CSE 158/258, MGTA 461, DSC 256, Fall 2024: Homework 2

Instructions

Please submit your solution **by Monday Oct 28.** Submissions should be made on **gradescope**. Please complete homework **individually**.

You should submit two files:

`answers_hw2.txt` should contain a python dictionary containing your answers to each question. Its format should be like the following:

```
{ "Q1": 1.5, "Q2": [3,5,17,8], "Q2": "b", (etc.) }
```

The provided code stub demonstrates how to prepare your answers and includes an answer template for each question.

`homework2.py` A python file containing working code for your solutions. The autograder *will not execute your code*; this file is required so that we can assign partial grades in the event of incorrect solutions, check for plagiarism, etc. Your solution should **clearly document which sections correspond to each question and answer**. We may occasionally run code to confirm that your outputs match submitted answers, so **please ensure that your code generates the submitted answers.**

You will need the following files:

Homework 2 stub : `https://cseweb.ucsd.edu/classes/fa24/cse258-b/stubs/`

Polish Bankruptcy data : `https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data`

GoodReads Young Adult Reviews : `https://cseweb.ucsd.edu/classes/fa24/cse258-b/data/young_adult_10000.json.gz`

Further code examples for regression and classification are available on the class and textbook webpages. Executing the code requires a working install of Python 2.7 or Python 3 with the scipy packages installed.

Each question is worth one mark unless otherwise specified.

Tasks — Model Pipelines and Diagnostics:

In the first homework, we began to explore a coulpe of issues with the classifiers we built. Namely (1) the data were not shuffled, and (2) the labels were highly imbalanced. Both of these made it difficult to effectively build an accurate classifier. Here we'll try and correct for those issues using the *Bankruptcy* dataset.

1. Download and parse the bankruptcy data. We'll use the `5year.arff` file. Code to read the data is available in the stub. Train a logistic regressor (e.g. `sklearn.linear_model.LogisticRegression`) with regularization coefficient $C = 1.0$. Report the accuracy and Balanced Error Rate (BER) of your classifier.
2. Retrain the above model using the `class_weight='balanced'` option. Report the accuracy and BER of your new classifier.
3. Shuffle the data, and split it into training, validation, and test splits, with a 50/25/25% ratio. **Use the code in the stub provided to ensure that your random split is the same as the reference solution.** Using the `class_weight='balanced'` option, and training on the training set, report the training/validation/test *BER*.
4. Implement a complete regularization pipeline with the above classifier. Consider values of C in the range $\{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$. Report the *validation* BER for each value of C.
5. Based on these values, which classifier would you select (in terms of generalization performance)? Report the best value of C and its performance (BER) on the test set.

Tasks — Recommendation:

For this question we'll use the Goodreads book review data. The first 90% of the data should be used for training and the remaining 10% for evaluation (the stub shows how to split the data).

6. Which 10 items have the highest Jaccard similarity compared to the first item (i.e., the item from the first review, '2767052')? Report both similarities and item IDs (your answer should be a list of `(similarity, item_id)` tuples). Note that the test data should not be used for this question.

7. Implement a rating prediction model based on the similarity function

$$r(u,i) = \bar{R}_i + \frac{\sum_{j \in I_u \setminus \{i\}} (R_{u,j} - \bar{R}_j) \cdot \mathrm{Sim}(i,j)}{\sum_{j \in I_u \setminus \{i\}} \mathrm{Sim}(i,j)},$$

   (there is already a prediction function similar to this in the provided example code, you can either start from scratch or modify an existing solution). Report the MSE (on the test set) of this rating prediction function when $\mathrm{Sim}(i,j) = \mathrm{Jaccard}(i,j)$.[1]

8. Modify the similarity function from Question 7 to interchange users and items (i.e., in terms of the similarity between users $\mathrm{Sim}(u,v)$ rather than $\mathrm{Sim}(i,j)$), and report its MSE on the test data.

[1] Note that e.g. item averages should be computed on the *training set only*! If the item never appears during training, you should use the global (training) average.