Assignment #3: Regression with Weighted Models

CSCI 373 Spring 2024 Oberlin College

Due: Wednesday April 3 at 11:59 PM

Background

Our third assignment this semester has three main goals:

1. Practice using popular Python libraries for creating machine learning models for **regression** problems,
2. Conduct extensive experiments evaluating machine learning performance, and
3. Practice visualizing and interpreting results using charts.

Gitting Started

You can get started on the assignment by following this link:

https://classroom.github.com/a/l1FvqcXt

Data Sets

For this assignment, we will learn from five pre-defined data sets:

1. **capitalbike.csv**: A data set describing bike rentals within the Capital bikeshare system. The task is to predict how many bikes will be rented hourly throughout the day over a two-year period. The 12 attributes are a mix of 6 categorical and 6 numeric attributes, including information such as the season, day of the week, whether it was a holiday, and current weather conditions. This data set comes the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset
2. **seoulbike.csv**: Another data set describing bike rentals in a metropolitan area (Seoul, South Korea). Again, the task is to predict how many bikes will be rented hourly throughout the day over a two-year period. The 11 attributes are a mix of 2 categorical and 9 numeric attributes, including information such as the season, whether it was a holiday, and current weather conditions. This data set comes the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/560/seoul+bike+sharing+demand
3. **energy.csv**: A data set describing the energy consumption in 10-minute increments by appliances in a low-energy residence in Belgium. The task is to predict how much energy was consumed by appliances. Each of the 27 attributes are numeric and describe measurements from sensors in the residence or nearby weather stations, as well as energy usage by lights. This data set comes the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction
4. **forestfires.csv**: A data set describing forest fires in northeastern Portugal. The task is to predict how much area was burned by forest fires. The 12 attributes are a mix of 2

categorical and 10 numeric values, including date and weather data, as well as the geographic location of the area within the Montesinho park. This data set comes the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/162/forest+fires

5. **wine.csv**: A data set of measurements of wine samples. The task is to predict the quality of the wine (on a numeric scale). The attributes are a mix of 11 numeric measurements from the wine, along with 1 categorical attribute describing the color of the type of wine. This data set is the most popular regression task from the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/186/wine+quality

The file format for each of these data sets is similar to Homework #1 and 2:

- The first row contains a comma-separated list of the names of the label and attributes
- Each successive row represents a single instance
- The first entry (before the first comma) of each instance is the correct label we want to the computer to learn to predict, and all other entries (following the commas) are attribute values.
- Some attributes are strings and others are numbers. New to this assignment: each label is a number since we are solving regression tasks.

Program

Your assignment is to write a program called **regression.py** that behaves as follows:

1) It should take four parameters as input from the command line:

    a. The path to a file containing a data set
    b. The percentage of instances to use for a training set
    c. An integer to use as a random seed
    d. Whether to use max-min normalization of numeric attributes (either `true` or `false`)

   For example, I might run

   ```
   python regression.py seoulbikes.csv 0.75 12345 true
   ```

   which will train regression models to learn how to classify the `seoulbikes.csv` data set using a random seed of `12345`, where `75%` of the data will be used for training (and the remaining `25%` will be used for testing)

2) Next, the program should read in the data set as a set of instances. Every categorical attribute should be converted to a one-hot encoding (as we did in Lab 4). If the user passed in `true` for the fourth command line argument, then each numeric attribute should be rescaled using max-min normalization (as we also did in Lab 4). [Please do **not** rescale the label for this assignment]. After processing the data, it should be split into training and test sets (using the random seed input to the program), similar to the process in Homework 1 and 2

3) Instead of training only a single regression model, the program should train each of the following eight models on the same training set: a linear regression model, a LASSO model, a Ridge Regression model, a SVM with the polynomial kernel of degree 2, a SVM with the polynomial kernel of degree 3, a SVM with the polynomial kernel of degree 4, a SVM with the radial basis function kernel, and a CART decision tree.

   The following documentation should be helpful for determining how to train each of the eight regression models listed above:

   - Linear Regression, LASSO, and Ridge Regression
   - Support Vector Machines
   - Decision Trees

4) Next, predictions on the test set created in Step 2 should be made for each of the eight models trained in Step 4.

5) The **mean absolute error (MAE)** of the predictions of each of the eight models from Step 4 should be calculated and logged to an output CSV file named similar to our results files from Homeworks 1 and 2 using the pattern: `results_<DataSet>_<TrainingPercentage>p_<Seed>.csv` (e.g., `results_seoulbikes_75p_12345.csv`). If the user passed in `true` as the final command line argument so that the data is rescaled, you should include `_rescaled` in the output filename between the `<seed>` and the `.csv` (e.g., `results_seoulbikes_75p_12345_rescaled.csv`).

   The mean absolute error of a sequence of predictions can be calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y^i - \hat{y}^i\right|$$

   where n is the number of instances in the test set, y^i is the true label of instance i, and $\hat{y}^i$ is the prediction for instance i. Here, MAE is the average amount of error between the true labels and the predictions. This measure of error is similar to the sum of squared error (SSE) that is optimized by during training by all eight models, but it is easier to interpret.

Please note that you **are** allowed to reuse your code from Homeworks 1-2 and any labs to help you complete this assignment

Program Output

The file format for your output file is different for this homework since we are not performing classification, so we do not have a confusion matrix. Instead, we want to create a table of the MAE calculated for each of the eight models that you trained in your program.

The file format for your output file should be as follows:

- The first row should be the text `Model,MAE` followed by a newline character (`\n`) to serve as the header of the table.
- Each of the remaining eight rows should be the name of one of the eight models that you trained, then a comma, then the MAE of that model, followed by a newline character.

For example, your output file might look like:

```
Model,MAE
LASSO,17.24486838351211
linear,20.184997665413107
ridge,18.598224592334375
svm_poly2,12.347029970704938
svm_poly3,12.407893630123702
svm_poly4,12.441983573610624
svm_rbf,12.389674614569861
tree,16.84457692307693
```

As in the earlier homeworks, the output for your program should be consistent with the random seed. That is, if the same seed is input twice, your program should output the exact same MAE values.

Programming Languages

In this class, we are using the **Python** programming language for all of our labs and assignments. For this assignment, you are allowed to use any Python libraries (e.g., `pandas`, `numpy`, `scikit-learn`) and not only those originally built into Python.

Research Questions and Experience

Please use your program to answer the four research questions in the provided `README.md` file. There are also three questions at the end of the `README.md` file that you should answer about your experience completing the assignment.

Please remember to commit your solution code, image files, and `README.md` file to your repository on GitHub. You do not need to wait until you are done with the assignment to commit your code. ***Make sure to document your code***, explaining how you implemented the different components of the assignment.

Honor Code

As with Homework 2, **each student is allowed to work with *one partner* to complete this assignment**. Groups are also encouraged to collaborate with one another to discuss the abstract design and processes of their implementations. However, sharing code or answers to the research questions (either electronically or looking at each other's files) between groups is not permitted.

Grading Rubric

Your solution and `README.md` will be graded based on the following rubric:

Followed input and output directions: /5 points
Properly processed the data: /10 points
Correctly trained each of the eight models: /16 points
Correctly makes predictions, calculates MAE, and creates the output file: /9 points
Correctly answered the research questions: /50 points
Provided requested README information: /5 points
Appropriate code documentation: /5 points

By appropriate code documentation, I mean including a header comment at the top of each file explaining what the file provides, as well as at least one comment at the top of each function explaining the purpose of the function (inline comments are also welcome).