Due: Mar. 20, 2023 @ 11:59 p.m.

General notes to keep in mind:

- *All deliverables for the assignment must be submitted as a **single ZIP file per group** via the Brightspace D2L course shell. Submissions containing multiple ZIP files per group or those with a file that is not in the ZIP format will NOT be graded.*
- *The code submitted must be written purely using the Python programming language and it should execute within the Python 3.11.1 interpreter running on the Windows operating system (version 10 or above). The submitted code should NOT require external python modules other than scikit-learn 1.2.1 , matlplotlib 3.6.3, pandas 1.5.3 and their dependencies.*
- *Read the "Assignment code submission requirements" carefully and prepare the code accordingly. It is your responsibility to ensure that the submitted code executes. If the grader is unable to execute your code and/or your code does NOT adhere to the submission requirements, your code may not be graded.*
- *The written responses required to the questions in the assignments must be compiled into **single PDF** file named as `report.pdf`. You are encouraged to use LaTeX for typesetting your written responses, but however, the use of Microsoft Word™ or any other such programs is also acceptable.*

Predicting strength of high-performance concrete

Concrete is one of the most commonly used building materials. The three basic ingredients of concrete are Portland cement, natural sand and water. However, additional materials such as fly ash, blast furnace slag and superplasticizer are generally added to improve the strength of concrete and create the so-called high-performance concrete[1]. The characterization of high-performance concrete's strength is important in building design to avoid failures (see Figure 1a). Concrete samples can be tested using experimental compressive strength tests (see Figure 1b, Figure 1c). Further, data based modeling may be used to predict compressive strength of new high-performance concrete mixtures. The goal of this assignment is to explore the **development of linear regression models for concrete compressive strength prediction** based on the relative amounts of ingredients used in a given concrete mixture and the age of the concrete.

(a) Buckling of a concrete column

(b) Testing concrete for compressive strength

(c) Failure under compressive loading

Figure 1: Illustration of concrete strength. ©2016 civildigital.com

[1] I-Cheng Yeh, "Modeling of strength of high performance concrete using artificial neural networks," Cement and Concrete Research, Vol. 28, No. 12, pp. 1797-1808 (1998)

Data

The data for this assignment can be downloaded from here and it is attributed to Prof. I-Cheng Yeh, Chung-Hua University, Taiwan[1]. The data consists of a total of 8 features, among which 7 features relate to the relative amounts of the ingredients (Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate respectively) in a concrete mixture, while a single feature denotes the age of the concrete. Experimentally determined compressive strength for the given concrete mixture is provided as the outcome variable. The "training" and "test" datasets consisting of 800 and 100 samples respectively are given in the `train.csv` and `test.csv` files respectively. Further, another 130 samples have been gathered and will be used as the "independent test" dataset to perform a "blinded" validation of your submitted regression model.

Performance reporting convention

Always report both the residual standard error (RSE) and the R^2 statistic performance metrics when summarizing the performance ("Err") of a regression model.

Question 1 [25 marks]

Train a multivariate *ordinary least squares ("simple") linear regression* model to predict the compressive strength of an input concrete mixture based on the relevant features. Estimate the "Err" using both the validation approach (i.e., train a model on the "training" dataset and test on the "test" dataset) as well as using a cross-validation (CV) approach (i.e. only using the "training" dataset). Discuss the choice of the number folds used in your CV approach, and compare the "Err" estimates obtained using the validation and CV approaches.

Question 2 [25 marks]

Now train a multivariate *Ridge regression* model for the above concrete compressive strength prediction task. Use the "training" dataset and a CV based grid-search approach to tune the regularization parameter "α" of the Ridge regression model. Using the "best" "α" setting, re-train on the entire "training" dataset to obtain the final Ridge regression model. Estimate the "Err" of this final model on the "test" dataset. Plot the performance of the models explored during the "α" hyperparameter tuning phase as function of "α". Compare the performance of the Ridge regression model with that of the "simple" linear regression model.

Question 3 [25 marks]

Repeat the above experiment with a multivariate *Lasso regression* model. Plot the performance of the models explored during the "α" hyperparameter tuning phase as function of "α". Compare the performance of the final Lasso regression model with that of both the Ridge regression and the "simple" linear regression models.

Question 4 [25 marks]

Leveraging the experience you gained from the experiments thus far and/or conducting further experiments using the rich array of regression models available in the `scikit-learn`, design the "best" regression model for predicting compressive strength of a concrete mixture from the 8 features mentioned previously. The only restriction is that, you may not use datasets other than the ones provided as part of this assignment. Submit this "best" regression model as the following method:

```
def ytest = predictCompressiveStrength(Xtest, data_dir):
"""Returns a vector of predictions of real number values,
corresponding to each of the N_test features vectors in Xtest

Xtest     N_test x 8 matrix of test feature vectors

data_dir  full path to the folder containing the following files:
          train.csv, test.csv
"""
```

The above method will be evaluated on the "independent test" dataset by the grader to determine the "Err" summarized by the R^2 statistic. See note below regarding the grading rubric for this question.

Note on grading

The grading for Question 1, Question 2 and Question 3 will be based on the appropriateness of the submitted code and the written responses. The grading for Question 4 will be based on the relative performance of your trained model. The submission(s) with the best performing model (referred below as 1^{st} ranked model) in terms of the R^2 statistic (rounded to 4 decimal places) will receive full marks on Question 4 (i.e., 25 marks). All other submissions will receive marks that are proportional to the decrease in performance of their model with respect to the 1^{st} ranked model. For example, if the R^2 statistic of the model of a given submission is 10% lower than the 1^{st} ranked model, then that submission will receive 22.5 marks for Question 4.