Machine Learning
Assignment 3: Regression & optimisation

Due date

This assignment should be submitted to Canvas before 11:59pm on **Friday 18/12/2020**.

Please submit a single ZIP file with your student number and name in the filename. Your submission should contain exactly 2 files:

- A detailed documentation of all code you developed, including the tests and evaluations you carried out. Please make sure that you include a .pdf document with every result you produce referencing the exact subtask and lines of code.
- All Python code you developed in a single .py file that can be executed and that generates the outputs you are referring to in your evaluation. Please make sure that you clearly indicate in your comments the exact subtask every piece of code is referring to.

Please do NOT include the input files in your submission.

You can achieve a total of 35 points as indicated in the tasks.

Objective

The Excel file "diamonds.csv" on Canvas contains prices and other attributes of diamonds. The value of a diamond is determined by its weight (carat), its quality (cut, colour, clarity) and its shape (depth, table).

The goal of this assignment is to train a regression function to determine the value of a diamond from its shape and weight.

Task 1 (pre-processing, 9 points)

Create a function that loads the file and extracts what types of cut qualities [1 point], colour grades [1 point], and clarity grades [1 point] are represented. For each combination of these cut, colour and clarity grades extract the corresponding data-points [1 point]. From now on all processing will be on these subsets corresponding to the various different grades (e.g.

('Ideal', 'E', 'VS2')). Going grade-by-grade split the data-points into features [1 point] and targets [1 point]. Use the carat, depth and table value as features and the selling price as target.
Create a loop going over all combinations of cut, colour, and clarity [1 point] and count the number of data-points within each subset [1 point]. Select only the datasets containing more than 800 data-points for further processing [1 point].

Task 2 (model function, 4 points)

Create a polynomial model function that takes as input parameters the degree of the polynomial, a list of feature vectors as extracted in task 1, and a parameter vector of coefficients and calculates the estimated target vector using a multi-variate polynomial of the specified degree [3 points]. Create a second function that determines the correct size for the parameter vector from the degree of the multi-variate polynomial [1 point].

Task 3 (linearization, 5 points)

Create a function that calculates the value of the model function implemented in task 2 [1 point] and its Jacobian [4 points] at a given linearization point. The function should take the degree of the polynomial, a list of feature vectors as extracted in task 1, and the coefficients of the linearization point as input and calculate the estimated target vector and the Jacobian at the linearization point as output.

Task 4 (parameter update, 5 points)

Create a function that calculates the optimal parameter update from the training target vector extracted in task 1 and the estimated target vector and Jacobian calculated in task 3. To do that start with calculating the normal equation matrix [1 point]. Make sure that you add a regularisation term to prevent the normal equation system from being singular [1 point]. Now calculate the residual [1 point] and built the normal equation system [1 point]. Solve the normal equation system to obtain the optimal parameter update [1 point]. The function should take the training target vector and the estimated target vector and Jacobian at the linearization point as input and calculate the optimal parameter update vector as output.

Task 5 (regression, 4 points)

Create a function that calculates the coefficient vector that best fits the training data. To do that, initialise the parameter vector of coefficients with zeros [1 point]. Then setup an iterative procedure that alternates linearization and parameter update [3 points]. The function should take the degree of the polynomial, the training data features and the training data targets as input and return the best fitting polynomial coefficient vector as output.

Task 6 (model selection, 5 points)

Setup a k-fold cross-validation procedure for all datasets extracted in task 1 [1 point]. Calculate the difference between the predicted prices and the actual sale prices for the test set in each fold [1 point]. Compare different model functions by selection different polynomial degrees ranging from 0 to 3 [1 point]. Use the mean absolute price difference as measure of quality for the different model functions [1 point]. Determine the best polynomial degree for each dataset [1 point].

Task 7 (visualisation of results, 3 points)

Estimate the model parameters for each dataset [1 point] using the selected optimal model function as determined in task 6. Calculate the estimated price for each diamond in the dataset using the estimated model parameters [1 point]. Plot the estimated prices against the true sale prices [1 point] and observe how your estimation corresponds to the actual prices.