Text Mining and Sentiment Analysis

Oumaima El Menni

September 2024

1 Introduction

In the era of big data, the ability to extract meaningful insights from unstructured text data has become increasingly important across various domains. Natural Language Processing (NLP) techniques have proven to be invaluable tools in this regard, enabling the classification and analysis of textual information with remarkable accuracy. This project leverages a combination of data scraping, machine learning, and sentiment analysis techniques to classify and interpret text data derived from a collection of food recipes and associated user reviews.

The project begins with the extraction of data from a large online repository of recipes, resulting in a structured dataset containing key information such as recipe names, ingredients, preparation directions, and user ratings. To categorize these recipes, two distinct classification approaches were employed. The first method utilized keyword-based classification, where predefined dictionaries of category-specific terms were applied to recipe names to determine their respective categories. This method laid the foundation for the subsequent application of a Support Vector Classifier (SVC), which further refined the classification by training on the results of the keyword-based method and employing various kernel functions to optimize performance.

In the second phase, sentiment analysis was conducted on user reviews to assess the general sentiment expressed towards selected recipes. A custom-built VADER sentiment analysis tool was constructed from scratch, incorporating domain-specific vocabulary to enhance its accuracy. This tool was then applied to extract sentiment scores and labels, providing valuable insights into user perceptions.

2 Research question and methodology

2.1 Research Question

The main research question addressed in this project is: How can we effectively classify recipes into categories using keyword-based methods and machine learning, and how can sentiment analysis of user reviews enhance our understanding of recipe feedback?

2.2 Goals and Objectives

- To construct a comprehensive dataset from a recipes website.
- To classify recipes into categories using a keyword-based method and an SVC classifier.
- To perform sentiment analysis on user reviews of selected recipes using a custom-built VADER sentiment classifier.

2.3 Methodology

2.3.1 Dataset Construction

Data was scraped from the recipes website "https://www.allrecipes.com/", resulting in a dataset with columns including recipe name, ingredients, directions, total time, rating, number of ratings, and URL.
The dataset contains 16252 different recipes.

Recipe Name	**Ingredients**	**Directions**	**Total Time**	**Rating**	**Number of Ratings**
Sloppy Joes for a Crowd	3 tablespoons vegetable oil, 6 pounds ground beef, ...	Heat 1 tablespoon oil in a large skillet over medium heat...	20 mins	4.6	7
Crawfish-Stuffed Jalapenos	8 ounces bacon, 8 ounces bulk lean breakfast sausage, ...	Place bacon in a large, deep skillet. Cook over medium-high heat...	1 hr	4.7	3
Breaded, Fried, Softly Spiced Tofu	1 (16 ounce) package extra-firm tofu, drained and pressed, ...	Cut pressed tofu into 1/2-inch thick slices; dredge in flour...	15 mins	4.3	280
Cabbage and Gnocchi	1 (16 ounce) package gnocchi, $\frac{1}{4}$ cup butter, ...	Bring a large pot of lightly salted water to a boil; add gnocchi...	20 mins	4.6	7
Buttermilk Sausage Gravy	1 (8 ounce) package pork link sausages (such as Bob Evans), ...	Place sausages and bacon drippings in a skillet over medium heat...	5 mins	2.8	5

Table 1: The first 5 rows of the dataset

2.3.2 Recipe Classification

Here are the steps that were followed:

Keyword-Based Method:
Recipes were classified into one of the following categories: main_course, soup, dessert, drink, bread, salad, side_dish, breakfast, and snack.
A dictionary of keywords was created for each category. Recipes were categorized based on the presence of these keywords in their names. This method provided a preliminary classification and established the initial labels used for subsequent machine learning model.

Support Vector Classifier (SVC) Architecture:

The Support Vector Classifier (SVC) functions by finding the optimal hyperplane that best separates different classes in the feature space. The primary goal of SVC is to maximize the margin, which is the distance between the hyperplane and the closest data points from each class, known as support vectors. These support vectors are crucial because they determine the position and orientation of the hyperplane.

SVC operates in the following way:

- **Hyperplane Selection:** The algorithm searches for a hyperplane that divides the data into two classes with the widest possible margin. This margin is maximized to ensure better generalization and robustness against new, unseen data. In cases where the data is not linearly separable, SVC uses kernel functions to transform the data into a higher-dimensional space where a linear separation is possible.
- **Kernel Functions:** To handle complex data distributions, SVC employs various kernel functions. The choice of kernel affects the algorithm's ability to model non-linear relationships:
  - **Linear Kernel:** This kernel is used when the data is linearly separable. It creates a straight-line decision boundary.
  - **Polynomial Kernel:** This kernel allows for non-linear boundaries by computing polynomial combinations of the input features. It is useful for capturing interactions between features.
  - **Radial Basis Function (RBF) Kernel:** Also known as the Gaussian kernel, the RBF kernel maps data into an infinite-dimensional space, making it effective for capturing non-linear relationships. It uses a parameter called gamma to control the influence of each training example.
  - **Sigmoid Kernel:** This kernel functions similarly to the activation function in neural networks, providing flexibility to model complex boundaries.
- **Regularization:** The SVC algorithm includes a regularization parameter, often denoted as C, which controls the trade-off between achieving a low error on the training data and minimizing the model complexity. A small C encourages a larger margin, potentially at the cost of a higher training error, while a large C aims to fit the training data more precisely, which may lead to overfitting.
- **Training and Optimization:** Training an SVC involves solving a convex optimization problem to find the optimal hyperplane that maximizes the margin while minimizing classification errors. This process can be computationally intensive, especially for large datasets, but it is crucial for achieving high performance.
- **Evaluation:** After training, the SVC model is evaluated using various performance metrics, such as accuracy, precision, recall, and F1-score, to assess how well it classifies new data points. These metrics help determine the effectiveness of the hyperplane in separating the classes and the overall quality of the classifier.

In our project, we applied the SVC algorithm to classify recipes into predefined categories. We experimented with different kernel functions, including linear, radial basis function (RBF) to determine which provided the best performance. To optimize the SVC's parameters, we conducted a Randomized Search with Cross-Validation, which allowed us to efficiently explore the parameter space and identify the best combination of kernel type, regularization parameter C, and other relevant hyperparameters.

2.3.3 Sentiment Analysis: VADER (Valence Aware Dictionary and sEntiment Reasoner)

This sentiment analysis tool is designed to analyze and interpret sentiments expressed in text, particularly useful for informal and social media content. VADER relies on a sentiment lexicon, where each word is assigned a sentiment score indicating its emotional valence. This lexicon includes scores for positive, negative, and neutral sentiments, and combines them into a compound score that reflects the overall sentiment of the text on a scale from -1 (most negative) to +1 (most positive). In our project, VADER was constructed from scratch to perform sentiment analysis on recipe reviews. The process involved tokenizing the text to extract individual words, scoring these words based on the VADER lexicon, and aggregating these scores to determine the overall sentiment of each review. This custom-built VADER tool allowed us to gain insights into user feedback by evaluating the sentiment expressed in the reviews of selected recipes.

3 Experimental Results

3.1 Keyword-Based Classification

This step involved checking each recipe name for specific keywords that matched different categories like "salad," "soup," "bread," "main_course," "dessert," "snack," "breakfast," "side_dish," and "drink." The categories were ordered by priority, so if a recipe name contained keywords from more than one category, it was classified under the highest-priority category. The recipe names were converted to lowercase to make the search easier. If no keywords matched, the recipe was labeled as "other." This method allowed to group the recipes based on their names, providing an initial classification that was later used for further analysis.

main_course	**dessert**	**other**	**side_dish**	**soup**	**bread**	**salad**	**breakfast**	**snack**	**drink**
5421	3250	2571	1122	1031	896	849	382	374	356

Table 2: Number of recipes per category

While the keyword-based classification method provided a straightforward way to categorize recipes, it also presented some challenges. One significant issue was the diversity of recipes from different countries, which meant that recipe names could be in various languages. This made it difficult to capture all relevant keywords, as the dictionary of keywords was limited and could not account for every possible language variation. Additionally, not all recipe names clearly indicated their category; some names were more abstract and did not include specific ingredients or category-related terms. This limitation meant that certain recipes were harder to classify accurately, leading to potential misclassification or a fallback to the "other" category.

3.2 Support Vector Classifier (SVC)

To optimize the Support Vector Classification (SVC) model, we employed Randomized Search to identify the best hyperparameters. This approach was chosen to balance efficiency and effectiveness in finding optimal parameters.

We focused on the following hyperparameters:

- **C (Regularization Parameter)**: Adjusted to control the trade-off between achieving a low training error and maintaining a simpler model. The values tested were $\{0.1, 1, 10\}$.

- **Kernel**: Specified to define the type of decision boundary. We experimented with {linear, rbf} kernels, which are commonly effective for a variety of tasks.
- **Gamma**: For the `rbf` kernel, `gamma` influences the shape of the decision boundary. We used {scale}, which adjusts `gamma` based on the number of features and training data variance.

The Randomized Search was configured with:

- **Number of Iterations (`n_iter`)**: Set to 6, allowing the search to explore a subset of the parameter space efficiently.
- **Cross-Validation Folds (`cv`)**: Set to 3 to provide a reasonable assessment of model performance while minimizing computation time.
- **Parallel Processing (`n_jobs=-1`)**: Enabled to utilize all available CPU cores for faster execution.

The Randomized Search identified the following best parameters:

- **Kernel**: Linear
- **C**: 10
- **Gamma**: Scale (`gamma` is not used with a linear kernel but was included in the search space)

The best cross-validation score achieved was approximately 97.88%, indicating strong performance on the training data. This optimal configuration of hyperparameters was then evaluated on a separate test set to confirm its effectiveness in a real-world scenario.

The results of the best model are:

Class	**Precision**	**Recall**	**F1-Score**	**Support**
bread	1.00	0.98	0.99	187
breakfast	0.95	0.97	0.96	76
dessert	0.99	0.98	0.99	645
drink	0.95	0.97	0.96	64
main_course	0.99	0.99	0.99	1103
salad	1.00	0.99	1.00	163
side_dish	0.94	0.98	0.96	217
snack	0.91	0.93	0.92	74
soup	1.00	1.00	1.00	208
accuracy			0.98	2737
macro avg	0.97	0.98	0.97	2737
weighted avg	0.98	0.98	0.98	2737

Table 3: Classification Report

The best Support Vector Classification (SVC) model demonstrated excellent performance across all recipe categories. The model achieved an overall accuracy of 98%, reflecting its high proficiency in categorizing recipes correctly. Precision and recall scores were particularly strong, with the model achieving perfect or near-perfect results for categories such as 'bread', 'salad', and 'soup'. For example, 'bread' and 'salad' categories had precision and recall rates of 100%, indicating flawless classification. While categories like 'snack' and 'side_dish' showed slightly lower performance with

precision and recall rates of 91% and 94% respectively, these results still signify robust model performance. The macro and weighted averages of precision, recall, and F1-score further underscore the model's ability to maintain high classification quality across various categories, making it a highly effective tool for recipe classification.

3.3 Sentiment Analysis: VADER Classifier

To better analyze the sentiment of recipe reviews, we began by expanding the VADER sentiment lexicon to include food-specific terms that were not originally covered. We manually added words like "tasty" and "stale," assigning them sentiment scores that reflect their positive or negative connotations.

The custom VADER classifier tokenizes each review, computes a sentiment score by summing the values of words found in the lexicon, and adjusts this score using context-based factors such as intensifiers, negations, and punctuation. Based on the final score, the review is classified as positive, negative, or neutral. This enhancement allowed for more accurate sentiment analysis, capturing the nuances of user feedback within the culinary context.

4 Concluding remarks

This project effectively demonstrates the application of advanced NLP techniques in classifying and analyzing textual data from food recipes and user reviews. The keyword-based classification method provided a foundational categorization, which was further refined using a Support Vector Classifier with optimized parameters, resulting in high classification accuracy. The sentiment analysis, performed using a custom-built VADER classifier, offered valuable insights into user feedback by capturing nuanced sentiment in reviews.