CS534 — Implementation Assignment 1 — Due 11:59PM Oct 7th, 2020

General instructions.

1. Please use Python 3 (preferably version 3.6+). You may use packages: Numpy, Pandas, and matplotlib, along with any from the standard library (such as 'math', 'os', or 'random' - for example).

2. You should complete this assignment alone. Please do not share code with other students, or copy program files/structure from any outside sources like Github. Your work should be your own.

3. Your source code and report will be submitted through Canvas.

4. You need to follow the submission instructions for file organization (located at the end of the report).

5. Please run your code before submission on one of the OSU servers (i.e. babylon01). You can make your own virtual environment with the packages we've listed in either your user directory or on the scratch directory. If you're unfamiliar with any of this process, or have limited access, please contact one of the TA's.

6. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. In particular, **the clarity and quality of the report will be worth 10 pts**. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables. It should be a PDF document.

7. In your report, the **results should always be accompanied by discussions** of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

Linear regression
(total points: 90 pts + 10 report pts)

For the first part of the assignment, you need to implement linear regression, which learns from a set of N training examples $\{\mathbf{x}_i, y_i\}_{i=1}^N$ an weight vector $\mathbf{w}$ that optimize the following Mean Squared Error (MSE) objective:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 \tag{1}$$

To optimize this objective, you need to implement the gradient descent algorithm. Because some features have very large values, for part of the assignment you are asked to normalize the features to the range between zero and one. This will have an impact on the convergence behavior of gradient descent.

Data. The dataset consisted of historic data on houses sold between May 2014 to May 2015. You need to build a linear regression that can be used to predict the house's price based on a set of features. You are provided with two data files: **train** and **dev** (validation), in csv format. You are provided with a description of the features as well. The first column of each file contains the dummy feature taking the constant value of 1 for all examples. Which means, you won't need to add a dummy/bias/intercept term yourself. The last column in the files **train** and **dev** stores the target y values for each example. You need to learn from the training data and tune with the provided validation data to chose the best model.

General guidelines for training. For all parts, you should train your model until the convergence condition is met, i.e., the norm of the gradient is less than $\epsilon = 0.5$. If you find that this specific threshold makes the training time too long for some learning rate values, feel free to use higher values and report the value you used. It is a good practice to monitor the norm of the gradient during the training. You need to report the MSE (the first term in the Eq. 1) on the training data and the validation data respectively for each value of the hyperparameter you tune (learning rate, λ).

Part 0 (20 pts) : Understanding your data + preprocessing. Perform the following steps:

(a) Remove the ID feature. Why do you think it is a bad idea to use this feature in learning?

(b) Split the date feature into three separate numerical features: *month*, *day* , and *year*. Can you think of better ways of using this date feature?

(c) Build a table that reports the statistics for each feature. For numerical features, please report the mean, the standard deviation, and the range. For categorical features such as waterfront, grade, condition (the later two are ordinal), please report the percentage of examples for each category.

(d) Based on the statistics and your understanding of these features/housing prices, which set of features do you expect to be useful for this task? Why?

(e) Normalize all numerical features (excluding the housing prices $\mathbf{y}$) to the range 0 and 1 using the training data. This is equivalent to doing $z_i = \frac{x_i - min(x)}{max(x) - min(x)}$, where z_i is the normalized feature for example x_i, and $max(x)$ and $min(x)$ are the max/min over all examples for a specific feature. The normalization is done on a per-feature basis. Note that when you apply the learned model from the normalized data to test/validation data, you should make sure that you are using the same normalization procedure as used in training. That is, when doing the scaling by max/min/mean anytime on your training data, you must save the training max/min/mean for normalization on any other data. (If curious about normalization, see `https://www.sciencedirect.com/topics/computer-science/max-normalization`)

Part 1 (40 pts). Complete the implementation and explore different learning rates for batch gradient descent. For this part, you will work with the preprocessed and normalized data, and consider at least the following values for the learning rate: $10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$.

(a) Which learning rate or learning rates did you observe to be good for this particular dataset? What learning rates (if any) make gradient descent diverge? Report your observations together with some example curves showing the training MSE as a function of training iterations and its convergence or non-convergence behaviors.

(b) For each learning rate that worked for you, Report the MSE on the training data and the validation data respectively and the number of iterations needed to achieve the convergence condition for training. What do you observe? Between different convergent learning rates, how should we choose one if the MSE is nearly identical?

(c) Use the validation data to pick the best converged solution, and report the learned weights for each feature. Which features are the most important in deciding the house prices according to the learned weights? Compare them to your pre-analysis results (Part 0 (d)).

Part 2 (20 pts). Training with non-normalized data Use the preprocessed data but skip the normalization. Consider at least the following values for learning rate: $100, 10, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. For each value, train up to 10000 iterations (Fix the number of iterations for this part). If training is clearly diverging, you can terminate early. Plot the training MSE and validation MSE respectively as a function of the number of iterations. What do you observe? Specify the learning rate value (if any) that prevents the gradient descent from exploding? Compare between using the normalized and the non-normalized versions of the data. Which one is easier to train and why?

Part 3 (10 pts). Feature engineering (exploration) Similar to Part 0(b), list any modifications to the features provided that you think may be useful for making better predictions. Implement 3 (or more) ideas you have, and report the results. This is an open question, so you are free to do your own exploration. For example, one simple extension is to try the two ways of representing the zip code feature (numerical or categorical), and use the opposite of what you have been for the assignment thus far. Also, you may try combinations of features (such as ratios, products, polynomial features, etc). Please report what you have tried and what (if any) effect it has on the predictions.

Submission. Your submission should include the following:
1) Your source code. **One file for each Part**. The files should be named (for example) **part1.py**, and should run with simply **python part1.py**. You do not need to generate plots in the submission code, please just include those in your report;
2) Your report (see general instruction items 6 and 7), which should begin with a general introduction section, followed by one section for each part of the assignment;
3) Please submit the report PDF, along with a .zip containing the code to Canvas. The PDF should be outside the .zip so it's easier to view the report.