Determining the Type of Cuisine based off a food's ingredients

Henry Truong
A10773987
hetruong@ucsd.edu

ABSTRACT

For assignment 2, I decided to analyze one of the datasets provided by Yummly for a Kaggle competition. The dataset provided 39k training points and 10k test set. The goal of this project is to see if we can predict the type of cuisine the food is based solely off its ingredients.

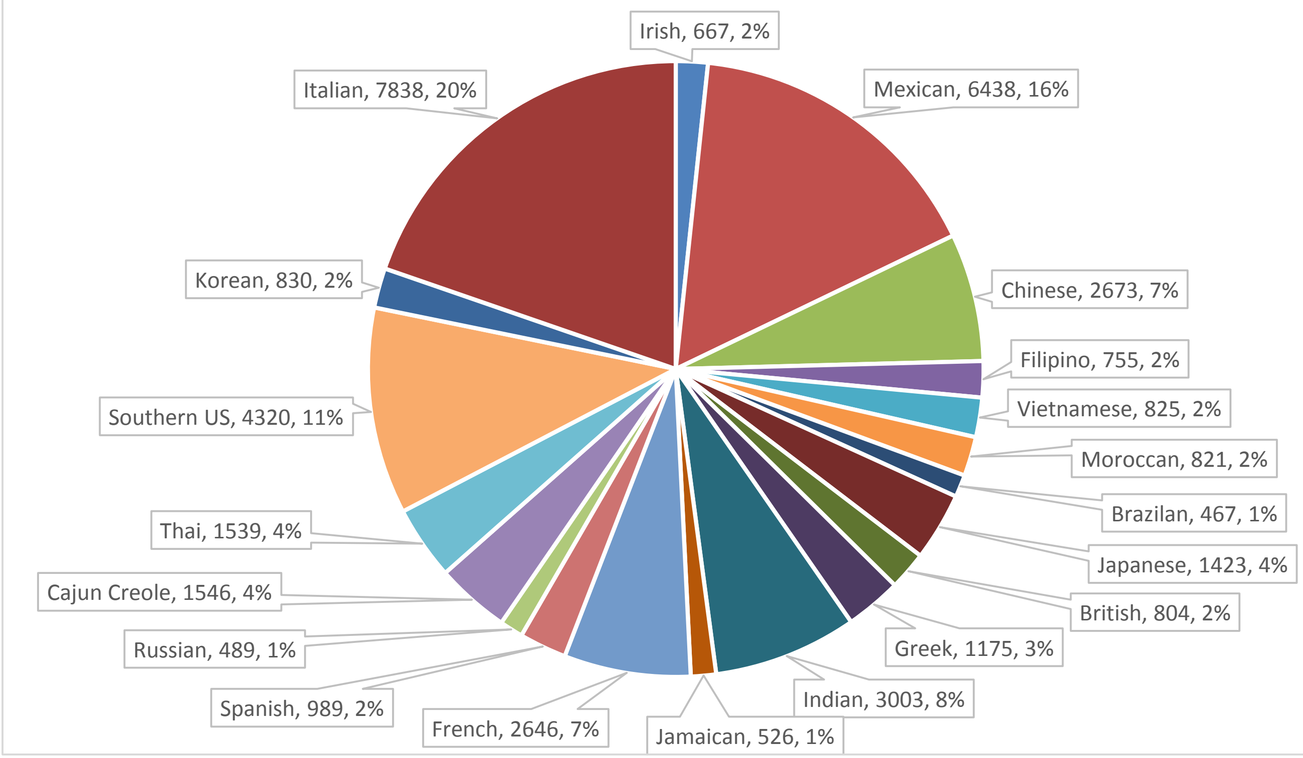
1. Introduction of Dataset

The training data from Yummly was a json file which 39k item entrees each which contained an item's id, its list of ingredients and the type of cuisine. While the test data was a json file with 10k item entrees with its item's id and the list of ingredients.

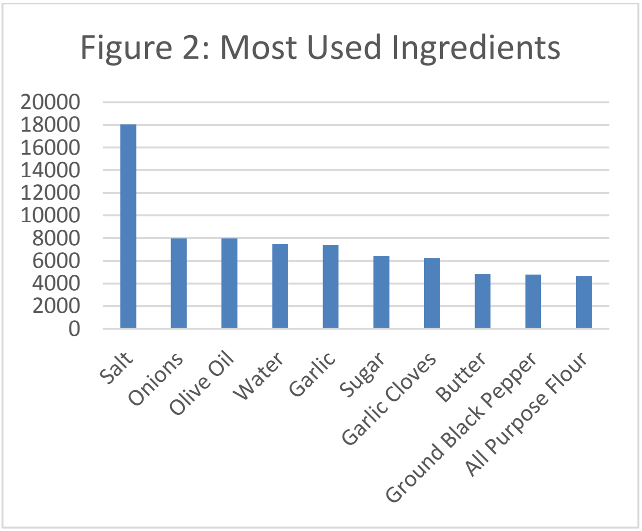
I first began to explore the data and see any simple trends. I first determined the spread of dishes across the twenty different cuisines. This is shown in figure 1. I noticed that certain cuisines dominated the data set, so it seemed important to make sure to give enough features so that these cuisines were distinguishable. I also noticed that many of the cuisine had less than five percent of the data sets entries so I would need to find interesting features that would make these cuisines stand out from the others.

Next I began to look at the ingredients to see if there was anything interesting. There were 6714

Figure 1: Percentage and number of Cuisine Dishes in Data



unique ingredients however 1759 of these ingredients were only used once and including these 3377 ingredients were used less than five times. Next I began to look at the ten most used ingredients which are shown in figure 2. I was not surprised that these ten ingredients were the most used, however it was surprising that the salt dominated by more than two times the second most used ingredient. Many of these ingredients were also used mostly for flavor which made me uncertain of their use in the model.



I continued to look at ingredients however this time with respect to cuisines. I also decided to ignore these top ten ingredients because I figured they would most likely dominate them as well. I figured that these ingredients would be key in identifying these keys. For example, here are the top five ingredients from Mexican cuisine, Italian cuisine and Chinese cuisine.

Mexican	
Ingredient	Count
Ground Cumin	1346
Chili Powder	1236
Jalapeno Chilies	1145
Sour Cream	1076
Avocado	1044

Italian	
Ingredient	Count
Grated Parmesan Cheese	1580
Extra-virgin Olive oil	1362
Pepper	965
Fresh Basil	787
Dry White Wine	658

Chinese	
Ingredient	Count
Soy Sauce	1363
Sesame Oil	914
Corn Starch	905
Green Onions	628
Vegetable Oil	602

Afterwards I also looked at the top pairs of ingredients for each cuisine again ignoring the ten most popular ingredients. Here are the top five ingredients from Vietnamese cuisine, Brazilian cuisine and Thai Cuisine.

Vietnamese	
Pair of Ingredients	Count
Fish Sauce, Shallots	97
Carrot, Fish Sauce	81
Beansprouts, Fish Sauce	80
Vegetable Oil, Fish Sauce	65
Fish Sauce, Rice Vinegar	64

Brazilian	
Pair of Ingredients	Count
Cachaca, Lime	43
Eggs, Milk	20

Pepper, Tomatoes	19
Coconut Milk, Tomatoes	19
Ice, Lime	17

Thai	
Pair of Ingredients	Count
Coconut Milk, Fish Sauce	177
Fish Sauce, Vegetable Oil	157
Fish Sauce, Fresh Lime Juice	136
Fish Sauce, Lime	130
Fish Sauce, Lime Juice	119

I also noticed that there were many ingredients with similar names but differed in one word. For example, fish sauce vs Asian fish sauce, garlic vs garlic cloves, ginger vs fresh ginger. Also another good example is in the thai pairings, there are three different variations of fish sauce and lime however, should these three pairings should be counted as one or separate. It would probably be important to determine if the model will do better if these ingredients are considered the same or if they are considered different.

2. The Predictive Task

2.1 The Model and Validity

The goal of the model is to predict cuisine based off the ingredients. This is a multiclass classification problem so the appropriate models learned in class would be support vector machines, logistic regression, and naïve Bayes. Some other models that may be usable are decision trees and random trees. A simple benchmark and also one the leaderboard is just simply predicting Italian all the time since it accounts for twenty percent of the recipes. However this should be relatively easy to beat. Another baseline would be to train the model using the ten most used ingredients and also with each cuisine's ten most used ingredients.

Since this data set is part of a Kaggle competition submitting the results from my model will help me assess the validity. However other than using a competition submission I randomly choose 5000 samples from the data, and made sure I had at least 100 samples from each cuisine as the validation set. To test my validity on the validation set, I used classification error: that is a mistake is counted for each time the model predicts the incorrect cuisine and then it is divided by the total amount of predictions.

2.2 Features

I plan on using a combination of single ingredients as well and pair of ingredients and if needed, triples of ingredients. Since these are the sole features I will just represent them as 1 if the item/pair/triple is in the ingredients and 0 otherwise.

To identify useable features I first started looking at easy to find features. Previously I found each cuisine's popular ingredients and pairs of ingredients as well. I then tested using these ingredients as features to determine if they were able to help the SVM figure out the cuisine.

However these ingredients were probably not good enough because there were 6714 possible ingredients (3337 ingredients ignoring ingredients used less than 5 times) so these features would probably be enough. I trained the SVM on these first simple ingredients. Then I looked at the predictions to determine which cuisines got mixed with others. For example, Thai and Vietnamese food got mixed up quite often so to help solve this issue I found both ingredients that were often in Thai food but not Vietnamese food and vice versa. This helped a lot with these small mix ups.

Also because Italian seemed to dominate the dataset, the SVM seemed to default to Italian if it was unable to determine the cuisine. So to reduce the times this happened, I aimed to find ingredients primarily used in Italian cuisine as well as ingredients that were never or rarely used in Italian cuisine. This would help the SVM differentiate between Italian and non-Italian cuisine reducing the amount of times Italian food was chosen.

Following a similar mindset, I also looked for ingredients that were primarily used in two or three cuisines, which would help the model narrow down the prediction. For example, Brazilian, Jamaican and Indian food seemed to use a lot of spices such as cumin, coriander and turmeric and others. So spices such as these were probably good indicator that recipe belonged to these cuisines.

Since ingredients were the only features available for the model, a main issue I was wondered about was overfitting. I tried to prevent this issue by choosing ingredients that appeared a good number of times, at least 50 for cuisines with large amount of entries and at least for 20 for others.

I also decided to look at ingredients that were mainly used by a certain cuisine. If a particular cuisine accounted for most of an ingredient's usage then that ingredient inclusion in the recipe is probably a good indicator that the recipe is of that certain cuisine.

3. The Model

I tried out a Gaussian Naïve Bayes, decision tree and logistic regressor along with a one vs one and one vs all SVM. The Gaussian Naïve Bayes strength is that it is able to perform well with very little data however its weakness is that it is not very complex and it also suffers from double counting. The decision tree weakness is that it easily suffers from overfitting. The logistic regressor provided in sklearn implements a one vs rest scheme. Logistic regression can work well if there is a linear decision boundary between the classes and has less variance. The one vs one SVM strength is that it compares each single class to each other so this may result in more accurate distinguishes between two hard to differentiate classes. However its main weakness is that because there are twenty cuisines which means twenty classes it has to train thirty nine classifiers which can take an immense amount of time.

I ended up combining a naïve ingredient checker along with the one vs all SVM. The naïve ingredient checker was implemented by counting the number of times an ingredient appeared throughout the whole data set and then for each cuisine. Then I checked to see if that any cuisine accounted for more than 85% of that ingredient's total usage and if that ingredient was used in at least 20% of the cuisine's different items. If so the ingredient was added to a dictionary. This was also done for the pairs of ingredients. Then while predicting the type of cuisine, if one these special ingredients or pair of ingredients appeared in the list of ingredients, the cuisine that accounted for 85% of the usage will be predicted and the SVM will not be used. My reasoning behind using this naïve checker is because in the training set these items were primarily used by a single cuisine and so it is very likely that this trend will continue on. Using this checker did help reduce training and test error so it did seem to help the model.

The issue with scalability was the amount of features being passed into the model. Because the only features that were available were ingredients and since there were several thousands of them, feature vectors for the model were very large, several hundred in fact and this made it so that training the model took a long time. This made an impact on the model I used, since I first tested logistic regression, naïve bayes, decision trees, one vs one SVM and one vs rest SVM on small amount of features, about fifty. Using these features the one vs rest SVM had the least validation error while the decision tree had the least training error. So I decided to use the one vs the rest SVM when training on these huge feature vectors. After I determined a good several hundred, I tested all these models again however I could not test the one vs one SVM because it took an immense amount of time to train, I waited thirty minutes and it was still not done and since the one vs rest SVM still had the lowest validation error I stuck with it.

4. Literature

The dataset came from Yummly as they hosted a public competition on Kaggle. There does not seem to be much work done on it. There actually is not much work that is being down on data sets similar to this, as it does not seem to be an important thing to figure out. The closest thing that I seemed to find online was finding certain cuisine's most common ingredients and pairs of ingredients however this was on a different data set. This was something I did as well to help determine the type of cuisine. I also could not find much research pertaining to determining a recipe's type of cuisine although there has been some research for food. An example was *Spices form the basis of food pairing in Indian cuisine* a paper by Anupam Jain, Rakhi N K, and Ganesh Bagler [1]. They found that the more two ingredients overlap in flavor, the less likely they are to appear in the same Indian dish. With this they provide a basis for recipe alteration as well as a way to design recipes.

Another interesting example of food related research was IBM Watson's cognitive cooking system [2]. The system aims to learn about food pairing theory, regional and cultural knowledge so that it can create new dishes.

However, this type of data does fall under the domain of classification. There are many models associated with classification beyond the ones I touched upon such as Random Forest, Neural Networks or Genetic Algorithms. In a paper *Text Classification Using Data Mining*[3] by S. M. Kamruzzaman, Farhana Haider, and Ahme Rydah Hassan they combine a Naïve Bayes classifier along with a Genetic Algorithms to classify a document's category. The resulting model ended up doing well using 50% of the training data opposed to models using a genetic algorithm or decision tree.

5. Results

My results seemed to perform pretty decently on Kaggle I had a score of .72814. The amount of data of each cuisine proved to have a large impact on the accuracy of my model. British cuisine only had 804 (2%) and Spanish cuisine had 989 (2%) recipes in the dataset. Most likely, due to the low amount of data that the model could be trained on for these cuisines, British cuisine had a 49% classification error and Spanish cuisine had a 48.7% classification error. On the other hand, cuisines that had very large had very low classification error because the model had a lot of data and unique ingredients to train upon. Italian which contributed about 20% of the data set, had only an 8.6% classification error.

However this was not only the most likely cause as unique ingredients seemed to play a large role in the classification accuracy. For example Mexican food, which was had the second most recipes at 6438(16%) had many unique ingredients such as many peppers, salsa, tortillas which made it so that the model only had a 6% classification error on Mexican cuisine. An even better example would be Indian cuisine which accounted for 8% of the data set but with its many interesting spices allowed the model to reach a 7.6% classification error on the cuisine. This shows how important it is to the model that each cuisine had unique ingredients and a good amount of data.

Similarly to the unique ingredients the task was difficult in that some cuisines were very similar to each other. Thai food and Vietnamese food both shared many common ingredients and this made it hard for the model to distinguish between the two. In the training data, out of the 220 mistakes made when the model predicted the recipe to be for Thai cuisine, 110 of those were actually Vietnamese recipes. And for Vietnamese recipes, 38 out of 82 mistakes made were for Thai food. This also probably contributed to British, Irish, French and Southern US cuisine having more errors as these four cuisines shared very similar ingredients.

What was surprising was that using ingredients that seemed to be used in most cuisines was actually helpful such as salt. Also another interesting thing was that those ingredients mainly used in one cuisine was actually helpful to include in the model, probably because the absence of these ingredients would indicate that its more improbable for it to be these cuisines. Following on that point, using the special ingredient and pair of ingredients dictionary in my model helped with the error quite a bit as well. I believe that the model succeeded decently well mostly because I aimed to not train on ingredients that seemed to be used very little. This way I was not over training on the training set and relying on these scarcely used ingredients.

A very difficult part for this assignment was the amount of different names for similar or the same ingredients. Some ingredients contained the word fresh, for example: fresh mozzarella cheese, fresh parsley, and fresh basil. However most recipes names for these ingredients were mozzarella cheese, parsley and basil. So I had to consider if these two examples were different or could they be considered the same. It was interesting that the model did better if some ingredients were grouped together, for example: Thai red curry paste and red curry paste however also some ingredients did better separately such as garlic and garlic cloves.

6. Conclusion

Overall the model seemed to do pretty decently, however it was not a close to be a top score on the Kaggle leaderboard. Perhaps to do better it would be necessary to try a different model from the different ones that I did try out.

7. References

[1] Anupam Jain, Rakhi N K, Ganesh Bagler. Spices form the basis of food pairing in Indian Cuisine. 12 Feb 2015. **arXiv:1502.03815**

[2] IBM Watson Cognitive Cooking Fact Sheet

http://www.research.ibm.com/software/IBMResearch/multimedia/Cognitive-Cooking-Fact-Sheet.pdf

[3] S.M. Kamruzzaman, Farhana Haiderr, Ahmed Ryadh Hasan. Text Classification Using Data Mining. **arXiv:1009.4987**