CSE 158/258, Fall 2021: Midterm

Instructions

Midterm is due at midnight, PST, on Wednesday November 3. Submissions should be made via gradescope.

The midterm is worth 26 marks.

You can base your solution on the stub code provided here:
`https://cseweb.ucsd.edu/classes/fa21/cse258-b/files/Midterm_stub.ipynb`
`https://cseweb.ucsd.edu/classes/fa21/cse258-b/files/Midterm_stub.html`

Submissions should take the form of a single pdf; we suggest extending the stub code and printing to pdf, though will accept other formats. **Please carefully label which pages correspond to which questions when turning in your work.**

Section 1: Regression

Note: none of these experiments should require more than a few seconds to run on modest hardware. Please only use a smaller fraction if absolutely stuck;[1] we may only give partial credit if we cannot verify the correctness of your solution.

1. Implement feature representation strategies to predict the cook time ('minutes') based on:

   (a) The length of the recipe (number of characters in 'steps'), and the number of ingredients

   (b) The year, and the month the recipe was entered (one-hot encoded) ('submitted')

   (c) A 50-dimensional binary vector indicating the presence or absence of the 50 most popular ingredients (across all recipes)

   Implement functions to extract each of the three feature vectors above. Split data using 75%/12.5%/12.5% training/validation/test splits,[2] and print the feature corresponding to the first training sample for each strategy. Report the MSE (on the test set) of each model (trained on the training set). For this question, it is fine not to use any regularizer in your model (6 marks).

2. (a-c) An *ablation* experiment measures the *relative* importance of a (set of) features in a model by comparing (1) a model with *all* features; to (2) a model with the features under consideration *excluded*. For example, to measure the importance of our 50-dimensional binary vector (1c), we would train a model with *all* features from Question 1, and compare it to one with only those from Parts 1a and 1b. Conduct ablation experiments to measure the importance of each of the three sets of features from Question 1a-1c. Again report performance (MSE) on the test set for each of the three ablations, and for a model which includes all features simultaneously. Briefly reason about which features are the most important (predictive) for this task (3 marks).

3. **(CSE158 only)** Implement a *validation pipeline*, using the 'Ridge' function and including values of $\lambda \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$.[3] Report the training and validation performance for each value of λ, and the test performance for the model which performs best on the validation set (2 marks).

4. **(CSE258 only)** (Design thinking) Suppose *cooking time* data has a large number of recipes with low cook times (e.g. most have cooking times of 15-30 minutes), but that there is a long-tail of recipes with cooking times in the 8+ hour range. Comment on the appropriateness of predicting cooking time and measuring accuracy in terms of the MSE in such a situation. Suggest some possible strategies (at least two, but more is fine) for designing a better predictor in such a dataset, either by transforming the output variable, modifying the dataset, changing the loss function, modifying the task, or otherwise (2 marks).

Section 2: Classification

5. Build a classifier to predict whether a recipe contains 'butter' from the other items in the ingredient list. You can reuse your feature representation from Question 1c.[4] Train a logistic regression classifier ('sklearn.linear_model.LogisticRegression') using the 'class_weight=balanced' option, with regularization parameter $C = 1$.[5] Report the Balanced Error Rate of the classifier on the test set (using the same train/validation/test sets as in Question 1) (2 marks).

6. **(CSE258 only)** Build a *regularization pipeline* for your model. This model has two hyperparameters: (a) the regularization constant C; and (b) the number of ingredients N used in your feature representation. Using a validation set, select optimal values for the two constants and report performance on your test set. You may use any choices of values for C and N, as long as you have at least three values for each, and your chosen solution should outperform that of Question 5.[6] Report training and validation performance for each experiment you perform, and test performance for your selected model (2 marks).

[1] If doing so, use the first 10% of the data

[2] The stub code already includes data splits; the dataset is already shuffled, and doesn't need to be shuffled further.

[3] Use all three features from Question 1 together.

[4] Be careful not to include the target category in your feature representation!

[5] Both options are arguments when creating the model.

[6] The simplest strategy is *grid search*, in which you iterate over all combinations of values for C and N, though you could select a different strategy if you wish. Describe the strategy you choose.

7. **(CSE158 only)** (Design thinking) Suppose our goal is to retrieve recipes that satisfy dietary restrictions (e.g. dairy- or gluten-free, vegan, etc.) by training classifiers such as those above. What would be appropriate evaluation criteria to use (e.g. accuracy, BER, precision, recall@k, etc.)? Consider possible use-cases, e.g. a vegan looking for suitable recipes for themselves versus a user cooking for a friend without understanding the restrictions themselves. You're welcome to describe a few criteria for different scenarios. Justify your choices (2 marks).

Section 3: Recommendation

Note: These questions contain no machine learning-based questions; solutions may be based on the entire dataset.

8. Here, we'll build a recommender that suggests one recipe based on its similarity to another. Build a recommender that recommends a similar recipe based on the *Jaccard similarity* of their ingredient lists ('ingredients'). What are the five most similar recipes to the first recipe in the dataset ('06432987')? Report their titles (or IDs) and Jaccard similarity values. In case of ties return the alphabetically first item (as in the homework) (2 marks).

9. Likewise, the Jaccard similarity could be used to measure which *ingredients* are the most similar, based on the overlap in terms of what recipes they appear in. Based on such a measure, which five ingredients are most similar (highest Jaccard similarity) to *butter*? Report ingredients and Jaccard similarities (again resolving ties alphabetically) (2 marks).

10. (Design and Implementation) Suppose a user has a *set* of ingredients in mind for a recipe, e.g. {cinnamon cherries, butterscotch, vodka}. How would you design a system to select recipes that most capture the 'essence' of those ingredients? While you could trivially just use the Jaccard similarity (or similar) to retrieve the recipe with the best ingredient overlap, this may be too inflexible (e.g. few recipes may match the ingredients exactly, but some may contain *similar* ingredients).[7]

    Design an approach that will retrieve recipes relevant to a given query (in the form of a set, as above). Describe your approach in a few sentences, implement it, and show retrieved recipes for a few representative examples. While not required, you may use fields other than the ingredient list if desired.

    Feel free to make appropriate design choices and to think creatively. I.e., build what *you* think would be a useful system for recipe discovery (5 marks).

[7] By all means, you should implement something trivial if you get stuck trying to implement a more complex solution, though may not receive full marks.