

CSE5104 – Fall 2025 Project 1

This assignment is to enhance your understanding of objective functions, regression models, the gradient descent algorithm for optimization, and how to communicate a data analysis problem.

This assignment can be submitted as either individual work, or in pairs.

Programming work

A) Data pre-processing

Pre-process the attribute values of your data by normalizing or standardizing each variable. Keep a copy that was not pre-processed, so you can analyze the difference in optimization.

B) Univariate and multivariate linear regression

In lecture, we discuss univariate linear regression $y = f(x) = mx + b$, where there is only a single independent variable x , using MSE as the loss function.

In practice, we typically have multi-dimensional (or multi-variate) data, i.e., the input x is a vector of features with length p . Assigning a parameter to each of these features, plus the b parameter, results in $p + 1$ model parameters.

Your code must be able to apply the gradient descent algorithm for optimizing univariate and multivariate linear regression models using the mean squared error objective function.

IMPORTANT: Regression is basic, so there are many implementations available, but you MUST implement your method yourself. You may use other functions like matrix math and/or standardization (ex. NumPy or scikit-learn), but the gradient descent algorithm must be implemented by yourself.

C) Regression analysis

Re-fit the multi-variate model using a package that gives you p-values for each predictor variable through regression analysis. Find a pre-processing method that improves these p-values.

Data to be used

We will use the Concrete Compressive Strength dataset in the UCI repository at

<https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>

A copy is also available for download on Canvas. There are 1030 instances in this dataset.

Assuming the columns labels are row 0, use rows 501 - 630 as a testing set of 130 samples. The remain 900 samples will be your training set. This means that you should learn parameter values for your regression models using the 900 training samples, and then use the trained models to predict the 130 testing sample's response values.

Model parameters should never be updated based on the testing dataset.

Part A: Gradient descent optimization

Write code (in the programming language of your choice) that can apply gradient descent optimization to fit a linear regression model, using MSE as the objective function. Apply that code to the concrete data. Evaluate the performance of your models by calculating variance explained (R-squared):

$$1 - \frac{MSE}{Variance(observed)}$$

Play around with hyperparameter values until you achieve a positive variance explained on the training data with at least two of the predictors. You can use different hyperparameter values for different models.

Once you're satisfied with your training fits, use your models to predict on the testing data.

Results to record:

- Your univariate model fits from training:
 - Set 1: the predictor values are normalized or standardized, and each predictor feature is used to fit its own univariate model
 - Set 2: the predictor values are not transformed in any way, and each predictor feature is used to fit its own univariate model
- Your multi-variate model fits from training:
 - Set 1: the predictor values are normalized or standardized, and all predictor features are used in the same multivariate model
 - Set 2: the predictor values are not transformed in any way, and all predictor features are used in the same multivariate model


Part B: Regression analysis

Find a function or package or library (in the programming language of your choice) that can perform statistical analysis of features in a linear regression model.

Results to record:


- P-values of each feature when fit to:
 - Set 1: the predictor values are normalized or standardized, and all predictor features are used in the same multivariate model
 - Set 2: the predictor values are not transformed in any way, and all predictor features are used in the same multivariate model
 - Set 3: the predictor values have been log transformed ($\log(x + 1)$), and all predictor features are used in the same multivariate model

Part C: Submit a Business Insights Report

Below is a template for a report you might submit to a client who hired you to use data mining to inform future decisions. Do some background research on why concrete compressive strength would be worthwhile to analyze for a business, government, or other organization, then fill in the  blanks based on the results of your work.

Title: Concrete Compressive Strength: Data-Driven Insights and Recommendations

1. Introduction

In this report, we present our findings from a data analysis aimed at understanding concrete compressive strength. Our goal is to provide clear, actionable recommendations based on data-driven insights that can help inform your decisions regarding  **<fill in a business/government/organization motivation here>**

2. Key Findings



<fill in your actual findings from your gradient descent and regression analysis work>

Some examples of the kind of findings you can report:

- *Certain concrete components significantly influence compressive strength*
- *Optimal proportions of components can lead to substantial improvements*
- *The use of specific techniques can enhance efficiency and quality during production*

3. Recommendations



<fill in at least two (2) actionable steps and include at least two (2) visuals that support those steps. The visuals can be figures or tables>

4. Benefits



<what are the expected benefits from following your recommendations? More profit? Greater safety? Make sure that these benefits would be of relevance to your chosen organization>

5. Conclusion and Next Steps



<fill in with a summary of what your recommended actions would accomplish, and include at least one follow-up action you would take to confirm your recommendations have the intended effect>

Part D: Submit a Reproducibility Report

Below is a template for a report you might submit to ensure your work is recorded and reproducible by a fellow data analyst. Fill in the sections to report your work for this project:

Title: Reproducibility Report for Concrete Compressive Strength Data Mining Analysis

The objective of this analysis is to understand the key factors of Concrete Compressive Strength.

1. Data Description

Dataset:

- Source: UCI Machine Learning Repository
- Variables: Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate, Age, Concrete Compressive Strength

Preprocessing Steps:

- Missing value handling: None (dataset is complete)
- Transformations:

<fill in any data transformations you applied for any of the methods>

2. Methods

Gradient Descent Algorithm:

- Pseudocode:

<fill in pseudocode of your gradient descent algorithm here>

- Parameters: **<what was your learning rate? Max iterations?>**

Regression Analysis:

<fill in how you got regression analysis results, e.g. the python library and/or the function call>

Implementation:

[REDACTED]

<fill in any software or environment dependencies for running your code. Include any recommended steps for loading in the data before fitting the models>

3. Results

Gradient Descent:

- Final weights: **[REDACTED]** **<insert a table of all the relevant MSE and R-squared values for fitting all univariate and multivariate models, both training and testing results>**
- Loss over iterations: **[REDACTED]** **<insert a figure that illustrates the successful decrease in MSE as your gradient descent algorithm iterated>**

Regression Analysis:

- p-Values: **[REDACTED]** **<insert a table of all the predictor variables and their p-Values for fitting the multivariate linear model on transformed and untransformed data>**

4. Discussion

Interpretation:

[REDACTED]

<fill in some interpretations of the model results, such as the influence of predictor variables on concrete compressive strength>

Strengths and Limitations:

[REDACTED]

<fill in what you think the models did well, and what you think the models could have done better>

5. Appendices

[REDACTED]

<attach your code as a file, or include here a link to a GitHub repository of your code>

Due date

Friday, October 17 (midnight, STL time). Submission to Gradescope via course Canvas.

If you are doing this work in pairs, make sure to check that the submission has **both** your names. Both members of the group should be able to see the submission on their own Gradescope page. If you do not see the submission, then you will not receive credit.

A one-week late extension is available in exchange for a 20% penalty on late submissions.

Each part of the assignment can be submitted separately at your own pace. The late penalty will only be applied to parts that are late, not the project as a whole.

Rolling TA feedback

Every week, TAs will review submitted work and provide feedback on its current state. It's recommended that you use this feedback to re-do your submission and re-submit for a higher grade, provided the final deadline has not yet passed.