# Machine Learning
# Homework 2
# (Regression and Ensemble Methods)
# (due Noon )

Instructions

1. The deadline for full score is You can get 50% credit for late submission ...).
2. Total marks =
3. You have to type the assignment using a word processing engine, create a pdf and upload on the form. Please note that only pdf files will be accepted.
4. Name the submission as {branch}_{roll_number}_{name}.pdf
5. All code/Jupyter notebooks must be put up as **secret gists** and linked in the created pdf. Again, only secret gists. Not public ones.
6. Any instances of cheating/plagiarism will not be tolerated at all.
7. Cite all the pertinent references in IEEE format.
8. The least count of grading would be 0.5 marks.

1. What is the time complexity of linear regression? Why?
2. (a) Derive an analytical solution for linear regression using $\ell_1$ loss: $|y - X\theta|$. If you cannot solve it analtically (derive a solution similar to the normal equation), explain why?
   (b) Implement a gradient descent based solution. For computing the gradients automatically, use Autograd. Create a matplotlib animation to show the learnt line for the following data. The animation should capture the following: i) gradient reducing over time; ii) the fit becoming better over time.
   (c) Describe scenarios when you will use $\ell_0$, $\ell_1$ and $\ell_2$ loss.
3. (a) What is the difference between multi-colinearity and co-linearity?
   (b) For the following X and y, use scikit-learn to learn a linear model. Solve the problem using normal equations. You may find that one of the matrix in the normal equation is non-invertible. Why can scikit-learn implementation still correctly solve this regression problem?

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \\ 4 & 8 \end{bmatrix}$$

and

$$Y = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

4. (a) Show the usage of scikit learn's linear regression module for the real estate price prediction regression problem. What is the RMS error on the test set?
   (b) Based on the regression co-efficients what can you comment about the importance of different features? Is it correct to assume that larger co-efficients means more important feature?
   (c) Now, standardize the dataset to have all features on a scale of 0 to 1. Re-learn the regression co-efficients and now comment on the importance of different features.
   (d) Use cross-validation to find the optimal set of features to use for regression. For simplicity, say you feature sets of size 1, 2, 3, or 4. What is the optimal feature set as per the validation set and how does it perform on the test set wrt the model learnt on the entire feature set?

5. (a) Implement the RANSAC algorithm from scratch. Do not use the inbuilt scikit-learn version. However, it should be a drop-in replacement for scikit-learn's version. Create a Jupyter notebook showing its utility over vanilla Linear regression on the following dataset.
   (b) Use your RANSAC implementation to do image stitching as done in this post. You are not expected to understand the details of the post and can directly borrow the code. This exercise is to warm you up to some practical applications of RANSAC.

6. Create a Jupyter widget for depicting the impact of coefficient of regularisation on LASSO and Ridge regression. Use the LASSO and Ridge regression implementations from scikit-learn.

   Create a 1X2 subplot matrix in Matplotlib. The columns are for LASSO and Ridge regression and show the linear fit for the two algorithms given the identical dataset. A common regularisation co-efficient ($\lambda$) drives the regularisation in both these cases. Export a GIF recording of the Jupyter widget in action where you vary $\lambda$ on a log-scale from $10^{-4}$ to $10^{4}$. Download the dataset from here.

7. Show the usage of scikit learn's linear regression module on Ridge regression for the real estate price prediction regression problem. Use cross-validation to find the optimum regularisation co-efficient ($\lambda$).

Some useful references for the homework:

1. A video on RANSAC algorithm