District University Francisco José de Caldas
Faculty of Engineering

Notebook Report
Data Science Project: What's cooking?

Andres Julian Vargas Medina
20221020069
Daniel Felipe Barrera Suarez
20212020097
Santiago Reyes Gómez
20221020098

Professor
Carlos Andrés Sierra Virgüez

Data Science Introduction

Bogotá
2023

Introduction

The objective of this project is to develop a predictive model to identify the cuisine type based on a list of ingredients. This task involves several stages, including data collection, data preprocessing, feature engineering, model selection, model training, model evaluation, and some business questions. Each stage of this workflow is essential in developing a robust and accurate predictive model, ensuring that accurately predict cuisine types.

Data Collection

It is a fundamental part of any data science project, since the quality and quantity of the data collected directly affects the effectiveness of the machine learning model. In this project, two main data sources were used: web scraping for ingredient nutritional information and a recipe dataset from Kaggle.

Web Scraping:
Web scraping is a technique used to extract information from websites in an automated way. In this case, ingredient nutritional data was collected from multiple URLs using the requests library to make the HTTP requests and BeautifulSoup to parse the HTML content.This process allowed us to build a DataFrame with the ingredients and their respective calories. We performed web scraping processes on the calories.info website, which stores up to 40 tables on the calorie content of different types and categories of ingredients. From these tables, we decided to extract both the names of the ingredients or foods and the number of calories per portion. After performing the necessary data cleaning and table merging, we obtained 2,882 rows and 2 columns (ingredients, calories).

Kaggle data:
In addition to the data obtained through web scraping, we use a Kaggle dataset containing recipes and their ingredients. This data set was instrumental in training our classification model. For this project, we obtained data from the Kaggle competition "What's Cooking?", which provides a dataset of recipes categorized by cuisine types. This dataset specifically has 3 columns: id (int64), cuisine (object), ingredients (object), and a total of 39,774 values (rows).

Exploring and Understanding Data (EDA):
Data Exploration (EDA) allows us to better understand the structure and characteristics of the data we are using. We perform several operations to analyze the information contained in our Kaggle dataset.We also visualize the distribution of recipes by type of cuisine, which gives us a clear view of the amount of data available for each category.

Missing Values
We identify and count missing values in our data set to ensure data quality.

Combining data obtained through web scraping and the Kaggle dataset provided us with a rich and varied database to train our machine learning model. These data were fundamental for the prediction of the type of cuisine based on the ingredients, allowing us to build a robust and accurate model.

Data Preprocessing

Ensures that data is clean, normalized, and in a format suitable for use by machine learning models. In this project, preprocessing included data cleaning and preparing the data for model training.

Data Cleaning:
Data cleaning is essential to eliminate inconsistencies, errors and noise in the data. In our case, for each ingredient, all numbers were removed except for '7 up,' anomalies such as '(oz.)' were eliminated, dashes were replaced with blank spaces, and any other special characters like /, &, %, !, (,), ®, ™, €, as well as any type of measurement like ounces (oz), pounds, inches, or kilograms (kg) or words that were less than or equal to 2 characters long were removed.

Normalization:
A standardization process was carried out to change the format of the ingredients. This was done using the 'unidecode' tool, which removed any accents such as î, ç, é, â, í. Similarly, the 'WordNetLemmatizer' from the 'nltk' (Natural Language Toolkit) library was used, which normalizes words to reduce them to their base (native) form, for example, chips → chip. This was done to ensure consistency throughout the dataset.

Data Preparation:
After cleaning the data, the next step was to prepare the data for model training. This includes converting the ingredient list to a text string, which is necessary to use text vectorization techniques in the next step. Each ingredient list is then transformed into a unique text string, making it easy to further process using text vectorization methods such as CountVectorizer or TF-IDF Vectorizer.

Feature Engineering

Is a crucial step in the data science process, where relevant features are created, transformed, and selected from raw data to improve the performance of machine learning models. This process allows models to better capture the underlying relationships and patterns in the data, increasing the accuracy and effectiveness of predictions.

In our project, we performed several feature engineering tasks to enrich the data set and provide additional information to our cuisine prediction model.

-Amount of ingredients:
First, we create a new column containing the number of ingredients in each recipe. This feature can be useful to differentiate types of cuisine that tend to use more or fewer ingredients. Then, for each list of ingredients we apply the 'len' function to count the number of ingredients.

-Estimated Calories:
Additionally, we added a feature that estimates the sum of calories of the ingredients in each recipe. To do this, we first perform data collection (scraping) to obtain the calorie information of each ingredient and store this data, next, we create a function to calculate the sum of calories of the ingredients present in each recipe. The ‘estimated_calories_sum’ function takes a list of ingredients, checks if each ingredient is in our scraping data set, and if so, sums the corresponding calories. Then, we apply this function to each recipe to obtain the estimated calories column.

These additional features help our model capture relevant information about recipes, such as the number of ingredients and estimated calorie content, which can improve the accuracy of our predictions. Through feature engineering, we enrich our data set and provide the model with more context and information to learn more complex and precise patterns.

Model Selection

In this project, we evaluate several classification models to predict cuisine type from a list of ingredients. The models considered were: Logistic Regression, KNeighborsClassifier, RandomForestClassifier and XGBoostClassifier. Next, we describe the steps we follow to prepare the data, train and evaluate each model, and finally select the most appropriate model.

Splitting the Data Set into Training and Test Sets:
First, we prepare our data set and divide it into training and test sets. This allows us to evaluate the performance of the models on data that they have not seen during training. The 4 models are trained and their accuracies are evaluated.

Model Comparison:
We compare the models using several performance metrics: Accuracy, precision, recall, and F1-score. We created a DataFrame to facilitate comparison and visualize the results in a bar graph.
After evaluating and comparing the models, we determined that the XGBoost model is the most suitable for our project due to its superior performance on all key metrics. XGBoost not only obtained the best accuracy, but also showed high scores in precision, recall and F1-score, making it ideal for the task of predicting the type of cuisine from a list of ingredients.

Choosing the XGBoost model ensures that our predictions are accurate and reliable, allowing us to provide ingredient-based cuisine recommendations with a high level of accuracy.

Model Training

Model training is a critical phase in the data science process, we use our preprocessed data and selected features to train a machine learning model. Below are the key steps we took to prepare the data, train the model, and evaluate its performance.

Feature Selection:
We select the relevant features for our model: the list of ingredients, the estimated calories, and the number of ingredients.

Data Scaling - Label Encoder:
To prepare our labels (cuisine types), we use 'LabelEncoder' to convert categorical labels into numerical values. Next, we split our data into training and test sets, using 30% of the data for testing and 70% for training.

Data Transformation with Count Vectorizer and Standard Scaler:
We use a ‘ColumnTransformer’ to apply different transformations to different columns. In this case, we use ‘CountVectorizer’ to transform the list of ingredients into a numerical representation and passing the numerical characteristics through the ‘StandardScaler’, this is used to standardize the features of a dataset, meaning each feature has a mean of 0 and a standard deviation of 1. This is useful because many machine learning algorithms perform better or converge faster when features are scaled in this manner.

Model Training:
We train the model using the training data set and creating a pipeline that first applies the preprocessor to our data and then trains an XGBoost model

Model Prediction and Evaluation:
Finally, we use the trained model to make predictions on the test data set and calculate the accuracy of the model. Model accuracy indicates how well the model can predict the type of cuisine based on the ingredients provided.

Model Evaluation

Model evaluation is an essential stage in the development of machine learning models, as it allows us to measure their performance and effectiveness in the prediction task.

Bar Plot with Metrics:
First, we convert the model evaluation metrics into a DataFrame for easy viewing. We calculate the accuracy, precision, recall and F1-score for our model and display them in a bar graph. This bar chart shows the different performance metrics of the model, providing a clear view of its effectiveness in terms of precision, precision, recall and F1-score.

Confusion Matrix Heat Map:
The confusion matrix allows us to visualize the performance of the model in terms of true positives, false positives, true negatives, and false negatives. This is particularly useful for identifying classes that the model may be predicting incorrectly. We generated and visualized the confusion matrix using a heat map.This heat map allows you to visually identify which cuisine types are most difficult for the model to predict and where classification errors occur.

Box Plot on Real Values vs. Predicted:
To compare the distribution of actual values versus values predicted by the model, we use a boxplot. This helps visualize how the model performs in different classes and if there is any bias in the predictions.
This boxplot provides a clear view of the distribution of predicted values compared to actual values, helping to identify any significant discrepancies.

Evaluation of the model with different metrics and visualizations confirms that our XGBoost model performs well in the task of predicting cuisine type based on ingredients. The combination of precision, precision, recall, and F1-score, along with confusion matrix and boxplot visualizations, provides us with a comprehensive understanding of the model's behavior and its areas for improvement.