Machine Learning Engineer Nanodegree

Capstone Project

Man-Fong Cheng
January 6th, 2019

I. Definition

(approx. 1-2 pages)

Project Overview

This project is about Kaggle's What's Cooking[3], which is a interesting topic to use recipe ingredients to categorize the cuisine. I choose this topic because I love eating cuisine from different countries, so I would be extremely happy to work with data with recipe information.

Problem Statement

It is a supervised multi-class classification problem because the result should categorize the recipe into cuisine type, such as Chinese cuisine, Japanese cuisine or Italian cuisine, etc. The input data is list of ingredients for making cuisine, which is non-structured text format. However, the ingredients are not sentences and steps, they are just word, which make it easier to pre-process. For the solution algorithm, I choose XGBoost [1].

Metrics

I would use accuracy to evaluate the model.

II. Analysis

(approx. 2-4 pages)

Data Exploration

- There are train data-set and test data-set. And it is JSON format.
- Data includes **39774 recipes ids** with list of ingredients and type of cuisine, and I use train_test_split, split 20% as testing data. Because there's no label at test data from Kaggle, That's why I chose to split train data to test data.
- Here are an example of train data. e.g. {
  "**id**": 24717,
  "**cuisine**": "indian",
  "**ingredients**": [
  "tumeric",
  "vegetable stock",
  "tomatoes",
  "garam masala",

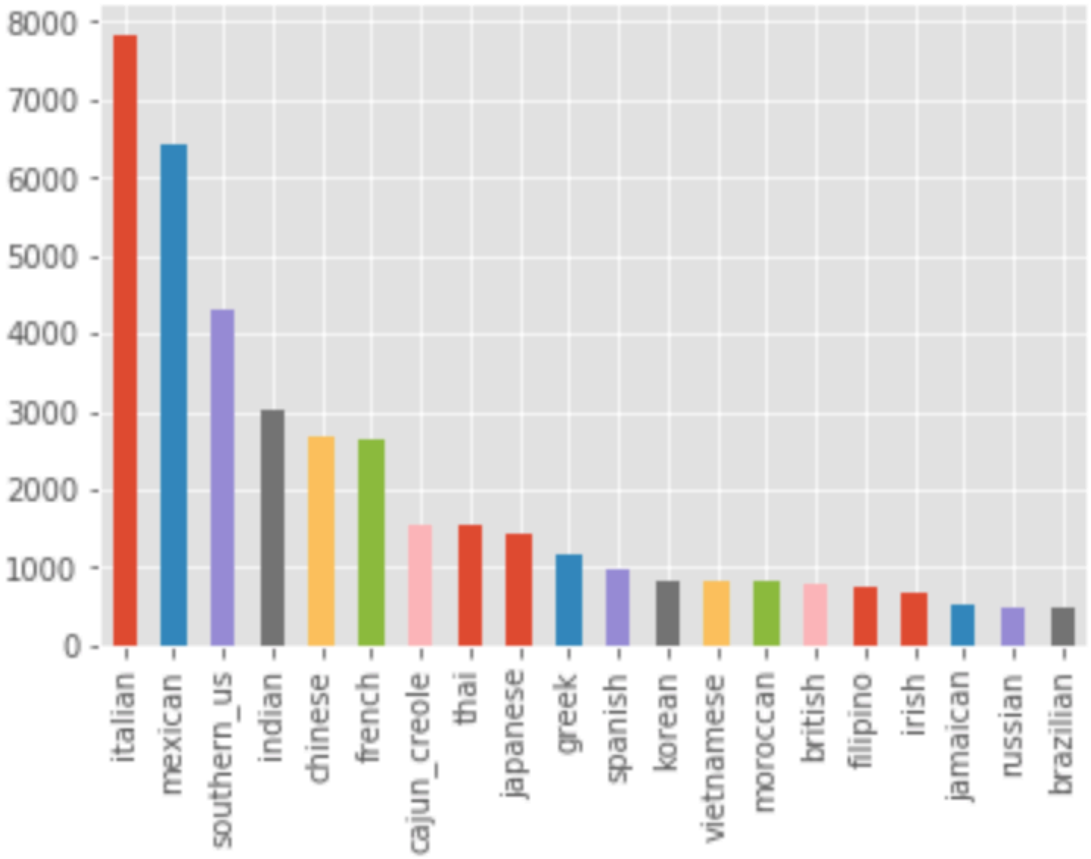
"naan",
"red lentils",
"red chili peppers",
"onions",
"spinach",
"sweet potatoes"
]
},

The input features include cuisine, id and ingredients. Let's discuss them below:

- id: It's discrete integer, and it is meaningless to our project.
- cuisine: There are **20 unique types of cuisines** in the data set, which means we are going to have 20 classes. There's no missing cuisine in this dataset. The highest frequncy cuisine in our dataset is italian, which includes 7838 cuisine. And we need to do one hot encoding to make cuisine appear in 0 and 1 in each type.
- Ingredients: There are ** 39774 cuisines**, and each of them have their own ingredients. There are some ingredients of cuisine are not unique. For 39774 cuisines we got 39674 unique ingredients. However, I don't think it matters, so cuisine use have same ingredients, but with different process step and ingredient's quatity make them different food. So I will not deal with the uniqueness of ingredients. There's no missing ingredients in this dataset, every cuisine has its own ingredients. However, as above mentioned in cuisine feature, ingredients also need to use one hot encoding to transfer data into 0 and 1 for further process.

Exploratory Visualization

As below, we can see each cuisine count from dataset. The highest frequncy cuisine in our dataset is italian, which includes 7838 cuisine. And we need to do one hot encoding to make cuisine appear in 0 and 1 in each type.



Algorithms and Techniques

I am going to use XGBoost as my algorithm, I will input ingredients as data, and cuisine as target. XGBoost has hyperparameterm and I will use grid search to pick a better one. And all the input data need to transfer by one hot encoding.

Benchmark

SVM is my benchmark, I use my preprocess data and run SVM, with parameter C: 0.5, and tolerance :1e-3. And I got 0.9611 accuracy from test data, which is very high.

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

In data preprocessing part, I first check top 10 ingredients in each cuisine, and we can see many similar ingredients, based on my google result, I list this can be seens as the same, and I will assign it as same while data preprocessing scallions = Green Onions [1] black pepper = pepper [2] eggs = large eggs [4] extra-virgin olive oil = olive oil [5] garlic = garlic cloves

And then stemming (potatoes and potato should be the same ingredients), also I do capitalize. And one hot encoding.

Implementation

- Data exploration: Explore the whole data, and make sure there's no missing data.
- Data pre-processing:
  - combine into same ingredient (As above mention)
  - stemming (porterStemmer)
  - capitalize
  - One Hot encoding (CountVectorizer, LabelEncoder, label_binarize)
- Training:XGBoost, and using grid search (learning_rate: 0.01, 0.1, max_depth: 3, 6)
- Testing: I am going to test my model from previous training result.

There's no any complications with the original metrics or techniques that required changing prior to acquiring a solution.

Refinement

These no refinement in this report, initial solution and final solution is the same.

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

The result is 0.961491341975562 accuracy, I thinkt the final model is reasonable and aligning with solution expectations. It is tested by spliting data which is not same as training data, so I think its result can be trusted.

Justification

- The final results(0.961491341975562) is stronger than the benchmark result(SVM,0.9611). And I think the final solution is significant enough to have solved the problem for categorizing ingredients into cuisine type.

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

Test result is: 0.961491341975562.

Reflection

In this project, I first explore the whole data, and make sure there's no missing data. Then in Data pre-processing part, I combine same ingredients with different wording into same ingredient with same wording, which need some domain knowledge, and I don't think many programmer consider about this part. Also, I do stemming, capitalize, One Hot encoding. And I trained the data by XGBoost, and for its hyperparater, using grid search for better parameter pair. Finally,test my model after training it. The most interesting part is to combine same ingredients, because I google top 10 ingredient of each cuisine, and read many chief blog to gain some domain knowlege of food. And I think the final model and solution fit your expectations for the problem, and it can be used in a general setting to solve predicting cuisine from its ingredients.

Improvement

I noticed that there's many ingredients include brand name, such as KRAFT cheese, and I think it can improved by trimming and ignoring brand name while preprocessing. And if I used my final solution as the new benchmark, I think there's some even better solution exists with better data preprocessing.

[1] https://www.thekitchn.com/whats-the-difference-between-spring-onions-scallions-and-green-onions-word-of-mouth-217111 [2] https://www.chefsteps.com/ingredients/black-pepper [3] https://www.kaggle.com/c/whats-cooking [4] https://www.thespruceeats.com/egg-size-conversions-1328750 [5] https://www.thekitchn.com/whats-the-difference-between-olive-oil-and-extra-virgin-olive-oil-word-of-mouth-218767