Cuisine Classification Based On Ingredients

Vinayak Arora
vinayak21112@iiitd.ac.in

Vinayak Goel
vinayak21113@iiitd.ac.in

Abstract

The aim of this project is to create an accurate Cuisine classifier with the help of Machine Learning. The classifier should be able to accurately predict the cuisine of the recipe based on the ingredients used in that recipe. With the help of multiple Machine Learning models and feature engineering, we aim to achieve this. The codebase can be accessed at [1]

1. Introduction

There are 195 countries in the world, and each country has its own set of cultures,traditions and beliefs. Naturally every country has unique food habits that is best described by popular cuisines there. Each cuisine has recipes that represent the cultural heritage of the region and the ingredients used in those recipes act as a solid differentiation between different cuisine. Identifying the relation between ingredients and cuisines can help design food recommender systems, recipe recommenders and personalized diet planner. Thus, cuisine classification based on ingredients used in a recipe can open door for further innovations.

2. Literature Review

Food Prediction based on Recipe using Machine Learning Algorithms [2] This article goes in depth about the different recipe platforms and the enormous amount of data which is present and can be extracted from such sites. They create a predictive model be extracting all the relevant data, and then pre-processing it, before utilising them in a bunch of Machine Learning based algorithms such as Fuzzy Rule, Neural Network, Naive Bayes and CNN. They noticed that CNN was the best performing on the pre-processed dataset.

Machine Learning Model for Predicting the Cuisine Category from a Dish Ingredients [3] The researchers in this article propose the utilisation of Support Vector Machines and Associative Classification. They did data preprocessing on the ingredients list be removing less relevant keywords, phrase mapping and plural to singular mapping, which allowed them to extract relevant ingredient phrases and infromation to train the models. They tested eight to ten different models such as K nearest Neighbours, Naive Bayes, Support Vector Machines, Decision Tree, XG Boost and a lot more, and came to the conclusion that SVM was the best with an accuracy of around 81 percent.

What Cuisine? - A Machine Learning Strategy for Multi-label Classification of Food Recipes [4] firstly conducted an Exploratory data analysis by analysing the frequency of ingreidents in different cuisines. They found out the noise distribution over the dataset, and the average recipe size and deviations over the cuisines. They utilised a custom tf-idf scoring method which ensured that the ingredients list was vectorised in a way for the machine learning algorithms to work efficiently. They also conducted pre-processing over the ingredient lists, to extract the relevant information, and tested Random Forest and Logistic Regression over the Test and Training set. They changed the different parameters for tuning, and found out that Logistic Regression performed similar or better than Random Forest.

3. Dataset

3.1. Dataset Overview

The following dataset was used for the project - [5]

The Dataset has 39774 rows and 3 columns. The features are as follows:

1. Recipe ID: ID of the Recipe in the Dataset
2. Cuisine: The Cuisine the particular set of ingredients belong to
3. Ingredients: The List of ingredients

3.2. Exploratory Data Analysis

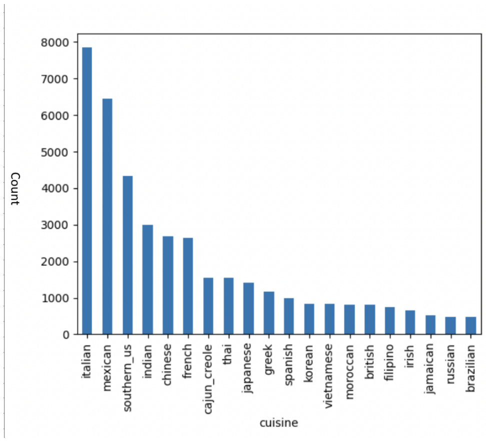


Figure 1. Cuisine Class Distribution

Figure 1 represents the 'cuisine' class distribution in the dataset. The distribution highlights imbalanced distribution with Italian cuisine having the highest representation (approximately 8000 data points) while jamaican, russian, brazilian having the lowest (approximately 900 data points).

olive oil
salt
onions
garlic
water
sugar
butter
garlic cloves
indian
cajun_creole
chinese
mexican
italian
southern_us
moroccan
spanish
russian
korean
irish
filipino
vietnamese
japanese
thai
greek
french
jamaican
british
brazilian

(a) Ingredient Word Cloud

(b) Cuisine Word Cloud

Figure 2. Word Cloud

Figure 2a represents a word cloud for ingredients present in ingredient list in the entire dataset. Salt, olive oil, onions, garlic, sugar, water are the most prominent ingredients.Figure 2b epresents a word cloud for cuisines present in the dataset. Italian, Mexican are the most prominent cuisines.

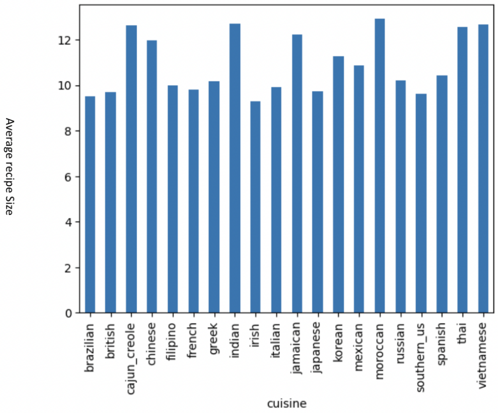
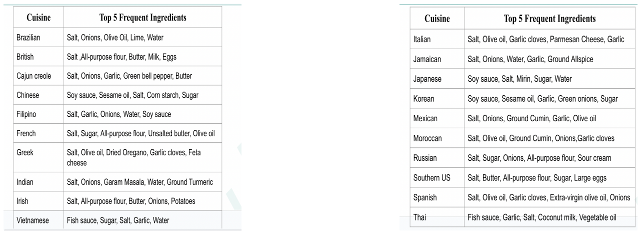


Figure 3. Average Recipe Size in Cuisines

Figure 3 represents the average recipe size for all the cuisines. The average recipe varies from 8.5 for some cuisines and 12.5 for others.



Cuisine	Top 5 Frequent Ingredients
Brazilian	Salt, Onions, Olive Oil, Lime, Water
British	Salt ,All-purpose flour, Butter, Milk, Eggs
Cajun creole	Salt, Onions, Garlic, Green bell pepper, Butter
Chinese	Soy sauce, Sesame oil, Salt, Corn starch, Sugar
Filipino	Salt, Garlic, Onions, Water, Soy sauce
French	Salt, Sugar, All-purpose flour, Unsalted butter, Olive oil
Greek	Salt, Olive oil, Dried Oregano, Garlic cloves, Feta cheese
Indian	Salt, Onions, Garam Masala, Water, Ground Turmeric
Irish	Salt, All-purpose flour, Butter, Onions, Potatoes
Vietnamese	Fish sauce, Sugar, Salt, Garlic, Water

Cuisine	Top 5 Frequent Ingredients
Italian	Salt, Olive oil, Garlic cloves, Parmesan Cheese, Garlic
Jamaican	Salt, Onions, Water, Garlic, Ground Allspice
Japanese	Soy sauce, Salt, Mirin, Sugar, Water
Korean	Soy sauce, Sesame oil, Garlic, Green onions, Sugar
Mexican	Salt, Onions, Ground Cumin, Garlic, Olive oil
Moroccan	Salt, Olive oil, Ground Cumin, Onions,Garlic cloves
Russian	Salt, Sugar, Onions, All-purpose flour, Sour cream
Southern US	Salt, Butter, All-purpose flour, Sugar, Large eggs
Spanish	Salt, Olive oil, Garlic cloves, Extra-virgin olive oil, Onions
Thai	Fish sauce, Garlic, Salt, Coconut milk, Vegetable oil

Figure 4. Top 5 Ingredients in each cuisine

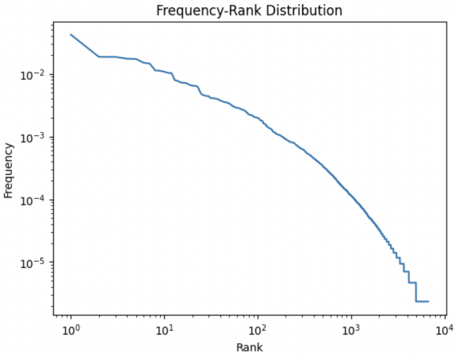


Figure 5. Frequency Rank Distribution

Figure 5 shows that the a certain set of ingredients dominate through most of the cuisines, while there are certain specialised ingredients, which are used by only a couple of cuisines

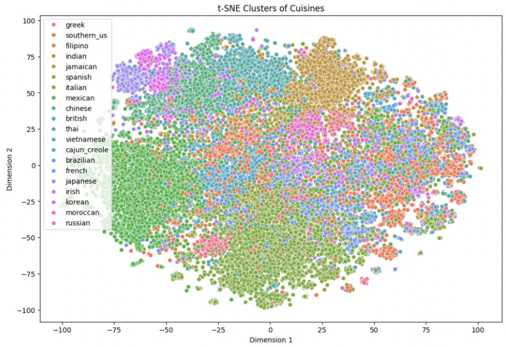


Figure 6. Dimensionality Reduced Visualization

Figure 6 represents the cuisine clusters in lower dimensions based on ingredient (after Tfidf vectorization). This is achieved by using t-SNE, which is useful in visualising high dimensional data in two or three dimensions The cuisines where clusters overlap the most are (Thai, Japanese, Chinese, Vietnamese, Korean), (Greek, French, British). Since there are no well defined linearly separable clusters in 2D space, simple linear models won't achieve very high accuracy.

4. Methodology

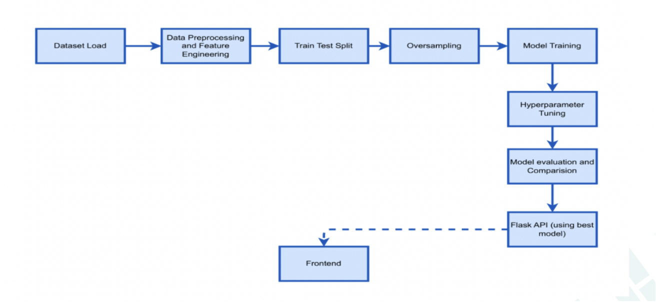

Dataset Load
Data Preprocessing and Feature Engineering
Train Test Split
Oversampling
Model Training
Hyperparameter Tuning
Model evaluation and Comparision
Flask API (using best model)
Frontend

Figure 7. Methodology

4.1. Data Preprocessing and Feature Engineering

The steps followed for text preprocessing [6] were :

1. Word Tokenization - This method takes a sentence and creates a list of tokens (words) in the sentence. This allows efficient handling of each word in the sentence.

2. Conversion to lowercase and removing whitespaces - This was achieved by using lower() and strip() function

3. Removing punctuations, numbers - This was achieved using regex matching, by keeping only alphabets and replacing everything else by empty string.

4. Removing stop words - This was achieved by using stopwords corpus provided by nltk and any token found to be a stopword was discarded

5. Removing 'ingredient form' words like chopped, cooked, sliced - This was achieved by creating a set for such words and checking if any token belonged to this set, then it was discarded

6. Lemmatization / Stemming to convert word to its base form - This was achieved by using WordNetLemmatizer provided by nltk.

For Feature Engineering , two different methods (TfidfVectorizer, Word2vec) were used separately to transform text into numerical data. **TfidfVectorizer** finds the weight of an item, through the frequency with which it existed in the dataset. It tests the relevancy of the ingredient against the ingredients list, and the entire dataset. The more common ingredients become less significant, while the ingredients specific to a particular cuisine become important. **Word2Vec** on the other hand, tries to find the relation of one ingredient against the ingredients list, and the dataset. it creates a vector representation of the words, better understanding the context of the ingredient with respect to the cuisine. It is more semantic and contextual. The final analysis was done by comparing the accuracy achieved by both methods. The original TfidfVectorizer transformed dataset and dimensionally reduced dataset using PCA, which retains the important information of the dataset over the reduction of dimensions, were both used for model accuracy comparison.

4.2. Oversampling

The class imbalance in the cuisine column was clearly visible during EDA, so we used Oversampling techniques like ADASYN and SMOTE to ensure that all classes have approximately equal data points.

Effect of oversampling :

Before oversampling, class distribution : [383 647 1218 2163 619 2096 926 2401 516 6271 435 1139 664 5102 655 400 3472 807 1224 681]

After oversampling, class distribution : [6341 6371 6372 6239 6228 6351 6255 6204 6351 6271 6238 6176 6214 5908 6219 6286 5840 6163 6269 6225]

4.3. Model Training And Details

The following baseline models were trained using each of the TfidfVectorizer, Word2vec datasets split in 80:20 train:test ratio :-

4.3.1 Support Vector Machine (SVM)

Support Vector Machines are based on the algorithm for optimising the maximum separating hyperplane between the different classes. There are different parameters such as C, the regularisation parameter, or the type of kernel (eg. rbf, linear) which can be fine tuned, to improve the accuracy of the model.

4.3.2 Neural Networks

Neural Networks is a Machine Learning Technique where we train a set of artificial neurons, over the training data, where it recognises hidden patterns and features, and conducts its own feature extraction, setting weights for optimising the accuracy of prediction. It can also be fined tuned by changing the Learning Rate, Loss function, Optimiser, Number of Epochs, etc.

4.3.3 Perceptron

Perceptron is a type of Neural Network, where weights and biases are trained on the basis of classification of an input. It is efficient in predicting on data which is linearly separable.

4.3.4 Logistic Regression

Logistic Regression is one of the classification models, where a probability of a data point belonging to a particular class is calculated. It uses the sigmoid function, and has different parameters which can be optimised to improve the accuracy of the model.

4.3.5 Random Forest

Random Forest is a Machine Learning Algorithm which utilises a combination of decision trees to train on the data, after which it utilises different techniques such as Majority Voting, to find out the class most of the decision trees think the data point should be a part of. It has a lot of parameters such as Tree depth, Number of Estimators, Min Node split etc, which can vastly affect the performance of the algorithm.

4.3.6 Multi Layer Perceptron

Multi Layer Perceptron , or MLP utilises the concept of Hidden Layers, where the existing Perceptron Algorithm is improved by adding additional layers, which compute further weights and biases on the training set, vastly improving the prediction accuracy of the model on the dataset using techniques like forward and backward propogation.

4.3.7 K Nearest Neighbors (KNN)

K nearest neighbours algorithm works on the concept of clustering, where similarly behaving data is clustered together under a label. It groups the different data points together, and predictions are based on the basis of voting by the clusters, on believing which cluster it is most likely to be a part of.

4.3.8 Naive Bayes

Naive Bayes is a classification algorithm which utilises Bayes Theorem under the Naive assumption, where independent features can be calculated as a product of individual features. It calculates the probabilities for which class the data point is most likely to be attached to.

4.3.9 Decision Tree

Decision Tree Algorithm is an easy to visualise algorithm which works on the concept of splitting decisions and impurity, where the data is splitted on the best feature for that iteration. It can lead to overfitting, and hence fine tuning of parameters is very important in a decision tree.

4.3.10 XGBoost

XGBoost is a gradient boosting technique, which optimises the algorithm, while keeping the accuracy high. It is useful in finding early stopping points when there are no improvements, or finding different strategies to employ on training the gradient trees, as it uses the concept of parallel trees.

4.3.11 Ensemble Model

Ensemble Models use the approach of combining a bunch of individual models, to product better predictions. The best baseline model accuracy was achieved for SVM. SVM was then trained and tested using oversampled datasets as well as PCA (dimensionality reduced) dataset.

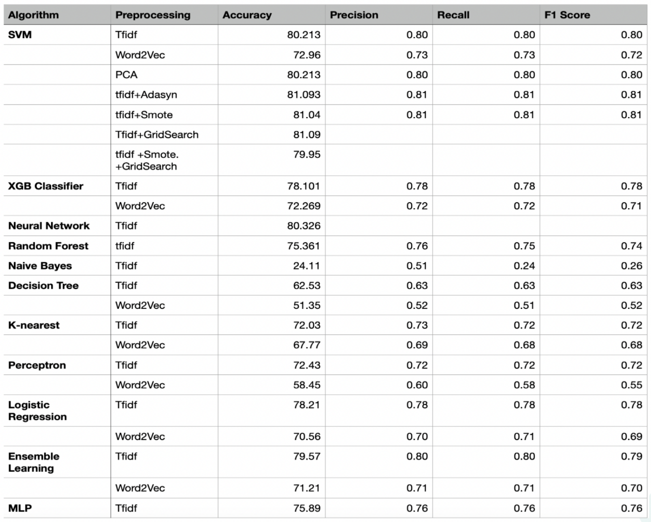
4.4. Hyperparameter Tuning

SVM trained using TfidfVectorizer transformed dataset oversampled with SMOTE / ADASYN achieved the highest accuracy among baseline models. SVM models parameters were then tuned using GridSearch with the following grid of parameters : param_grid = ’C’: [0.1,1,10],’gamma’: [’scale’,’auto’],’kernel’: [’poly’, ’rbf’, ’sigmoid’]. The best of parameters achieved were : ’C’: 10, ’gamma’: ’scale’, ’kernel’: ’poly’.The hyperparameter tuned model then underwent cross validation to detect and prevent overfitting

4.5. Frontend and API

The Best performing machine learning model was saved as .pkl file, to create a flask application, where a POST API controller was defined to processes the json body of the http request, extract the list of ingredients and predict the cuisine using the model. The API is hosted on pythonanywhere.com.The Frontend was created using the React and CSS which has a simple, easy-to-use interface where the user can add ingredients. On pressing ‘Submit’, a request is send to the flask API, which processes these ingredients and returns the predicted cuisine. This cuisine is then displayed on the frontend. The Frontend has been hosted on Render.

5. Results



Algorithm	Preprocessing	Accuracy	Precision	Recall	F1 Score
SVM	Tfidf	80.213	0.80	0.80	0.80
	Word2Vec	72.96	0.73	0.73	0.72
	PCA	80.213	0.80	0.80	0.80
	tfidf+Adasyn	81.093	0.81	0.81	0.81
	tfidf+Smote	81.04	0.81	0.81	0.81
	Tfidf+GridSearch	81.09			
	tfidf +Smote. +GridSearch	79.95			
XGB Classifier	Tfidf	78.101	0.78	0.78	0.78
	Word2Vec	72.269	0.72	0.72	0.71
Neural Network	Tfidf	80.326			
Random Forest	tfidf	75.361	0.76	0.75	0.74
Naive Bayes	Tfidf	24.11	0.51	0.24	0.26
Decision Tree	Tfidf	62.53	0.63	0.63	0.63
	Word2Vec	51.35	0.52	0.51	0.52
K-nearest	Tfidf	72.03	0.73	0.72	0.72
	Word2Vec	67.77	0.69	0.68	0.68
Perceptron	Tfidf	72.43	0.72	0.72	0.72
	Word2Vec	58.45	0.60	0.58	0.55
Logistic Regression	Tfidf	78.21	0.78	0.78	0.78
	Word2Vec	70.56	0.70	0.71	0.69
Ensemble Learning	Tfidf	79.57	0.80	0.80	0.79
	Word2Vec	71.21	0.71	0.71	0.70
MLP	Tfidf	75.89	0.76	0.76	0.76

Figure 8. Model Comparision

For our analysis and conclusion, we extract the relevant features from a classification model such as Accuracy (Proportion of Correct Classifications) , Precision (Actually Positives), Recall(True Positives) and F1 score(Harmonic Mean of Precision and Recall). They tell how well the machine learning model performs under different criteria, and gives us insight whether the model overfitted, or gave a lot of false positives.

6. Conclusion

SVM performs the best for the balancing the different metrics such as Precision, Recall and F1 Score for the different cuisine classes, which was calculated using the Weighted Average over the classes.Neural Network and Logistic Regression come close, with a performance better than the other Regressors and Classifiers, but not as good as the SVM model. Ensemble Model was another model which behaved similarly.Tfidf Vectorisation performed better than Word2Vec in all the models they were tested for. Similarly Adasyn and Smote, the two sampling techniques utilised to overcome the class imbalance, showed a slight improvement over the base model. However there was a Computation to Accuracy tradeoff, as the miniscule improvement was accompanied by a large increase in model training time

7. Future Work

The classification model obtained can be used in the future in a number of ways.A recipe recommendation system based on ingredients and cuisine.A personalized diet recommendation system based on favourite ingredients and cuisine combination. This model can also help identifying common ingredients across recipes in cuisine paving the way for efficient recipe infusion between cuisines

References

[1] VinayakG311. Cgas-mini-project. https://github.com/VinayakG311/CGAS-MINI-Project, 2024.

[2] https://ieeexplore.ieee.org/document/10250758.

[3] https://ieeexplore.ieee.org/document/10087436.

[4] https://jmcauley.ucsd.edu/cse190/projects/fa15/022.pdf.

[5] Kaggle. https://www.kaggle.com/competitions/whats-cooking/data?select=train.json.zip. Dataset.

[6] Awaldeep Singh. Understanding the essentials: Nlp text preprocessing steps! https://medium.com/@awaldeep/understanding-the-essentials-nlp-text-preprocessing-steps-b5d1fd58c11a, 2023.