Machine Learning Engineer Nanodegree

Capstone Proposal

Man-Fong Cheng
December 25th, 2018

Proposal

Domain Background

I will take Kaggle's What's Cooking[3] as my topic, 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]

Datasets and Inputs

- There are train data-set and test data-set. And it is JSON format.
- Train data includes **39774 recipes ids** with list of ingredients and type of cuisine
- Test data includes of **9944 recipes ids** with list of ingredients
- There are **20 types of cuisines** in the data set, which means we are going to have 20 classes.

- 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”
  ]
  },

Solution Statement

The solution would predict cuisine by its ingredients, and the result would be like “Italian”, “Indian”, and “Chinese”, etc. For pre-process step, I will first pre process my input data by capitalize(Onion and onion should be the same), stemming [2] (potatoes and potato should be the same ingredients). Then, I would use a bag of word feature vector for each possible ingredient where number represent the presence times of that ingredient and 0 the mean absence in the current recipe. After pre-processing, I would use XGBoost algorithm, because it is winner algorithm in most Machine Learning Competition[1]. For XGBoost model’s hyperparameters, I would use GridSearch to tuning parameters for better result.

Benchmark Model

I would use SVM as benchmark model, most of the kernels in Kaggle use SVM, and they got good result, so I set this as my benchmark.

Evaluation Metrics

I would use accuracy to evaluate the model.

Project Design

- Data exploration: I would explore the whole data, and make sure how many category do I need.
- Data pre-processing: I might need pre process the data, because it may have some capitalize issue (e.g. Milk and milk), and it may have some similar words, which can combine into same word (e.g. Whole Milk and Milk can all seems as Milk).
- Training: I am going to train my model by XGBoost, and using grid search.
- Testing: I am going to test my model from previous training result.
- Evaluate: Finally, I would evaluate result by its accuracy, and compare with using bench mark model SVM.

---

[1] https://medium.com/syncedreview/tree-boosting-with-xgboost-why-does-xgboost-win-every-machine-learning-competition-ca8034c0b283
[2] http://sklearn.lzjqsdd.com/modules/feature_extraction.html
[3] https://www.kaggle.com/c/whats-cooking