**Assignment 2– Regression, Probability Theory, and Optimization**

***Assignment overview.*** This assignment is designed to review gradient descent regression with regularization and probability theory. This Assignment requires you to load a cleaned version of the dataset ***House Sales***[[1]](#footnote-1), learn and predict using your own implemented linear models, and evaluate the models. Follow the steps as indicated and complete the tasks. You are expected to figure out details of syntax by consulting Python’s Help. Since the material builds from one question to the next, it will be easiest to do them in order. Please answer each question by copying and commenting as needed on the material that you produce in the Jupyder.

The second part of the assignment is a practice with probability theory. You are asked to write a program to plot a histogram for a specific probabilistic process that we can model by drawing samples from a specific distribution. Graduate students will also be asked to provide some analytic solutions.

***Submission.*** Submit a Jupyder file and a pdf file for question 3 called ML\_Assignment2

***Submission deadline.*** Tuesday Feb 6, 2:00 pm.

***Late submission policy.*** No late submission will be accepted.

***Academic Integrity.*** Dalhousie academic integrity policy applies to all submissions in this course. You are expected to submit your own work. Please refer to and understand the academic integrity policy, available at <https://www.dal.ca/academicintegrity>

***Python:*** In addition to the previous python libraries, we will be using [**pandas**](http://pandas.pydata.org/) - Python data analysis library, including structures such as data-frames

***If you have a question:*** Teaching Assistants (TAs) will be present during the labs to help you with any questions you may have. If you still have questions, feel free to email me at [tt@cs.dal.ca](mailto:tt@cs.dal.ca).

**Questions:**

1. **[25 marks] House data:** This Assignment requires you to write a Python script for linear regression. You are not allowed to use any Python public libraries related to regression and metrics. You can compare the results of your program with sklearn, numpy, or scipy linear models, but the whole exercise is to write the algorithm yourself.
   1. Load the ***House sales*** dataset from the houses.csv file and place them in a data-frame df. (Hint: You can use pandas.read\_csv function. New in pandas? Click [here](http://pandas.pydata.org/pandas-docs/stable/10min.html) ). Then generate and show various statistic summary using pandas.DataFrame.describe method. **What** is pandas dataframe? Using pandas.DataFrame methods split the dataset into target value Y (price) and feature matrix X (all feature columns). In addition, extract the sqft\_living column into a feature vector name X\_1.
   2. Write a function named linear\_regression to implement Linear Regression **without using public libraries related to regression**. The inputs of this function should be predictor values (X or X\_1), a target value (Y), a learning rate (lr), and the number of iterations (repetition). The function must build a linear model using gradient descent and output the model (params) and loss values per iteration (loss). Set the iteration to 10000 and calculate and show the *mean squared error (MSE)* for the models obtained from both X and X\_1 predictors (hint: you might write another function named predict to predict the values based on X or X\_1 and params) and plot the learning curve (loss) for both models in one figure (hint: use log scaling plot). Try different learning rates and show the results.
   3. Visualize the best-obtained model for X\_1 using a scatter plot to show price vs area and plot the linear model. Then, visualize the best-obtained model for all features (X) using a scatter plot to show the predicted vs actual target values.
   4. Modify the linear\_regression function in a way that applies Ridge regression, and name them linear\_regression\_Ridge. Then repeat the assignments 1.2 and 1.3. You can thereby use a fixed learning rate that you find appropriate, but you should try and plot different values for the regularization penalty alpha.
   5. Use linear\_regression\_Ridge and write a function named linear\_regression\_Ridge\_momentum in which you add a momentum term. Try different momenta and plot the learning curves with and without momentum for a fixed learning rate.
   6. Modify the linear\_regression\_Ridge\_momentum function in a way that it fits the feature vector X\_1 with a polynomial of order 2 and name it as polynomial\_regression\_optimized. Calculate the MSE, plot the learning curve and show the quadratic model on the scatter plot (price vs area).
2. **[15 marks] Working with random numbers**:
   1. Write a program that rolls two dice (randomly selects a number between 1 and 6 inclusive for each die). Repeat rolling dice 20 times. For each trial, you should add the two numbers that appear on each die and save it in a vector. Plot a histogram of the values you have gathered in the vector. Based on this histogram, **what** are the estimates for the probabilities of each number?
   2. Run your code for 1000 times. **What** is your estimation now?
   3. Change your program and assume that you have one fake die and one correct die. The fake die has 0 instead of 2. This means when you roll this defect die, a random number should be chosen among (1,0,3,4,5, and 6). Plot histogram of the values you have gathered in the vector. **Calculate** the probability of getting a 7 as the sum of the numbers appeared on the dice. **What** is the probability of getting 3?
   4. **[Grad student only]** Try the same problem with seven dice and plot the distribution together with a Gaussian fit. **Report** the mean and variance as calculated directly from the data as well as the fitted parameters.
3. **[10 marks] Theoretical limit of Gaussian classification:** Given are two classes who both have one normal distributed attribute with variance 1. The first class is centered around 1=1, and the second class around

 2=2. Calculate analytically the theoretical limit of the optimal classification accuracy. Provide your answer with a brief outline of the calculation as separate pdf file.

1. This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015. [↑](#footnote-ref-1)