CS589: Machine Learning - Spring 2018

Homework 4: Regression

Assigned: April 4th, 2018 Due: April 18th, 2018

Getting Started: In this assignment, you will train and evaluate different regression models on a facebook datasets. Please install Python 3.6 via Anaconda on your personal machine. For this homework (and most likely for this course) you will only be using numpy, scipy, sklearn and matplotlib packages. Download the homework file HW04.zip via Moodle. Unzipping this folder will create the directory structure shown below,

```
HW04
--- HW04.pdf
--- Data
--- Submission
     |--Code
     |--Figures
     |--Predictions
```

The data files the data set are in 'Data' directory. You will write your code under the Submission/Code directory. Make sure to put the deliverables (explained below) into the respective directories.
NOTE: All essay style answers must be no longer than 5 sentences, otherwise the question will not be graded and you will lose all points. You are allowed to use any sklearn's built-in function.

Deliverables: This assignment has three types of deliverables:

- **Report:** The solution report will give your answers to the homework questions (listed below). Try to keep the maximum length of the report to 5 pages in 11 point font, including all figures and tables. Reports longer than five pages will only be graded up until the first five pages. 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, it will involve implementing a regression models. Your code must be Python 3.6 (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.
- **Kaggle Submissions:** We will use Kaggle, a machine learning competition service, to evaluate the performance of your regression models. You will need to register on Kaggle using a umass.edu email address to submit to Kaggle (you can use any user name you like). You will generate test prediction files, save them in Kaggle format (helper code provided called Code/kaggle.py) and upload them to Kaggle for scoring. Your scores will be shown on the Kaggle leaderboard. The Kaggle links for each data set are given under respective questions.

Submitting Deliverables: When you complete the assignment, you will upload your report and your code using the Gradescope.com service. Place your final code in Submission/Code, and the Kaggle prediction files for your best-performing submission only for each data set in Submission/Predictions/Fb/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 (NO rar, tar or other formats). Upload this single zip file on Gradescope as your solution to the 'HW04-Regression-Programming' assignment. Gradescope will run checks to determine if your submission contains the required files in the correct locations. Finally, upload your pdf report to the 'HW04-Regression-Report' assignment. When you upload your report please make sure to select the correct pages for each question respectively. Failure to select the correct pages will result in point deductions. The submission time for your assignment is considered to be the later of the submission timestamps of your code, report and Kaggle submissions.

Academic Honesty Statement: Copying solutions from external sources (books, internet, etc.) or other students is considered cheating. Sharing your solutions with other students is also considered cheating. Posting your code to public repositories like GitHub, stackoverflow is also considered cheating. Any detected cheating will result in a grade of -100% on the assignment for all students involved, and potentially a grade of F in the course.

Model selection: For each trained model you will try different parameters. Which set of parameters provides the best performance? Is the one that provides a lower training error? Not necessarily, lower training error does not mean lower testing error (or better generalization). For this homework we will be using mean absolute error (MAE). One way to estimate the best model is via model selection. In other words, after training a model one would like to know how well will that model generalize, *i.e.* how well will the model perform on new unseen data. This cannot be known, but can be estimated via cross-validation. This consists of splitting the training data into K pieces, training using a subset of $K - 1$ pieces, and evaluating the final performance on the unused piece. This is repeated K times, leaving out for testing one different piece each time. The average of the accuracy obtained in this K simulations can be used as an estimation for the out-of-sample error.

Models: You will train decision trees with different maximum depths, nearest neighbors with different number of neighbors and distance metrics, inear models with different regularization parameters, SVM with different kernels and a neural network with different number of hidden units.

Data: You will work with a facebook dataset to try to predict the number of comments a post will get in the next hour from a specific date (base time). The dataset provides the following features:

- 1: Page Popularity/likes
- 2: Page visits
- 3: Page discussion topic
- 4: Page category
- 5-29: Precomputed helpful features
- 30: Total number of comments before selected date
- 31: Number of comments in the last 24 hours

- 32: The number of comments in last 48 to last 24 hours
- 33: Number of comments in first 24 hours after publication
- 34: Difference between 32 and 31
- 35: Base time
- 36: Character count of post
- 37: Post share count
- 38: Whether post was promoted or not
- 39: The hour for which we have the target variable/ comments received.
- 40-46: Day of the week (Sunday...Saturday) post was published.
- 47-53: Day of the week (Sunday...Saturday) of selected base time.

The target outputs for the kaggle set are not provided. You have to predict them and upload the results of the best model to Kaggle (best model chosen using cross-validation). To help you get started we have provided you with sample code in Code/run_me.py file. This file has functions to read in data files and to compute MAE given predicted and original outputs.

Questions:

1. (*17 points*) **Decision trees:**

(10) a. For the provided dataset train 5 different decision trees using the following maximum depths $\{3, 6, 9, 12, 15\}$. Using 5-fold cross-validation, estimate the output of sample error for each model, and report them using a table. Measure the time (in milliseconds) that it takes to perform cross-validation with each model and report the results using a graph (make sure to label both axis; points will be deducted for missing labels).

(7) c. Is the predicted out of sample error close to the test error? Make sure that your report clearly states which model was chosen and what was the predicted out of sample error for it.

2. (*17 points*) **Nearest neighbors:**

(10) a. Train 5 different nearest neighbors regressors using the following number of neighbors $\{3, 5, 10, 20, 25\}$. Using 5-fold cross-validation, estimate the out of sample error for each model, and report them using a table. Choose the model with lowest estimated out of sample error, train it with the full training set, predict the outputs for the samples in the test set and report the MAE (follow the steps as question 1 to report MAE).

(2) b. Is the predicted out of sample error close to the real one? Make sure that your report clearly states which model was chosen and what was the predicted out of sample error for it.

(5) c. It is common to use Euclidean distance as the distance metric to determine the similarity of points in a KNN classifier. However, this is not the only way to measure distance between points (e.g., Manhattan distance, geodesic, Minkowski) . Pick the best performing model from the previous exercise and experiment using two different distance metrics of your choice. How do the results compare? Report the results and give a brief explanation on why you think the results changed the way they did.

3. (*17 points*) Linear model:

(10) b. Train a Ridge and a Lasso linear model with the following regularization constants $\alpha = \{10^{-6}, 10^{-4}, 10^{-2}, 1, 10\}$. Using 5-fold cross-validation, estimate the out of sample error for each model, and report them using a table. Choose the model with lowest estimated out of sample error (out of the 10 trained models), train it with the full training set, predict the target outputs for the samples in the test set and report the MAE (follow the steps as question 1 to report MAE). Make sure that your report clearly states which model was chosen and what was the predicted out of sample error for it.

(7) c. We have discussed that LASSO can be a useful tool for feature selection. Looking at the results from the previous exercise, list 4 features that seem to be less useful and explain why you are drawing that conclusion.

4. (*17 points*) SVM:

(10) b. Train an SVM model using polynomial kernels with degrees 1 and 2, and an RBF kernel. Using 5-fold cross-validation, estimate the out of sample error for each model, and report them using a table. Choose the model with lowest estimated out of sample error (out of the 10 trained models), train it with the full training set, predict the target outputs for the samples in the test set and report the MAE (follow the steps as question 1 to report MAE). Make sure that your report clearly states which model was chosen and what was the predicted out of sample error for it.

(7) c. Give a concise explanation of why you think there is such difference in performance based on the kernel.

5. (*17 points*) Neural Networks:

(10) b. Train a neural network with 1 hidden layer of 10 units, 20 units, 30 units, and 40 units. Using 5-fold cross-validation, estimate the out of sample error for each model, and report them using a table. Choose the model with lowest estimated out of sample error (out of the 10 trained models), train it with the full training set, predict the target outputs for the samples in the test set and report the MAE (follow the steps as question 1 to report MAE). Make sure that your report clearly states which model was chosen and what was the predicted out of sample error for it.

(7) c. Does adding more hidden units always reduce the error in the training set? Why or why not? How about the test set?

6. (*10 points*) Kaggle Competition:

(10) a. Train a regression model of your choice from from the ones we discussed in this homework. Pick k in k-fold cross-validation used to tune hyperparameters. Your task is to make predictions on the test set, kagglize your output and submit to kaggle public leadership score (limited to ten submissions per day). Make sure to list your choice of regression model, hyperparameter range, k in k-folds, your final hyperparameter values from cross-validation and best MAE. Save the predictions associated to the best MAE under Submissions/Predictions/best.csv. Kaggle submission should be made to `https://inclass.kaggle.com/c/hw4-fb-regression`. You are allowed to pre-process data and apply any tricks you have learned to improve your results. However, you **MUST** choose one of the models discussed in this homework.

7. (*5 points*) Code Quality:

(5) Your code should be sufficiently documented and commented that someone else (in particular, the TAs and graders) can easily understand what each method is doing. Adherence to a particular Python style guide is not required, but if you need a refresher on what well-structured Python should look like, see the Google Python Style Guide: `https://google.github.io/styleguide/pyguide.html`. You will be scored on how well documented and structured your code is.