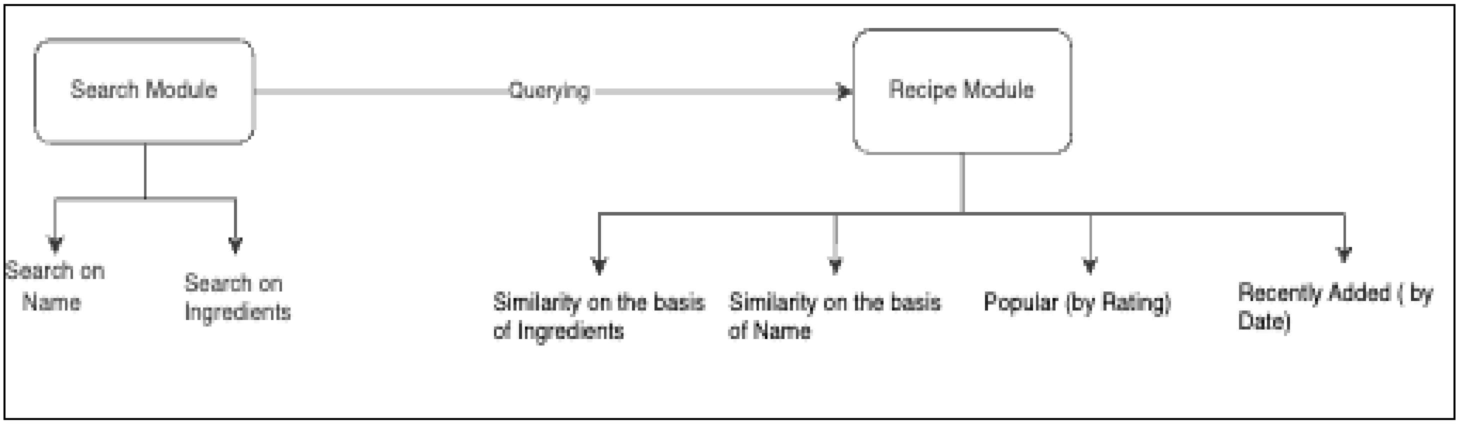


# Component-level Design

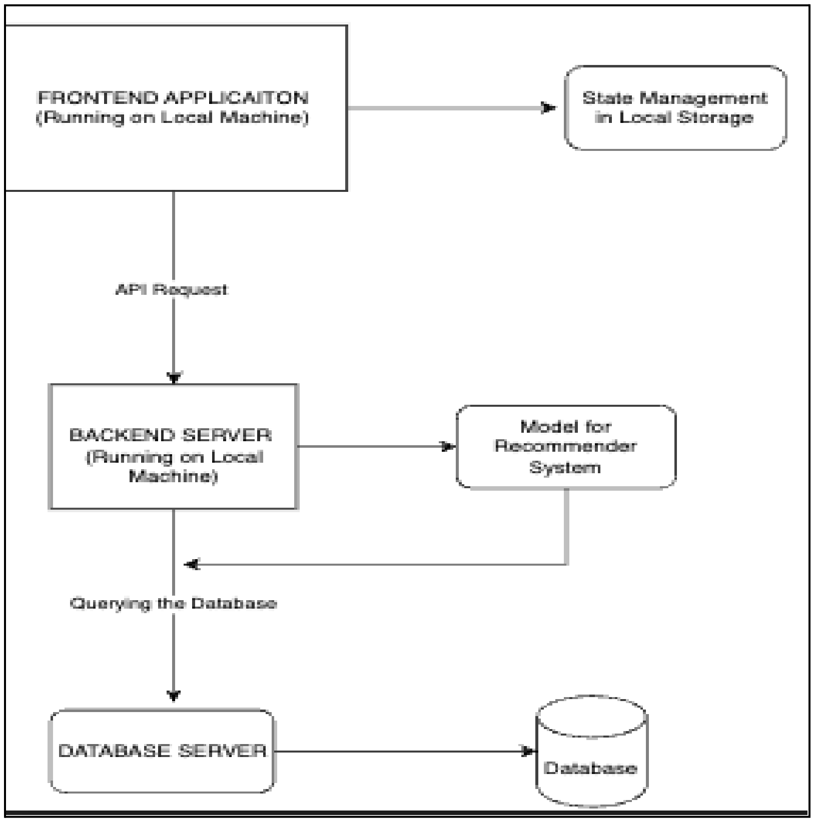
## Component Level Design of Recommendation Engine



**Component-level design of web application**



# Workflow


FRONTEND APPLICAITON
(Running on Local Machine)
State Management
in Local Storage
API Request
BACKEND SERVER
(Running on Local
Machine)
Model for
Recommender
System
Querying the Database
DATABASE SERVER
Database

# Data Science Algorithms

The following algorithms are at play with respect to the different features on which recipes are recommended. The recommendations are based on the recipe searched for by name, ingredients, or recipe features.

1. **Features Input - Recipe ingredients**
   **Algorithm - Term Frequency Inverse Document Frequency**
   TF is the number of times a term appears in a particular document. IDF is a measure of how common or rare a term is across the entire corpus of documents. If the word is common and appears in many documents, the IDF value (normalized) will approach 0 or else approach 1 if it's rare. The TfidfVectorizer is trained on ingredients in all records and when a recipe is queried by name, the ingredient embeddings of the searched recipe are extracted from the database, and recipes with the highest cosine similarity are returned.

   tf(t) = (No. of times term 't' occurs in a document) / (No. Of terms in a document)

   $idf(t) = \log_e [ n / df(t) ]$

2. **Features Input - Recipe steps of preparation**
   **Algorithm - Sent2Vec**
   Sent2Vec is an unsupervised model for learning sentence embeddings. It can be seen as an extension of the C-BOW model that allows to train and infer numerical representations of whole sentences instead of single words. The recipe steps of preparation are embedded using Sent2Vec. The searched recipe's steps are matched with the recipes that follow a similar method of preparation, using cosine similarity measure.

3. **Features Input - Recipe ingredients and steps**
   **Algorithm - Word2Vec**
   Word2vec is a combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model. These are shallow neural networks that map words to the target words. The learned weights act as word vector representations. The embeddings of ingredients and steps of the searched recipe are compared to the embeddings of other recipes using cosine similarity.

4. **Features Input - Recipe ingredients, type of recipe, type of meal, cuisine, healthiness, recipe tags**
   **Algorithm - FAISS Approximate Nearest Neighbor**
   Approximate Nearest Neighbour algorithm is implemented using the FAISS library. IVFPQ index, i.e. Inverted File Product Quantization, is used to create an indexing of the text column containing the above-mentioned metadata about the recipes, and a search in that index of the query is performed. Finally, the recipes with the highest cosine similarity score are returned as recommendations.

5. **Features Input - Recipe nutrition values**
   **Algorithm - Pearson Correlation Coefficient**
   The correlation of nutrition values of the searched recipe with respect to all other recipes is found and the recipes with a correlation > 99.9 are returned.

# Server-side Design

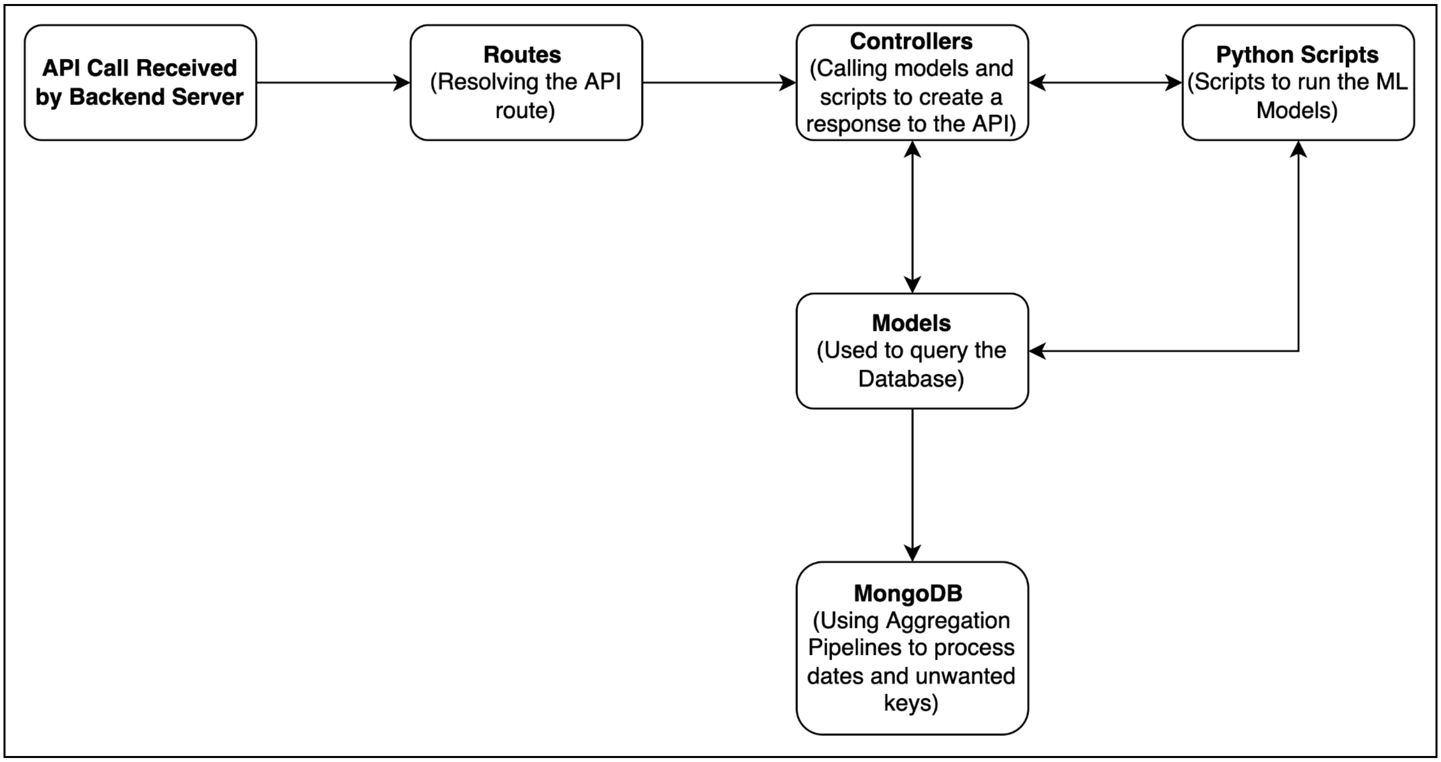
There are currently 4 APIs in the backend.

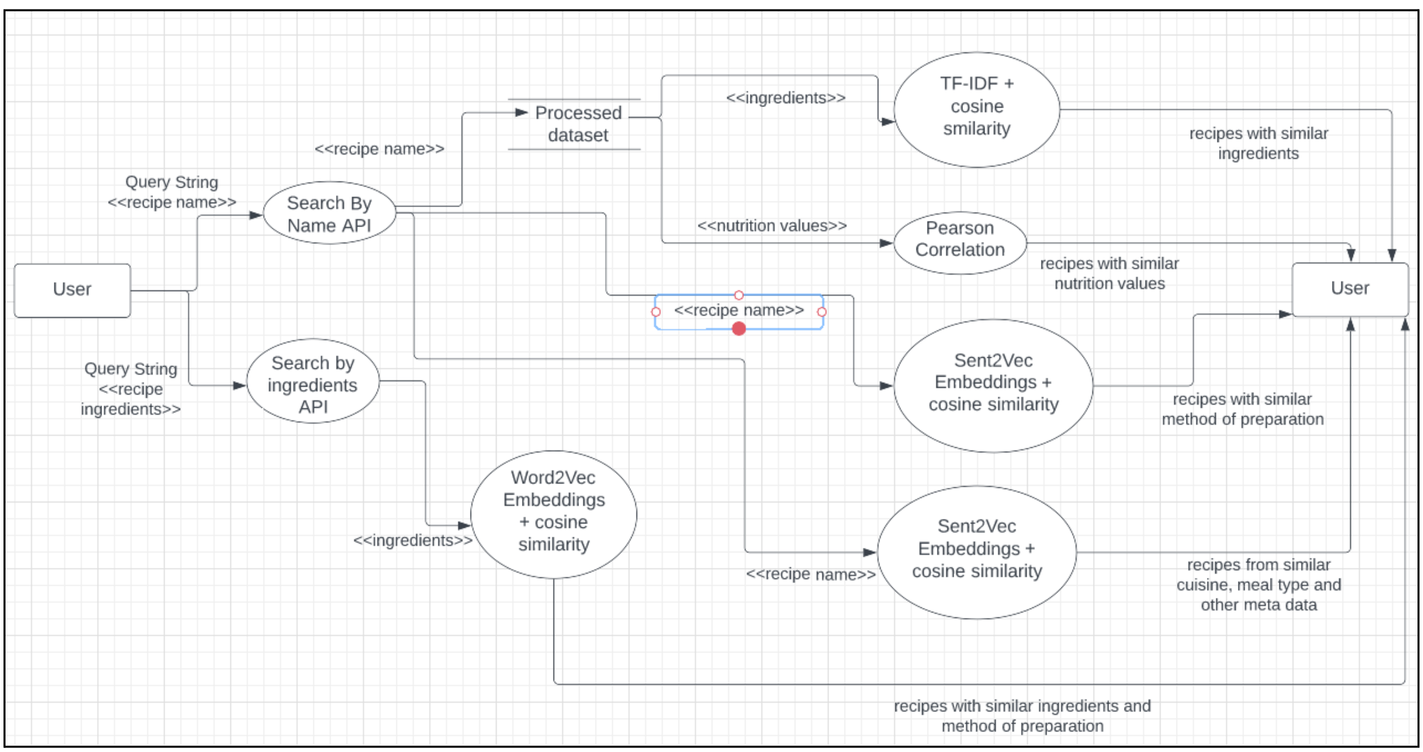
1) Get All recipes -
   Returns the 20 most popular and 20 most recently added recipes as per the ratings and recipe upload date on Food.com

2) Get Recommended recipes -
   An API that populates the different carousels with recommended recipes based on different search criteria and metadata about the searched recipe.
   a) Searching by name - Returns recipes with similar ingredients, similar nutritional value, and a similar method of preparation.
   b) Searching by ingredients - Returns recipes with similar ingredients and method of preparation, and recipes that have similar ingredients combined with other metadata like cuisine, type of meal, healthiness factor, etc.
   c) Searching by description/ any string query - Return similar recipes derived from metadata columns like ingredients, method of preparation, cuisine, meal type, etc.

3) Get Details -
   Returns all the details of the selected recipe like description, preparation time, ingredients required, preparation steps, ratings and reviews, etc.

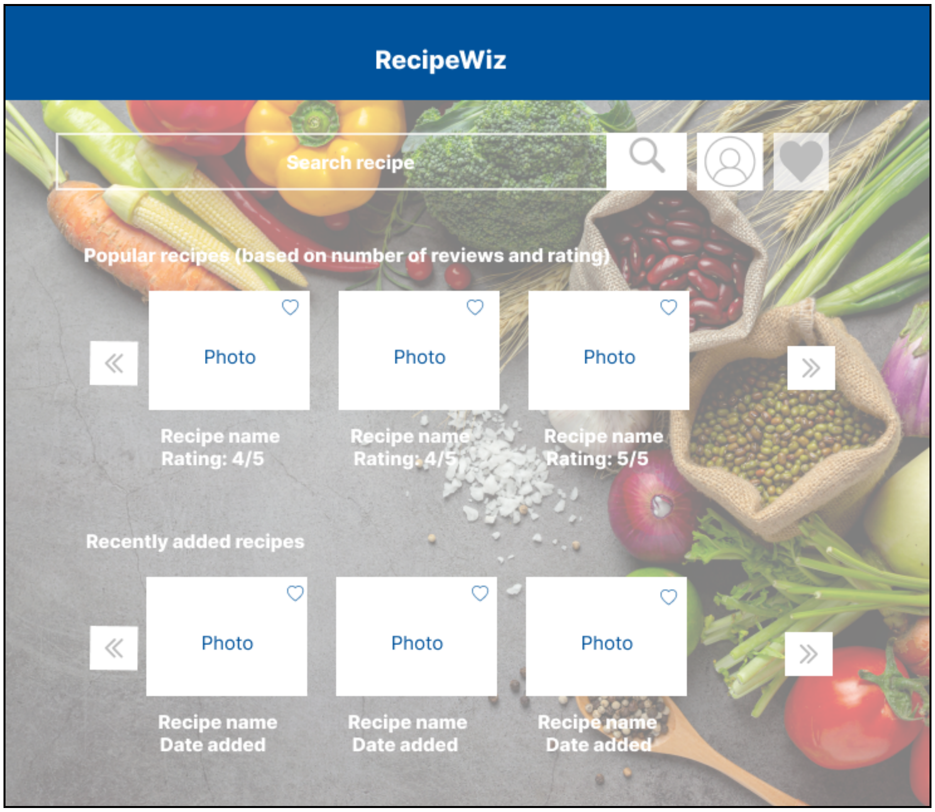
4) Get Item name for Search -
   Returns the 10 most relevant recipe names in our database.


API Call Received by Backend Server
Routes
(Resolving the API route)
Controllers
(Calling models and scripts to create a response to the API)
Python Scripts
(Scripts to run the ML Models)
Models
(Used to query the Database)
MongoDB
(Using Aggregation Pipelines to process dates and unwanted keys)

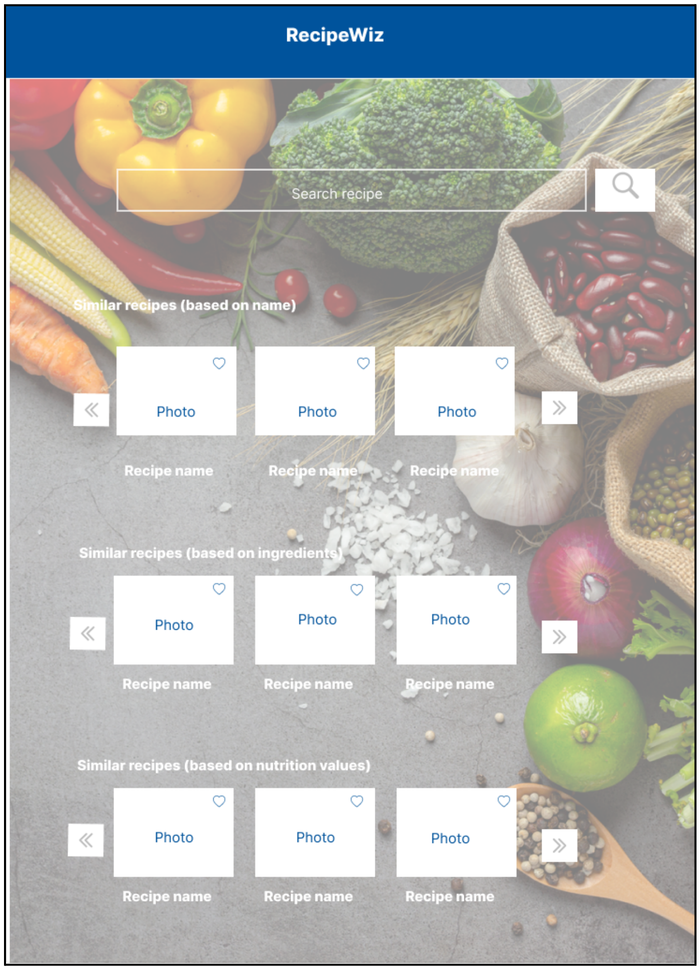

User
Query String
<<recipe name>>
Search By Name API
<<recipe name>>
Processed dataset
<<ingredients>>
TF-IDF + cosine similarity
recipes with similar ingredients
<<nutrition values>>
Pearson Correlation
recipes with similar nutrition values
User
<<recipe name>>
Sent2Vec Embeddings + cosine similarity
recipes with similar method of preparation
Query String
<<recipe ingredients>>
Search by ingredients API
<<ingredients>>
Word2Vec Embeddings + cosine similarity
<<recipe name>>
Sent2Vec Embeddings + cosine similarity
recipes from similar cuisine, meal type and other meta data
recipes with similar ingredients and method of preparation

# Client-side Design

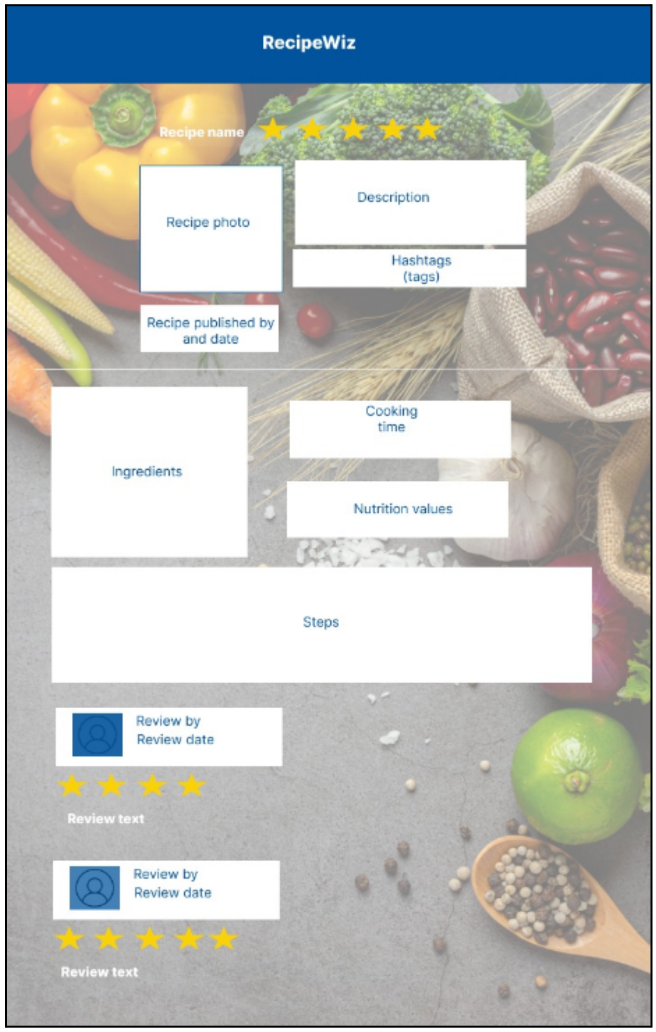
1. RecipeWiz is a single-page application. The landing page consists of a search bar and 2 carousels that show recently added recipes and the most popular recipes.



2. Upon searching by one of the three criteria - name/ingredients/tags - different recipes will be recommended based on the metadata of the recipes.



3. On selecting a recipe, all the recipe details will be displayed.



# Interpretability of Models

Each of the algorithms is tested on the following recipes and below are the recommended recipes using the different techniques. The recommended recipes can be seen to be similar to the test query, thus showing the effectiveness and relevance of the recommendations.

1. Test recipe name - ‘`chicken tortilla enchilada bake`’

TF-IDF

| name |
|---|
| enchilada lasagna |
| best easiest low fat chicken verde enchiladas |
| berdie s cheese enchilada casserole |
| speedy cheese and chicken enchiladas |
| cheese pork enchiladas |
| easy enchiladas |
| crock pot chicken enchilada |
| easy cheesy enchiladas |
| skillet chicken cheese enchiladas |

Pearson Correlation Coefficient

```
name
hamburger and green bean casserole
crock pot chicken   cornbread dressing
packs a wallop beef stew
morning breakfast panini
spicy italian hero crescent ring
crock pot cheeseburger supper  so easy
everyone loves chicken casserole
wedding lasagna
carnivore s lasagna
spicy king ranch chicken
```

1. Test recipe name - ‘`lemon sugar cookies`’

Sent2Vec

```
Similar dishes
              great grandma s chocolate zucchini cake
              rice crispy chocolate chip oatmeal cookies
              peppermint candy crisps
              frosted ginger cookies
              go big red cake
              easy cheesecake tarts
              gobble them up oatmeal raisin cookies
              lemon blueberry tea bread
              black pepper cake
              glazed hazelnut chocolate torte
```

1. Test ingredients - ‘`turkey sandwich cheese`’
2. Test ingredients - ‘`cake orange cream`’

Word2Vec (1)

| name |
|---|
| turkish towel sandwich |
| wasawich turkey and pepper jack |
| pilgrim sandwich |
| new york chicken burger |
| exotic grilled cheese |
| ham swiss roast beef and cheese wrap |
| italian gut busters |
| grilled gouda cheese sandwiches with smoked ha... |
| turkey ranch and cheese snacks |
| the tld sammy sandwiches |

Word2Vec (2)

| name |
|---|
| 4 points diet soda cake |
| my version of a sunshine cake ww style |
| just one more bite orange zucchini cake |
| berries on a cloud |
| basic trifle recipe |
| raspberry lemon cream cake |
| fresh orange cream cheese frosting |
| no bake orange cheesecake |
| strawberry lemon angel food trifle |
| creamsicle milkshake |

1. Test string - ‘`american orange cake with frosting`’

FAISS Approximate Nearest Neighbor

```
array(['special occasion pumpkin roll aka hazel s pumpkin cake roll',
       'no bake orange cheesecake', 'chiffon pumpkin pie',
       'tiramisu  cooking light',
       'caramel drizzled butterscotch toffee crunch pie',
       'goat cheese cheesecake w caramel sauce   english walnuts',
       'amazing tiramisu', 'bread machine sourdough cinnamon rolls',
       'a new yorker s real italian cheesecake',
       'pistachio shortbread cookies'], dtype=object)
```

# Model Deployment

**Python Notebooks**

1. Exploratory Data Analysis
   https://colab.research.google.com/drive/17W6-kN4g5Lw8hE-mTYw9UWdSDJMRj-wl?usp=sharing

2. Data Preparation
   https://colab.research.google.com/drive/1dOElQB5dlxFjBwszSwhXFcmTKax4af4N?usp=sharing

3. Data Modelling
   https://colab.research.google.com/drive/1XxL1BjGsvoRi0wWpnSm3cPCV41sgrH3J?usp=sharing

**Github Link**

https://github.com/soumyendra98/CMPE-256-Term-Project