AI506: Data Mining and Search (Spring 2022)

Term Project: Predicting Cuisine from Ingredients

Release: Mar 18, 2022
Progress Report: May 6, 2022, 11:59 pm
Final Report: June 3, 2022, 11:59 pm
Presentation: June 6, 2022, 11:59 pm

The ultimate goal of this project is to practice data mining research by classifying recipes and inferring missing ingredients using a dataset of recipes. In this project, you will design, implement, and evaluate your approach for finding the labels of recipes and inferring missing ingredients in recipes. Also, you will (a) write a progress report, (b) write a final report, and (c) present your approach. While details of the following steps will be announced later, tentative schedules are as follows:

- Progress Report - May 6, 2022, 11:59 pm
- Final Report - June 3, 2022, 11:59 pm
- Presentation - June 6, 2022, 11:59 pm

This is a team project, and each team should consist of two or three members. You can find your teammates by all means (e.g., Classum), and one progress report should be submitted per team.

Your submission will be evaluated based on

- Presentation (final report & oral presentation) - 40%,
- Novelty of your proposed approach - 20%,
- Validity of your proposed approach - 20%,
- **Accuracy - 20%**.

Note that accuracy is not our only concern. Instead of spending all your time optimizing the accuracy, we recommend spending more time on developing novel and valid approaches and making your presentation clear and complete.

1 Problem: Predicting Cuisine from Ingredients

1.1 Provided Data

The provided dataset contains the ingredients and label (i.e., cuisine) of recipes. They contain 6,714 ingredients and 35,319 recipes in total. All ingredients are represented as integer IDs, and the labels of recipes are represented as strings (e.g. japanese and italian). Fig. 1 illustrates an example of a recipe (set of ingredients) and its label (Ingredients: 281, 937, 5486, Label: Italian).

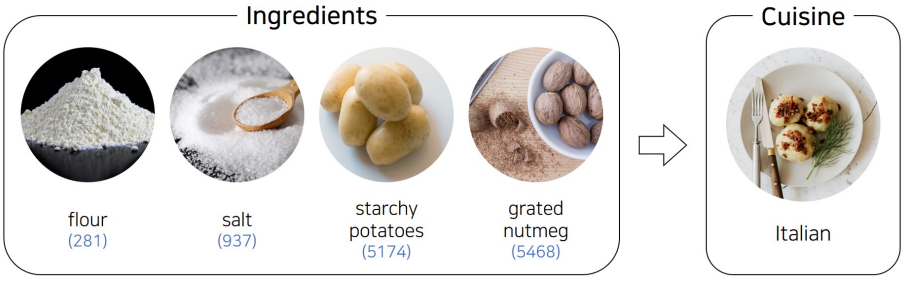

Figure 1: An example of a recipe (i.e., set of ingredients) and its label (i.e., cuisine). The number below each ingredient is the corresponding integer ID.

There are 20 types of labels (i.e., cuisines) for recipes in this dataset. All recipes include at least two ingredients. The goal of this project is to infer missing ingredients and classify recipes. In Fig. 2, we illustrate the completion and classification tasks. Specifically, the goal of the completion task is to infer a missing ingredient in a given incomplete recipe. The goal of the classification task is to classify a given complete recipe.

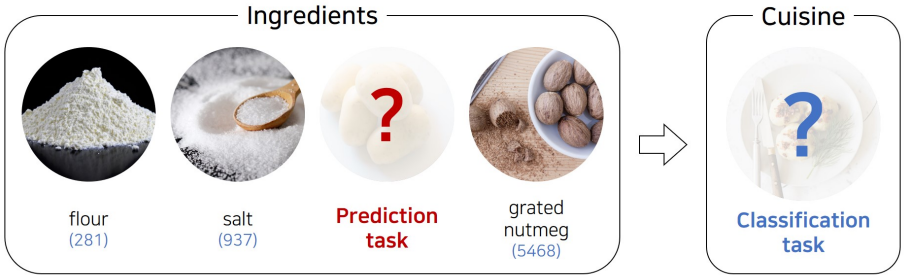

Figure 2: Example of completion and classification task.

We provide a dataset that consists of training, validation, and test sets. The validation and test sets are provided for each task. Below, we describe each set.

Training set

The training set (`train.csv`) contains 23,547 recipes, and each recipe has its own label (i.e., cuisine). Each line of `train.csv` consists of ingredient IDs separated by a comma and the last word is the label of the recipe.

Validation sets

Four files are provided for validation, and each of them contains information about 7,848 recipes. Each line of `validation_classification_question.csv` consists of the ingredients of a recipe without its label. The label of the recipe is provided in `validation_cla ssification_answer.csv`. Each line in `validation_completion_question.csv` consists of all the ingredients except for one of a recipe. The missing ingredient is provided in `validation_ completion_answer.csv`.

Test sets

The test sets (`test_classification_question.csv` and `test_completion_question .csv`) contain 3,924 recipes. Labels are missing in `test_classifcation_question.csv`, and one ingredient is missing in each recipe in `test_completion_answer.csv`.

	# recipes	Does the label of recipe exist?
Training dataset (train.csv)	23,547	yes
Validation dataset (validation_classification_qeustion.csv)	7,848	no
Validation dataset (validation_classification_answer.csv)	7,848	yes
Test dataset (test_classification_question.csv)	3,924	no

Table 1: Statisics of the datasets for the classification task

	# recipes	Is one ingredient missing?
Training dataset (train.csv)	23,547	no
Validation dataset (validation_completion_qeustion.csv)	7,848	yes
Validation dataset (validation_completion_answer.csv)	7,848	no
Test dataset (test_completion_question.csv)	3,924	yes

Table 2: Statistics of the datasets for the completion task

1.2 Evaluation

The goal of this project is to classify the recipes in `test_classification_question.csv` and infer the missing ingredient in each recipe in `test_completion_question.csv`. You should save the results in `test_classification_answer.csv` and `test_completion_answer.csv` for the classification and completion tasks, respectively. There must be one label per line in `test_classification_answer.csv` and one ingredient per line in `test_completion_answer.csv`. The i-th row of `test_classification_answer.csv` is the label for the recipe in the i-th row of `test_classification_question.csv`. The i-th row of `test_completion_answer.csv` is the missing ingredient of the recipe in the i-th row of `test_completion_question.csv`.

Using these files, we will evaluate the performance of your approach using F1 scores and accuracy. F1 scores will be measured on `test_classification_answer.csv`, and accuracy will be measured on `test_completion_answer.csv`. We will use two types of F1 scores: macro F1 and micro F1.

$$\textit{macro F1 score} = \frac{\sum_{s \in S} TP_s / \{TP_s + \frac{1}{2}(FP_s + FN_s)\}}{|S|},$$

$$\textit{micro F1 score} = \frac{\sum_{s \in S} TP_s}{\sum_{s \in S} TP_s + \frac{1}{2}\sum_{s \in S}(FP_s + FN_s)},$$

$$\textit{accuracy} = \frac{\textit{\# correct completions}}{\textit{\# completions}},$$

where TP_s, FP_s, and FN_s mean the numbers of true positives, false positives, and false negatives for the label $s \in S$, respectively. In addition, S is the set of labels.

Note that we may run the submitted code on another query dataset if your answer is suspiciously similar to any other group's answer.

1.3 Notes

- You may encounter some subtleties when it comes to implementation, please come up with your design and/or contact Taehyung Kwon (taehyung.kwon at kaist.ac.kr) and Hyeonsoo Jo (hsjo at kaist.ac.kr) for discussion. Any idea can be taken into consideration when grading if it is written in the *readme* file.
- Unlike the other assignments, you can use any programming language and any external library.

2 How to submit your project

2.1 Progress Report

Submit your progress report written on the attached template to KLMS by May 6, 2022, 11:59 pm. The file should be named report-[your student ids].pdf (e.g., report-20189000_20199000_20209000.pdf). Details will be announced soon.

2.2 Final Report Submission

1. Submit project-[your student ids].tar.gz (e.g., project-20189000_20199000_20209000.tar.gz) to KLMS. Your submission should contain the following files:

   - **final_report.pdf**: a final report written on the attached template with LaTeX.
   - **test_answer.tar.gz**: this file should contain the `test_classification_answer.csv` and `test_completion_answer.csv`.
   - **readme.txt**: this file should contain the names of individuals from whom you received help and the natures of help that you received. This includes help from friends, classmates, lab TAs, course staff members, etc. In this file, you are also welcome to write any comment that can help us grade your assignment better, your evaluation of this assignment, and your ideas. This file also should describe how to run your code.
   - **code.tar.gz**: your implementation

2. Make sure that no other files are included in the tar.gz file.

2.3 Video Presentation Submission

1. Submit project-[your student ids].tar.gz (e.g., project-20189000_20199000_20209000.tar.gz) to KLMS. Your submission should contain the following files:

   - **slides.pdf**: slides used for the final presentation.
   - **video.mp4**: recorded video presentation that does not exceed 5 minutes.

2. Make sure that no other files are included in the tar.gz file.