CS 5293, Spring 2020 Project 3

Due 5/8 8am

The Analyzer

Introduction

In this project, imagine you are a data scientist for a hotel chain. This hotel chain is well-known for its variety of foods. Experts have claimed that tweaking the menu can improve success by 15%. As the chief data scientist, you are given the task of improving menu profits. To do this, you are asked to first help the hotel staff with menu planning and preparation. The goal the project is to assist the Executive Chef in understanding the large menu set better by providing a cuisine predictor.

You are given the master list of all possible dishes, their ingredients, an identifier, and the cuisine for thousands of different dishes. You realize that if you cluster foods by their ingredients you can help the restaurant change foods but keep the ingredients constant. You present a display of clustered ingredients and train a classifier to predict the cuisine type of a new food. To complete this project, you will be asked to create a small interface that asks a user to supply a list of ingredients and returns the predicted cuisine type.

The data sets for this project is provided by Yummly.com — it was used as part of a Kaggle contest (some people in the course have already participated in this contest.) Below are the requirements for this project. This project involves a large amount of data. To help the development of your code, test on a subset of the data first. Make sure to start today!! You are free to discuss approaches using the help board, but solutions must be your own. I also encourage reaching out and asking questions early as many students will have similar concerns.

Here are two prominent papers that analyzed food data. We used data from this document in previous classes.

(1) Flavor Network and the Principles of Food Pairing http://www.nature.com/articles/srep00196; (2) Inferring Cuisine - Drug Interactions Using the Linked Data Approach http://www.nature.com/articles/srep09346

Project Specification

The goal of the project is to create an application that take a list of ingredients from a user and attempts to predict the type of cuisine and similar meals. Consider a chef who has a list of ingredients and would like to change the current meal without changing the ingredients. The steps to develop the application should proceed as follows.

1. Pre-train or index all necessary classifiers using the existing datasets.
2. Ask the user to input all the ingredients that they are interested in.
3. Use the model to predict the type of cuisine and tell the user.
4. Find the top N closest foods (you can define N). Return the IDs of those dishes to the user. If a dataset does not have IDs associated with them you may add them arbitrarily.

Click here to access the yummly.json data set. The yummly data set contains sets of foods and a cuisine. You can use the cuisine type as a label.


```
{
  "id": 10276,
  "cuisine": "mexican",
  "ingredients": [
    "chili powder",
    "crushed red pepper flakes",
    "garlic powder",
    "sea salt",
    "ground cumin",
    "onion powder",
    "dried oregano",
    "ground black pepper",
    "paprika"
  ]
}
```


Above is an example entry from the yumly.json data set.

But there are many ways to complete the project. I encourage you to be creative and take advantage of the scikit-learn and Spacy libraries to design your system. Should consider using word2vec or BERT pretrained embeddings.

The application may either be a command line application or a website, but the logic should be executed using Python. *You may also get the input from the user by changing variables in your code (specifically applicable if you are using Jupyter notebooks that*

cannot take command-line input). Below is an example run of a command line application.

```
$ python3 project2.py --ingredient paprika --ingredient banana --
Cuisine: America (.91)
Closest 5 recipes: 10232 (.34), 10422 (.15), 45 (.13), 7372 (.04)
```

In this example, 3 ingredients were passed in to the program. The program then ran and matched the closest cuisine type, followed by a distance to this cuisine. The next line has the list of the top 5 closest meals with their distance in parenthesis. This is a synthetic example, feel free to use creative solutions.

Create a README that allows us to understand all assumptions and design decision you encountered when developing the system. It should also include directions on how to recreate and run your applications. Use the python project structure that we used in previous assignments. Include tests to validate your code. You will submit the README along with all source code as a zip or tar.gz file.

Download yummly https://www.dropbox.com/s/f0tduqyvgfuin3l/yummly.json

Project Rubric

Code: 60 points

- 10 pts: Parse dataset
- 10 pts: Conver the text to features
- 10 pts: Train or prepare classifiers
- 5 pts: Get input
- 10 pts: Prediction
- 10 pts: Select closest N recipes
- 5 pts: Give output

Readme: 40 points

- 10 pts: How did you turn your text into features and why?
- 10 pts: What classifiers/clustering methods did you choose and why?
- 10 pts: What N did you choose and why?
- 5 pts: Describe functions/code
- 5 pts: Describe tests

Submission

To submit your project, please do the following:

1. Create a Github and add collaborators `cegme`, `kbanweer`, and `mghirsch42` by going to `Settings` > `Collaborators`.
2. Submit all files to Gradescope. This can be accessed through Canvas using the Gradescope tab.

Addendum

4/29: Project Rubric updated

4/29: Submission instructions updated

4/29: User input may be from variables in the code, not necessarily command line input.
