Machine Learning Engineer Nanodegree

Capstone Proposal : What's cooking

Mandeep Kaur
April 08, 2017

Proposal

Domain Background

Cuisine is the style of cooking. Every culture in this world has its own cuisine. Every cuisine has its own unique taste. Everyone can agree that this uniqueness comes from the choice of ingredients. Nobody can deny that few flavours are strongly related to some certain parts of world. Many times people, including the food critics, try to define a cuisine entirely on the bases of its ingredients.

This project takes inspiration from a competition on Kaggle. I like trying new dishes, new flavours. There are many implementations and different ideas for this problem. For example 10 Most used ingredients [1] and data exploration using word2vec algorithm[2]

Problem Statement

- The goal is to create a model that can, given the ingredients of a recipe, find the cuisine of that recipe it belongs to. This is a supervised learning classification problem. Based on list of ingredients we are going to predict the cuisine of the recipe.
- Find diffusion of cuisines. How similar and different they are based on the ingredient used. This is an unsupervised learning problem. We are going to find relationships and groups between different cuisine using ingredients.

Datasets and Inputs

For this project the dataset is publically available on Kaggle's competition "What's cooking?"[3] The dataset includes the recipe id, cuisine and list of ingredients.

An example is:


```
{
 "id": 24717,
 "cuisine": "indian",
 "ingredients": [
     "tumeric",
     "vegetable stock",
```

```
      "tomatoes",
      "garam masala",
      "naan",
      "red lentils",
      "red chili peppers",
      "onions",
      "spinach",
      "sweet potatoes" ]
 }
```

Total size of data is 1.76MB. This dataset contains a total of 6714 unique ingredients and 39774 sets of recipes' ingredient lists with their cuisines.

This dataset has 20 different cuisines.

```
['british',
 'french',
 'moroccan',
 'russian',
 'filipino',
 'japanese',
 'mexican',
 'brazilian',
 'indian',
 'chinese',
 'italian',
 'vietnamese',
 'spanish',
 'irish',
 'greek',
 'korean',
 'southern_us',
 'cajun_creole',
 'thai',
 'jamaican']
```

These cuisines include Asian, African, European and American.

Solution Statement

First of all I am going to perform some transforms on my data, because the list of ingredients is long. For this I am going to TfIdf representation.

Finding cuisine : For this part I am going to train my data on different algorithms : Random Forest, Stochastic Gradient Descent and Logistic Regression. Then I will compare these algorithms based on different metrics like speed, accuracy etc. After the comparison, I will choose one that is best and use that to test my model.

Cuisine diffusion : For this I will first get a count of different ingredients for all the recipes. Then I will apply PCA transformation on this data to reduce the dimensionality. After that I will apply K-Means unsupervised algorithm on this data to find clusters of cuisines

Measurable: with the help of the below defined evaluation metric we can easily evaluate the efficiency of our model. Replicable: This problem is easily reproducible. Anyone can define

a list of ingredients of a recipe and replicate it. Quantifiable: The problem can easily be expressed in terms of mathematics. We can encode the list of ingredients because this a categorical data and then run it through a classification and unsupervised algorithm.

Benchmark Model

For the benchmark model my choice is to use decision tree with the following parameters to find cuisine:
criterion : gini,
min_samples_split:2
min_samples_leaf=1
(This is the default configuration of the decision tree classifier in sklearn)

Decision tree classifier is a classic algorithm for this type of problem, so this is the best choice to create a benchmark model. The accuracy of predictions made on test set with this model will be used as a result to compare with our model.

For cuisine diffusion, my benchmark model will be predefine set of general rules, According to which all the cuisines will be clustered based on the region. The more geographically and continentally close the region the more close the clusters would be and vice versa. For example all the East Asian cuisines like Korean, Japanese will be closer to each other but far from Western ones like Mexicans or Spanish etc.

As the metrics for these benchmarks I will be using the below explained metrics as these are the same metrics that would be used for model evaluation.

Evaluation Metrics

The most suitable evaluation metric for Finding Cuisine problem is logloss. This is a classification problem and the efficiency of the solution can be evaluated on the number of cuisines it can identify accurately. And because this is a small dataset using this metric will be more efficient.

This function quantifies the accuracy of the classifier by penalizing false classifications. The classifier assigns the probability to each class rather than simply yielding the most likely class.

So we can simply define LogLoss in mathematical terms as:
$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(p_{i,j})$$

For the cuisine diffusion part the total number of clusters will be 20 as are our cuisines and the efficiency will be evaluated by the rules we defined in our benchmark model.

Project Design

The workflow for this solution using python is :

Data Transformation :

1. Download the datasets
2. Split the columns into two sets: X for ingredient list and Y for respective cuisines
3. Now encode the ingredients list into tfidf matrix.

Finding cuisine :

4. Divide the data into train and test.
5. Train the benchmark model and record the test results
6. Train this data on all the three algorithms.
7. Plot graphs for all algorithm performances.
8. Evaluate the best model.
9. Use this model to find the accuracy on test data.
10. Compare the results with benchmark model.

Cuisine diffusion:

11. Use the TfIdf matrix to find the count of ingredients for each cuisine.
12. Apply PCA
13. Compare the explained variance of variance dimensions
14. Choose the number of principal dimensions required to reduce dimensionality.
15. Generate reduced data using these dimensions
16. Apply k-means on this reduced data
17. Plot the resulting data model for visualization.
18. Check whether the result is in accordance with our predefined set of rules.

---

citations:

[1] https://www.kaggle.com/manuelatadvice/whats-cooking/noname

[2] https://www.kaggle.com/ccorbi/whats-cooking/word2vec-with-ingredients

[3] https://www.kaggle.com/c/whats-cooking