Data Science Project: What's cooking?

Andres Julian Vargas Medina, Daniel Felipe Barrera Suarez, Santiago Reyes

Abstract: This project aims to predict the cuisine type of a cuisine based on its ingredients, since recipes vary in ingredients across different parts of the world, reflecting cultural differences, knowing the recipes of a dish can help predict to some extent where in the world it is cooked most frequently and for that purpose, a predictive model will be used and configured.

In other external solutions, detailing how the configuration and setting of predictive models was highly considered, defined, and specified many parameters for the correct functioning of the model has been achieved.

In our case, we took an additional step: web scraping, to create new columns that add more value to the predictions and how these can be related to different characteristics to achieve better performance.

I. Background

Cooking is a fundamental part of human culture. It is a form of creative expression, a source of nourishment, and a central element of social gatherings. Throughout history, different cultures have developed their own unique culinary traditions, using specific ingredients and techniques to create distinctive dishes.

In the modern era, globalization has led to a greater blending of culinary cultures. This has resulted in increased availability of ingredients and dishes from around the world, enriching our culinary experiences. However, it has also made it more challenging to identify the origin of a dish or its culinary type.

This is where artificial intelligence (AI) can be helpful. AI models can be trained to identify patterns in ingredients and cooking techniques, enabling them to predict the type of cuisine from a list of ingredients. This capability has the potential to be used in a variety of applications, such as recipe recommendation, meal planning, and identification of fraudulent dishes.

Predicting cuisine type from a list of ingredients is a relatively new problem in the field of artificial intelligence. However, in recent years, there has been growing interest in this topic, and several different approaches have been proposed to address it.

Rule-Based Approaches: Early approaches to predicting cuisine type relied on rules. These approaches used a set of manual rules to determine the cuisine type from a list of ingredients. For example, a rule could state that if a recipe contains rice, chicken, and soy sauce, then it is likely to be a Chinese dish.

Rule-based approaches are relatively simple to implement but are not very scalable. As the number of ingredients and possible dishes increases, it becomes increasingly difficult to create and maintain a comprehensive set of rules.

Machine Learning-Based Approaches: Recent approaches to predicting cuisine type are based on machine learning. These approaches use machine learning algorithms to automatically learn how to predict the cuisine type from a list of ingredients.

Machine learning-based approaches are more scalable than rule-based approaches and can achieve higher accuracy. However, they require large training datasets to be effective.

Challenges and Opportunities: One of the main challenges in predicting cuisine type from ingredients is the lack of data. There is a shortage of large, high-quality publicly available datasets of recipes labeled with their cuisine type. This makes it difficult to train accurate machine learning models.

Another challenge is ambiguity. Sometimes, the same ingredient can be used in different types of cuisines. For example, rice can be used in Chinese, Indian, and Thai dishes. This can make it challenging for machine learning models to learn how to accurately predict cuisine type.

II. Methods

To address the problem, we conducted web scraping, data cleaning, column transformations, added new columns, performed feature engineering, evaluated models, selected the best one, and subsequently fine-tuned it for optimal performance.

Web Scraping: Initially, we conducted an extensive search to identify features that could help us obtain more columns, as the dataset from Kaggle only included the columns for ID, ingredients, and cuisine type. It was essential to gather more valuable information about our data. Therefore, we decided to collect information on the calorie content of the ingredients, allowing us to estimate the total calories in a recipe. Through this process, we discovered that the average calorie content varied between different cuisines. There are some values obtained:

Cuisine	**Average Calories**
irish	427.347826
chinese	370.970071
filipino	305.031788
vietnamese	258.073939

Data Cleaning: Fortunately, the Kaggle data did not contain any null values. However, both the data obtained from Kaggle and the data from the website we scraped (calories.info) had many errors and anomalies. For example, the calorie table values were strings, e.g., '210 cal'. Using regex, we extracted only the numbers and then converted that column to integers. Similarly, the ingredient columns in both the calorie table and the Kaggle table contained special characters that would complicate future processing by the learning models. It was vital to remove all special characters such as hyphens, periods, slashes, percentages, numbers, measurements, and anything that is not considered an ingredient to accurately determine the cuisine type.

On the other hand, we also had to change the values of the ingredients because the tables were provided as lists rather than separated by rows or columns. Therefore, we decided to join each array to facilitate a vectorization process, which we will discuss later.

Feature engineering: At this stage, we weren't sure which columns to add from the Kaggle data, as the process of classifying strings can be challenging. However, there was one particular value that could somewhat determine the cuisine of a recipe: the number of ingredients per recipe. Initially, this seemed like a simple and perhaps naive metric, but upon evaluating the percentages and examining the average number of ingredients used by each cuisine, it became an interesting feature to analyze. Thus, we can observe some of the data obtained by calculating the number of ingredients per recipe and then determining the average number of ingredients used by each cuisine:

Cuisine	**Average ingredients**
vietnamese	12.675152
chinese	11.982791
filipino	10.000000
irish	9.299850

From this information, it was very interesting to realize that the average number of ingredients did not correlate proportionally with the average number of calories. As observed from the previous table, Irish cuisine has the highest average calories but the lowest average number of ingredients. Conversely, Vietnamese cuisine has one of the lowest average calorie counts but one of the highest averages in terms of the number of ingredients.

For obvious reasons, we also added the column for calories per recipe, which allowed us to obtain the initial table shown in the web scraping section. This feature gave us greater insight into the contents of each recipe and how these characteristics generally correlate.

Model Selection and Training: For this part, we compared four models: Logistic Regression, K-Neighbors Classifier, Random Forest Classifier, and XGBoost Classifier. Since our target and the ingredient feature were of string type, it wasn't efficient to pass these data directly to the model. We had to vectorize the data, which is used to convert a collection of text documents into a matrix of token

features. This means converting the text into a numerical representation that machines can process.

Once we had the vectorized data and the encoded target, we passed them to the models for training. Here is a list of the accuracies:

Model	**Accuracy**
LogisticRegression	0.7824520237995475
KNeighborsClassifier	0.6412469622056483
RandomForestClassifier	0.7537920053632783
XGBClassifier	0.791921562054806

This is why we decided to use the XGBoost model for training with the rest of the data.

To train it, we created a pipeline that processes the string data using Countvectorizer and the numerical columns using Standard Scaler. This allows the model to process both text and numerical data seamlessly, ensuring it can handle different types of features without issues. As a result, we achieved an accuracy of 0.7907483449258359, which is almost 0.8.

III. Results

In the model evaluation, we conducted several tests in both column selection and parameter tuning, as well as in the division between test and training cases.

In the following table, we can find the different results when testing with different columns.

Features	**Score Accuracy**
ingredients, calories	0.7884857118913936
ingredients, ingredients quantity	0.7897427302438615
ingredients, calories, ingredients quantity	0.7907483449258359

From what we can observe, we realize that the best option is evaluating with all three columns.

Similarly, we tested the conversion of numerical features, conducting tests both by passing them through the Standard Scaler method and using the 'passthrough' parameter.

Parameter	**Score Accuracy**
StandardScaler	0,7911673510433253
passthrough	0,7907483449258359

With this, we noticed that we were able to improve the accuracy by 0.0004190061174894 more. It's not much, but it's an improvement.

Now with this, we proceeded with the other metrics, which include accuracy, precision, recall, and F1-Score.

Metric	**Result**
Accuracy	0.790748
Precision	0.789881
Recall	0.790748
F1-score	0.786602

As we can see, these metrics reflect the model's performance, which was quite good. Nearly all scores approach 1, indicating that the model predicted the test data quite accurately. This underscores good data preprocessing, including cleaning, column treatment, conversions, and more.

We also generated a heatmap based on the confusion matrix, illustrating true positives/negatives versus false positives/negatives. This provided a clearer visualization of how many test data points were correctly classified and how many were misclassified into different labels. The matrix was exactly 20 columns by 20 rows, representing the 20 possible cuisine classifications.

Additionally, we created a boxplot comparing the distribution of actual versus predicted data. Here, we observed that the actual data was mostly centered around a classification of 9, likely indicating Italian cuisine since most recipes fell into this category. However, the range of dispersion ranged from classification 13 (Mexican) to 6 (Greek). On the other hand, the predicted data showed a trend ranging between classification 13 and 7 (Indian).

References:

1. A Comprehensive Survey on Food Recipe Classification

2. Food Recognition Using Deep Learning

3. Recipe Classification with Convolutional Neural Networks

4. A Comprehensive Survey on Food Recipe Classification (https://arxiv.org/pdf/2212.05093)

5. Food Recognition Using Deep Learning (https://arxiv.org/abs/2004.03357)

6. Recipe Classification with Convolutional Neural Networks (https://arxiv.org/pdf/2310.15693)