CS589: Machine Learning - Spring 2017

Homework 2: Regression

Assigned: Thursday, Feb 16. Due: Thursday, Mar 2 at 11:55pm

Getting Started: Download the assignment archive from Moodle and unzip the file. This will create the directory structure shown below. The data files for each data set are under the 'Data' directory. You will write your code under the Submission/Code directory. Make sure to put the deliverables (explained below) into their respective directories.

```
HW02
--- Data
    |--AirFoil
    |--BlogFeedback
--- Submission
    |--Code
    |--Figures
    |--Predictions
       |--BlogFeedback
```

Data Sets: In this assignment, you will experiment with different regressors on two different regression problems: AirFoil and BlogFeedback. The basic properties of these data sets are shown below. Additional details are given in the README.txt files contained in each data set directory.

Dataset	Training Cases	Test Cases	Dimensionality
AirFoil	751	752	5
BlogFeedback	52397	7624	280

Each data set has been split into a training set and a test set and stored in NumPy binary format. The provided Submission/Code/run_me.py file provides example code for reading in both data sets. Note that test outputs are provided for the AirFoil data set, but not for the BlogFeedback data set, where Kaggle will be used for testing.

Deliverables: This assignment has three types of deliverables: a report, code files, and a Kaggle submission file.

- **Report:** The report will give your answers to the homework questions (listed below). The maximum length of the report is 5 pages in 11 point font, including all figures and tables. You can use any software to create your report, but your report must be submitted in PDF format.
- **Code:** The second deliverable is the code that you wrote to answer the questions. Your code must be Python 2.7 (no iPython notebooks or other formats). You may create any additional source files to structure your code. However, you should aim to write your code so that it is possible to re-produce

all of your experimental results exactly by running *python run_me.py* file from the Submissions/Code directory. 10% of your assignment grade is based on reproducibility and code quality.

- **Kaggle Submissions:** We will use Kaggle, a machine learning competition service, to evaluate the regressors you create for BlogFeedback. You will generate test prediction files, save them in Kaggle format (see run_me.py for example code to generate Kaggle compliant prediction files), and upload them to Kaggle for scoring. Your scores will be shown on the Kaggle leaderboard, and 5% of your assignment grade will be based on how well you do in these competitions. There is a limit of 10 uploads a day to Kaggle. The Kaggle link for BlogFeedback is given below:
  - `https://inclass.kaggle.com/c/hw2-blog-feedback`

Submitting Solutions: When you complete the assignment, you will upload your report and your code using Gradescope.com. Place your final code in Submission/Code, and the Kaggle prediction file for your best-performing BlogFeedback submission (according to public leaderboard RMSE) in Submission/Predictions/BlogFeedback/best.csv. If you used Python to generate report figures, place them in Submission/Figures. Finally, create a zip file of your submission directory, Submission.zip. Upload this single zip file to Gradescope as your solution to the HW02-Programming assignment. Gradescope will run checks to determine if your submission contains the required files in the correct locations. Finally, upload your report as your solution to the HW02-Report assignment. The submission time for your assignment is considered to be the later of the submission timestamps for your code and report.

Academic Honesty Statement: Copying solutions from external sources (books, web pages, etc.) or other students is considered cheating. Sharing your solutions with other students is considered cheating. Posting your code to public repositories like GitHub is also considered cheating. Any detected cheating will result in a grade of 0 on the assignment for all students involved, and potentially a grade of F in the course.

1. (*70 points*) **AirFoil:** For this problem, you will create and evaluate a learning pipeline consisting of a regression method, a hyperparameter selection method, and a feature selection method. You will be provided with the full train and test sets for the data set used in this question. This question does not use Kaggle to evaluate predictions. Use the provided test data only to compute test error. Note that for Question 2, your pipeline will need to be able to run with many more data cases and features, which may impact the choices you make below.

1.1. (*10 pts*) Name and describe the regression method you selected and give its regression function equation. Describe the strengths and weaknesses of the method. Describe the method's key hyperparameters.

1.2. (*5 pts*) Name and describe the hyperparameter selection method you selected. Explain why you selected this method.

1.3. (*10 pts*) Name the feature selection method you selected and describe how it works using equations and/or pseudo code as appropriate. Explain why you selected the method including its strengths and weaknesses.

1.4. (*10 pts*) Explain how you will combine your chosen feature selection method with your chosen hyperparameter selection method to yield a pipeline that performs model learning, feature selection, and hyper-

parameter selection with the goal of minimizing RMSE on test data while being computationally tractable. Provide a pseudo code description of your pipeline, and discuss the trade-offs involved in your chosen design.

1.5. (*10 pts*) Implement your pipeline. You may use Scikit-Learn's regression, hyperparameter optimization, and feature selection implementations to implement your pipeline. You may also write your own methods. You may not use functions in the sklearn.pipeline module in your pipeline implementation. Include the code listing for your pipeline at the end of your report (the code listing does not count against the page limit) as the answer to this question.

1.6. (*5 pts*) Train your selected regression method using its default hyperparameters and no feature selection on the provided training data. Evaluate the learned model on the provided test data using RMSE as the performance measure. Report your test RMSE.

1.7. (*5 pts*) Use your chosen hyperparameter selection method to optimize the hyperparameters for your selected regression method, without applying feature selection. Produce a train-validation curve plot showing train RMSE and validation RMSE versus the hyperparameter values you searched over (if you chose to optimize multiple hyperparameters, provide a plot showing train/validation RMSE versus values of the primary model complexity hyperparameter, with the remaining hyperparameters set to their optimal values). Finally, use the test data to evaluate the test RMSE of your regression model using the optimal hyperparameters you found. Report your final test RMSE result, and the optimal hyperparameter values.

1.8. (*5 pts*) Apply your chosen feature selection method to determine an optimal subset of features. If your chosen feature selection method requires training your regression model, do so using default values for all of the regression model's hyperparameters (except those that may be required for feature selection). Report the set of features that your method selects. Finally, train your regression model (with default hyperparameter values) using just the optimal subset of features your method chooses, and report the resulting test RMSE.

1.9. (*5 pts*) Apply your complete pipeline to both select features and perform hyperparameter selection. Report the optimal set of features your method finds, as well as the optimal hyperparameter values. Finally, use the test data to evaluate the test RMSE of your regression model using the optimal hyperparameters you found, and the optimal subset of features you found. Report your final test RMSE.

1.10. (*5 pts*) Consider your results. What has a larger impact on test performance for this data set and your choice of regression model? Hyperparameter selection or feature selection? Why do you think this might be?

2. (*20 points*) **BlogFeedback:** For this problem, test outputs are not available. Instead, test results will be provided through Kaggle. Starting from the pipeline that you developed for the previous question, your goal in this question is to improve its performance on BlogFeedback.

2.1. (*5 pts*) Run the pipeline that you developed in the previous section on the BlogFeedback training data to determine optimal hyperparameters and an optimal subset of features. Use these optimal settings to make predictions on the test data. Upload your predictions to Kaggle and report the public leaderboard RMSE score.

2.2. (*10 pts*) To improve your pipeline, you can consider adding feature transformations, changing the regression model, using an ensemble, changing (or removing) the feature selection method, changing the way you combine feature selection and hyperparameter selection, etc. If you are able to improve on the test performance of the pipeline you developed for Question 1, include a description of your modified pipeline and give details (equations, descriptions, etc.) for any of the components that you changed. You only need to describe your final best-performing pipeline. If you were not able to improve on your initial pipeline, provide a description of the unsuccessful modifications that you tried, along with the resulting performance numbers (their Kaggle public leaderboard RMSE scores).

2.3. (*5 pts*) Report the test results for your best-performing pipeline as reported by the Kaggle public leaderboard RMSE score (this score will be the same as reported in the first part of this question if you were not able to improve on your initial pipeline). Include the corresponding Kaggle prediction file in Predictions/BlogFeedback/best.csv. Your code should reproduce this file when run_me.py is executed. Your score for this question will depend on how well your pipeline performs relative to baselines set by the course staff, and how well your pipeline performs relative to the top results achieved by the class.

3. (*10 points*) **Reproducibility and Code Quality:** You should aim to design your code so that all experimental results shown in your report can be reproduced by running the run_me.py file, including the output of your best performing predictions for BlogFeedback. You are strongly encouraged to use the Python plotting library Matplotlib to create figures from your results programatically. Your code must also be sufficiently documented and commented that someone else can easily understand what each method is doing. You will be scored on how reproducible your results are, and how well documented and structured your code is.