AutoBasket

"With AutoBasket, it's never been easier to get everything you need for a great meal!"

Dhiraj Kumar Sah
Dhru Sanjay Prajapati
Harshil Bhavsar
Keerat Singh

TABLE OF CONTENTS

ABSTRACT

AutoBasket, an innovation in the grocery and recipe management sphere, traces its roots back to the visionary minds of Larry and Veronica Smiles. Situated in the heart of Toronto, the company's headquarters pulsate with the energy of a city known for innovation and diversity. Larry and Veronica embarked on a mission to simplify the lives of households globally, envisioning an app that seamlessly connects recipes with essential grocery items.

Selected as the newest group of interns, we are set to contribute to the ongoing success of AutoBasket. The founders' vision was straightforward – to create an app automating grocery and recipe lists, streamlining the shopping experience for busy families and individuals. AutoBasket now extends its reach worldwide, helping households save time and energy on their weekly grocery runs. As an intern, your role spans across various areas of development for the organization, focusing on creating solutions to common industry issues.

INTRODUCTION

In our first week at AutoBasket, we were given the task of building a recommendation system for Italian dishes. Our goal was to create a system that would simplify the process of finding and preparing delicious Italian meals. To do this, we began by collecting and organizing data on various Italian dishes, ingredients, and recipes.

We then used this data to create a model that could compare and contrast different recipes, and provide personalized recommendations based on the user's preferences. We implemented this model using scikit-learn, a powerful machine learning toolkit, and Flask web development. The result was a user-friendly web app that allows users to choose a recipe from a dropdown menu and receive recommendations based on their selection.

Our proof of concept, demonstrated through the Flask web app, showcases our journey from raw data to actionable insights. As we continue to develop our skills in data-driven gastronomy, we are excited to contribute meaningfully to AutoBasket's mission of culinary convenience and innovation. The project code can be found on GitHub for a more detailed explanation, and you can check out our video presentation.

DATA PREPROCESSING

Data preprocessing is a crucial step in machine learning where we organize, clean, and transform raw data to make it suitable for training a machine learning model. The performance of the model greatly depends on the quality of the data it receives. Therefore, the goal of data preprocessing is to enhance the quality and relevance of the data.

In simpler terms, data preprocessing is like getting the data ready for the model to understand. Think of it as tidying up the data so that the model can learn from it effectively. When we clean and organize the data properly, it helps the model perform better and provide more accurate results. This step is essential because the success of the whole machine learning process relies on having good-quality data at the beginning.

1. DATA GATHERING

For our initial week, our primary objective was to establish a recommendation system for dishes based on user-preferred cuisine, specifically focusing on Italian cuisine. To accomplish this, we proactively sought a comprehensive dataset from CosyLab, an online platform specializing in computational Gastronomy with a primary focus on data-driven investigations into food and cooking. The acquired dataset encompasses crucial information such as dish names, ingredients, and associated items.

The dataset, conveniently packaged in a Zip folder, contains an array of CSV files, each catering to different facets of our analysis. These files include comprehensive details on recipes, ingredients, compound ingredients, and aliases for recipe ingredients.

Importantly, these distinct files are interconnected, forming a cohesive web of information detailing dishes, their constituent ingredients, and their respective components. This interconnected structure enables us to explore relationships between different elements, enhancing the depth and breadth of our dataset for a more nuanced and accurate recommendation system.

2. DATA PREPARATION

To accomplish our project goal and meet its requirements, we made use of all the dataset files, creating a dataframe with the necessary columns.

In the process of preparing the data, we went through each dataset file, combining them to gain a comprehensive overview. These datasets, however, contained redundant columns and information scattered across them, posing a potential challenge for further analysis and possibly deviating from the intended project scope.

To address this data challenge and streamline our efforts, we initially gathered all the datasets. Subsequently, we focused on removing redundant features from the dataset. This strategic approach not only helped us save time but also ensured that the dataset was optimized for our analysis.

For the current project week, our primary objective is to recommend Italian dishes to potential users based on their dish preferences. To achieve this, we collectively decided to prioritize ingredients as the key factor in recommending dishes. This approach aligns with our hypothesis and sets the stage for a targeted and effective analysis within the given timeframe.

3. DATA CLEANING

The dataset we obtained did not align with our desired format; additionally, it presented a considerable amount of noise in the form of NaN (empty values).

Each dish had multiple rows, each containing ingredients along with their respective recipe IDs. To address this structure, we opted to transform these multiple rows into columns, creating a more organized and streamlined representation.

The imperative to recommend dishes based on ingredients necessitated a dataset in a specific format. Consequently, we undertook a meticulous restructuring of the data, with the primary goal of enhancing its usability. This restructuring, informed by our intention to leverage cosine similarity, aimed to uncover relationships between ingredients. The refined dataset, organized in this specific manner, now positions us to recommend dishes that align closely with user preferences.

DATA VISUALIZATION:

Exploratory Data Analysis (EDA) is a crucial step in understanding and deriving insights from a dataset. It helps getting to the answer or hypothesis by determining the best way to manipulate data sources, hence making it easier to detect patters. In the context of your Italian cuisine dataset, EDA involves examining key aspects of the data to gain a comprehensive understanding.

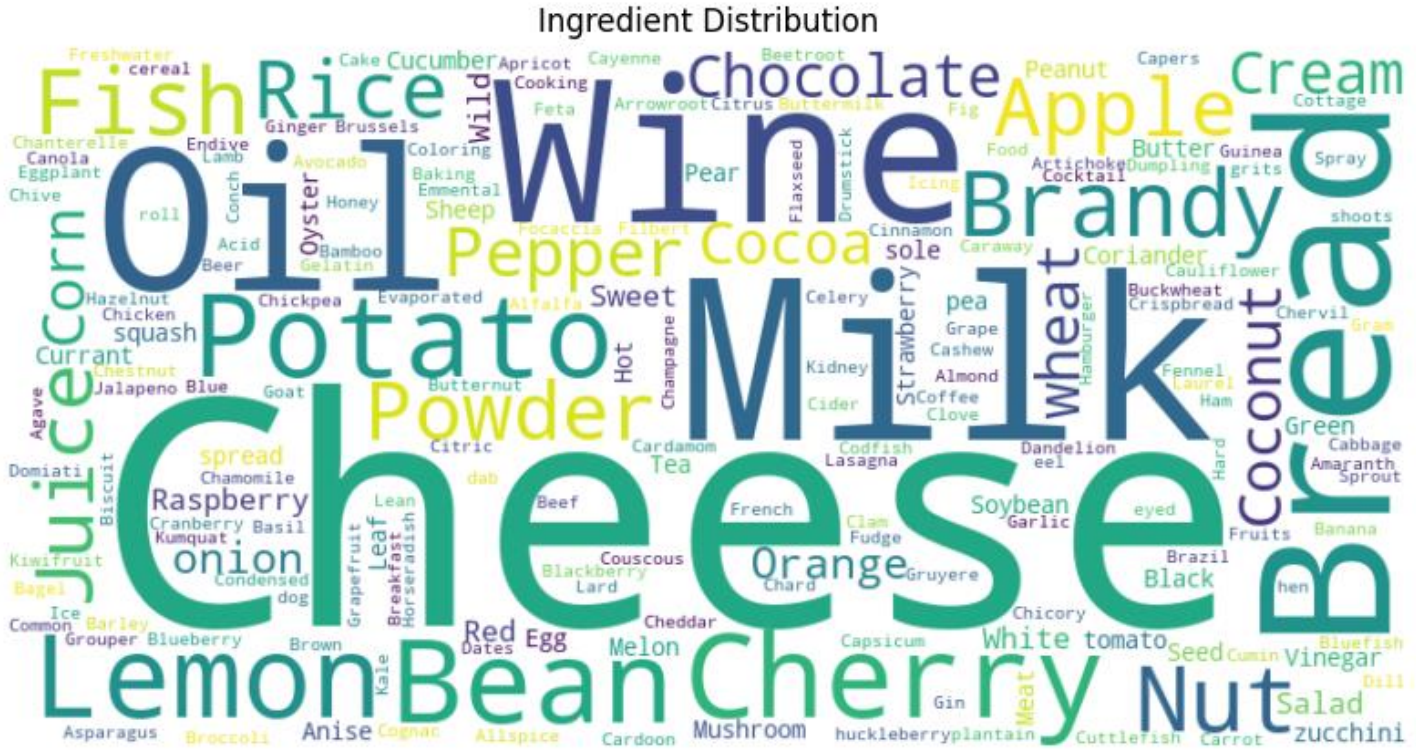

Figure 1 Wordcloud to represent different ingredients

To visually represent the variety of ingredients, present in the Italian cuisine dataset, we generated a WordCloud. With this you can highlight the frequency and prominence of different ingredients in the recipes. Larger, bolder words in the cloud indicate more frequently occurring ingredients, offering a quick overview of ingredients.

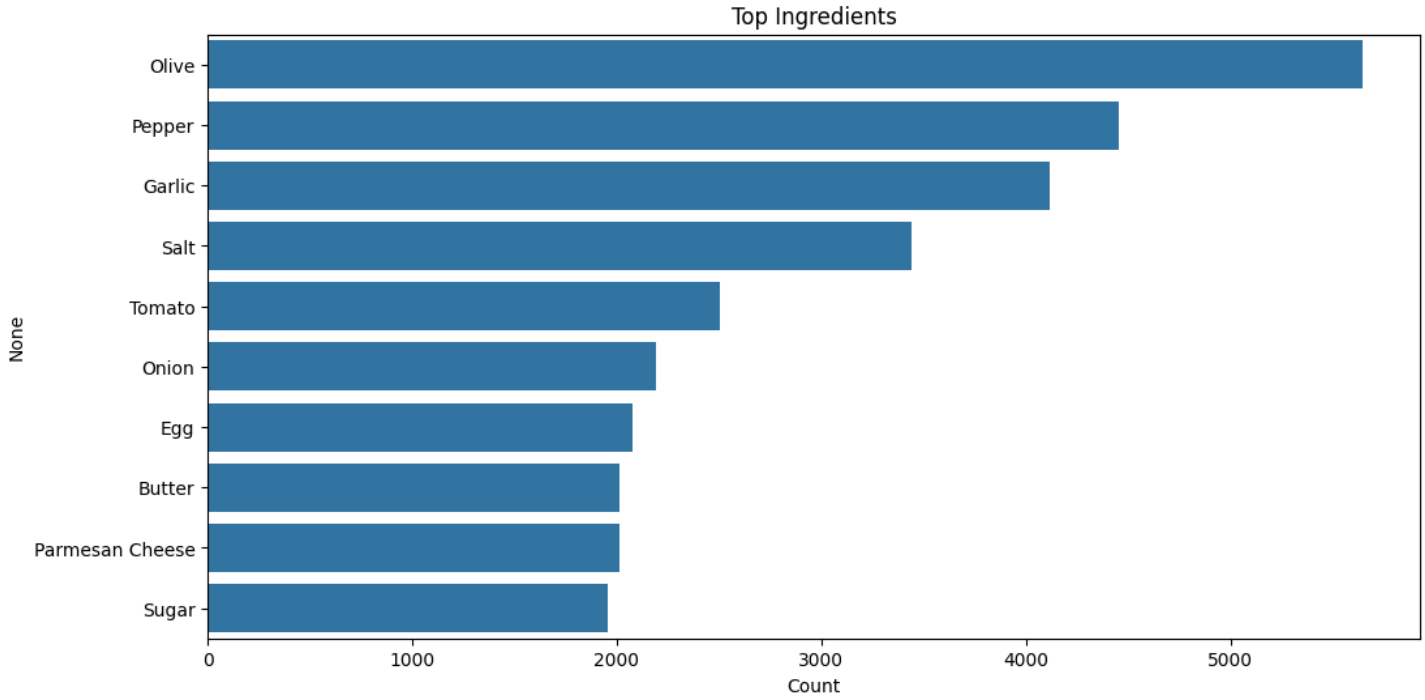

Figure 2 Top 10 ingredients

By analyzing the dataset, we identified the top 10 most common ingredients used in our Italian recipe dataset. This information is valuable for understanding the foundational elements that contribute to the distinct flavors and textures of Italian cuisine. Whether it's olive, tomatoes, garlic, or basil, recognizing the predominant ingredients provides insights into the culinary essence of these dishes.

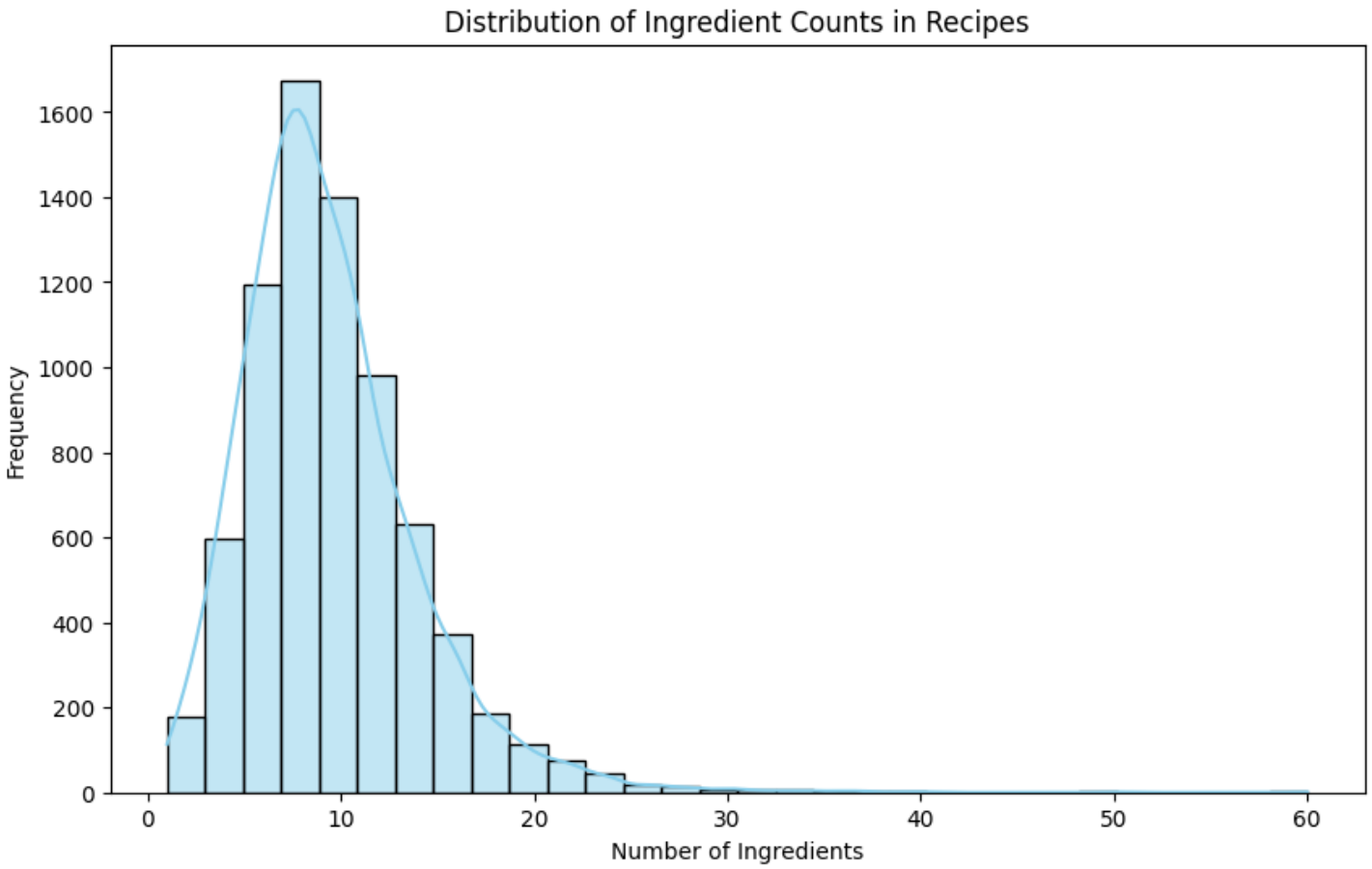

Figure 3 Distribution of ingredients

Examining the distribution of ingredient counts across the recipes in your dataset can provide valuable insights into the complexity and diversity of Italian dishes. You can visualize this distribution through histograms or statistical summaries. This analysis will help identify patterns, such as whether most recipes are simple with a few key ingredients or elaborate with a wide variety of components. Here the x axis represents number of ingredients and y axis represent the frequency of those individual ingredients.

RECOMMENDATION SYSTEM

After gathering, cleaning, and analyzing data, we built a personalized recommendation system for Luigi. As our dataset consists of many ingredients features according to the respective recipe, we decided to implement a dimensionality reduction technique on our ingredients columns to improve our recommendation model, which helped us escape the curse of dimensionality. Moreover, we also used cosine similarity to find similarities between our recipes and to finalize our recommendations. To implement such techniques, we have leveraged the use of the scikit-learn module of Python.

SCIKIT-LEARN

A robust and intuitive Python machine-learning toolkit, scikit-learn, also goes by the name sklearn, and provides a full suite of tools for creating and implementing machine-learning models. Regardless of their machine learning experience, users may easily experiment with different algorithms and strategies because of scikit-learn's consistent and user-friendly API.

Numerous supervised and unsupervised learning algorithms are implemented, including regression, classification, clustering, and dimensionality reduction. By building upon the NumPy and SciPy libraries, this library uses its array data structures and mathematical functions for practical calculation while guaranteeing interoperability with other scientific computing tools inside the Python ecosystem.

One of the key strengths of scikit-learn lies in its focus on simplicity, performance, and flexibility. It prioritizes ease of use, making machine learning accessible to users of all skill levels while optimizing algorithms for performance with implementations in low-level languages such as C and Python. scikit-learn also provides a comprehensive suite of tools for data preprocessing, model evaluation, and performance metrics, allowing users to seamlessly preprocess data, evaluate model performance, and fine-tune hyperparameters. Furthermore, the library supports custom implementations and extensions, enabling advanced users to create custom transformers, estimators, and pipelines tailored to their specific requirements.

All things considered, scikit-learn provides a strong foundation for creating, honing, and implementing machine learning models in Python, making it a valuable and essential tool for machine learning practitioners.

PCA (PRINCIPLE COMPONENT ANALYSIS)

PCA, which stands for principal component analysis, is a statistical technique for dimensionality reduction in data analysis and machine learning tasks. It is a method for converting high-dimensional data into a new coordinate system (a set of linearly uncorrelated variables known as principal components) in such a way that the most significant variance lies along the first coordinate (the first principal component), the second-greatest variance lies along the second coordinate (the second principal component) and so on.

The main goal of the PCA here is to reserve as much information as possible while decreasing the number of features. Several benefits of the PCA are dimentionality reduction, insight extraction, noise reduction and visualization. It can also be used in the customer segmentation, risk assessment and product development.

THE ROLE OF PCA IN OUR RECOMMENDATION SYSTEM

Applying PCA to our features helped us manage the features more efficiently and tackled the multidimensional issue of the dataset. In our system , we have reduced our ingredients matrix to retain only 10 principle components. We have used the n_components variable while feeding our ingredients features into the PCA.

The reduced matrix that we gained after undergoing the dimensionality reduction is more robust and easy to manipulate.

COSINE SIMILARITY

The basis of our recommendation system relies heavily upon the ingredients of the recipes. The system that we implemented finds the similarity between the recipes, in this case the Italian recipes, by comparing the recipes' ingredients. To compare the reduced ingredients matrix that we developed with the help of PCA, we have decided to use cosine similarity.

The main idea behind the use of cosine similarity is to measure the cosine angle between recipe vectors in multidimensional space. It can be calculated by using the following:

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Figure 4 The mathematical intuition behind cosine similarity

By leveraging cosine similarity, our recipe recommendation system can provide personalized and relevant recipe suggestions based on the ingredients specified by the user, facilitating the exploration and discovery of new and similar recipes.

GETTING THE RECOMENDATION

After utilizing the necessary machine learning algorithms to reach our goal of creating a personalized recipe recommendation system for Luigi, we have implemented a python function that combines the techniques of PCA and cosine similarity and gives the top 10 most similar recipe according to the recipe entered. The result of the function can be seen below.

```
recommendations = get_recommendations(9319)
print(recommendations)
```

	Recipe ID	Title
711	10229	Italian Wedding Soup II
2067	22748	Zucchini Ripieni (Stuffed Zucchini)
835	10382	Italian Wedding Soup I
45	9364	Mama's Italian Wedding Soup
1458	11101	Meat-Free Stuffed Shells
5217	39992	Asparagus and Parmesan Puddings
938	10504	A 20-Minute Chicken Parmesan
4892	39585	Risotto Croquettes with Mozzarella and Prosciutto
1560	11212	Easy Lasagna I
771	10299	The Soup with the Little Meatballs

Figure 5 Results of the recommendation function

ANALYSZING THE RECOMENDDATION

To analyze the accuracy of our recommendations system we tried to construct a function which evaluates the accuracy of a recipe recommendation system by computing the cosine similarity between actual ingredients and recommended recipes, and then averaging the similarity scores across multiple queries.

While calculating the accuracy we stumbled upon many issues that held us to examine the performance of the system. Moreover, while looking at the input recipe and recommendations that system made it can be said that the system is performing well.

FLASK WEB APP

To display our recommendations system in action, we have decided to develop a web app in the Flask architecture of Python. Flask is lighweight, adoptable, and contains a structure that can be used as an API and a Web application. We have decided to consume its capabilities and built a system that allows users to select the recipe from the dropdown menu and after the submission web application returns the top 10 recommendations for the chosen recipe. Below figures displays the functionality of our system.

PROOF OF CONCEPT

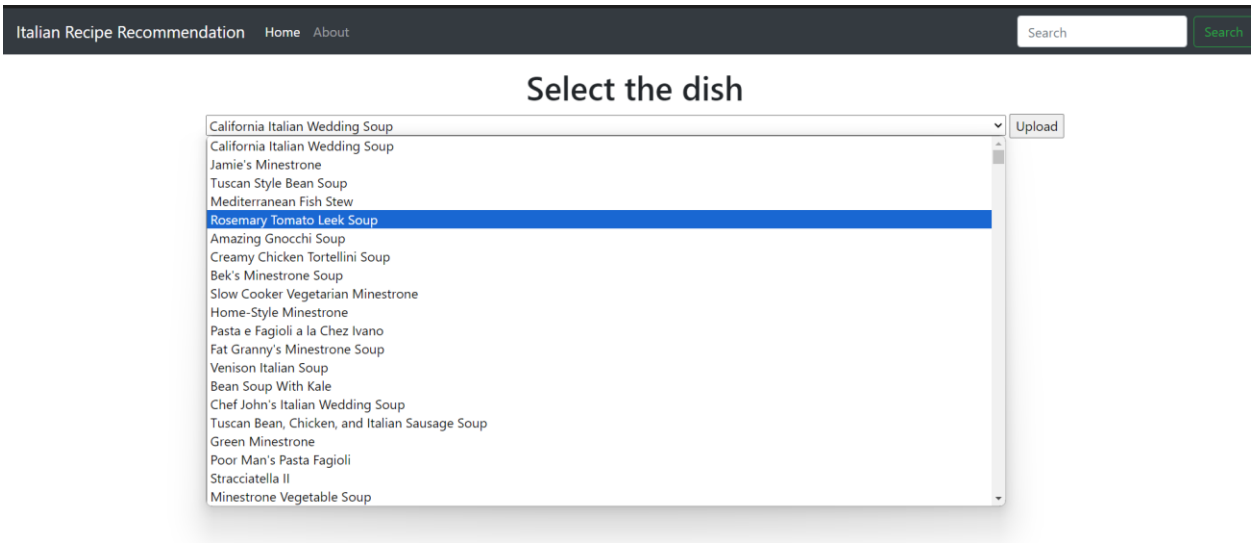

Figure 6 Recommendation system home page with dropdown menu

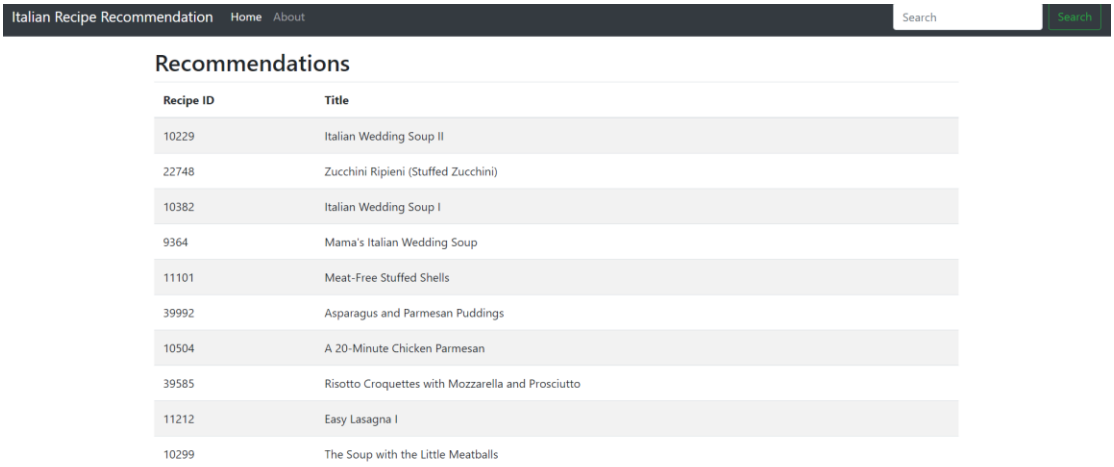

Figure 7 Recommendations based upon the selected recipe

CONCLUSION

Our inaugural week at AutoBasket has set the stage for a challenging yet rewarding internship. With a focus on developing a recommendation system for Italian dishes, we navigated the onboarding process, emphasizing the importance of both technical prowess and essential soft skills. AutoBasket, founded by Larry and Veronica Smiles, stands out as a beacon of convenience, automating grocery and recipe lists to simplify the shopping experience for modern households.

Our data-centric journey began with the acquisition of a comprehensive dataset from CosyLab, a computational gastronomy platform. Rigorous data preprocessing ensured the dataset's alignment with our project goals, and insightful visualizations, including WordClouds and histograms, provided a nuanced understanding of Italian cuisine ingredients. As we progress, armed with a refined dataset and a clearer perspective, we are well-positioned to contribute meaningfully to AutoBasket's mission of enhancing the culinary journey for individuals and families alike.

REFRENCES

[1] (n.d.). Retrieved from scikit-learn: https://scikit-learn.org/stable/

[2] (n.d.). Retrieved from GitHub: https://github.com/

[3] (n.d.). Retrieved from Python: https://www.python.org/

[4] */over-half-of-consumers-place-an-online-grocery-order-once-a-week*. (2024, 01 26). Retrieved from euroshop: https://mag.euroshop.de/en/2021/02/over-half-of-consumers-place-an-online-grocery-order-once-a-week/

[5] *build-a-recipe-recommender-system-using-python*. (2024, 01 22). Retrieved from javatpoint: https://www.javatpoint.com/build-a-recipe-recommender-system-using-python

[6] *candinavia-food-python-recommendation-systems*. (2024, 1 26). Retrieved from duarteocarmo: https://duarteocarmo.com/blog/scandinavia-food-python-recommendation-systems

[7] *Cosine_similarity*. (2024, 01 25). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Cosine_similarity

[8] IBM. (2024, 01 26). *What is exploratory data analysis?* Retrieved from IBM: https://www.ibm.com/topics/exploratory-data-analysis

[9] *palletsprojects*. (2024, 01 25). Retrieved from flask: https://flask.palletsprojects.com/en/3.0.x/

[10] *pca-practical-guide-principal-component-analysis-python/*. (2024, 01 25). Retrieved from analyticsvidhyaanalyticsvidhya: https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/

[11] Simplilearn. (2024, 01 24). *data-preprocessing-in-machine-learning-article*. Retrieved from Simplilearn - Online Certification Training Course Provider: https://www.simplilearn.com/data-preprocessing-in-machine-learning-article#:~:text=Data%20preprocessing%20is%20the%20process,algorithms%20to%20reduce%20its%20complexities

[12] tableau. (2024, 01 25). *data-visualization*. Retrieved from tableau: https://www.tableau.com/learn/articles/data-visualization

APPENDIX

Git hub link - https://github.com/Keerat-Singh/WIL_Week-1.git
Commercial video Presentation - Video Presentation - AUTOBASKET - WEEK-2