|  |  |  |
| --- | --- | --- |
| **AIN 380** | **Homework #7**  **Ridge and Lasso Linear Regression ML** | **Due:** |

In the field of engineering, it is crucial to have accurate estimates of the performance of building materials. These estimates are required in order to develop safety guidelines governing the materials used in the construction of buildings, bridges, and roadways. Estimating the strength of concrete is challenging and one of the critical components. Although it is used in nearly every construction project, concrete performance varies greatly due to a wide variety of ingredients that interact in complex ways. As a result, it is difficult to accurately predict the strength of the final product. A model that could reliably predict concrete strength given a listing of the composition of the input materials could result in safer construction practices. While there are several reasons for the failure of concrete structures, it can be fatal when it occurs. For example, on 24 June 2021, the Champlain Towers South building in Miami collapsed, resulting in the death of 98 people.

The strength of concrete is dependent on several variables. The purpose of this assignment is to use the **sklearn** library to assess a few machine learning models that include many of these variables. The dataset used to create the machine learning models contains 1,005 examples of concrete strength with eight features describing the components used in the mixture. These features are thought to be related to the final compressive strength and they include the amount (in kilograms per cubic meter) of cement, slag, ash, water, superplasticizer, coarse aggregate, and fine aggregate used in the product in addition to the aging time (measured in days). The dataset also contains the dependent variable, which is called MSA and stands for *megapascal, which measures the compressive strength of the concrete. In the US and UK, PSI (Pounds per Square Inch) is used rather than MSA to represent this compressive strength and is roughly equivalent to MSA (Factually, one psi value is about 0.0068915 MPA, but you do* ***NOT*** *need to reference this fact in this assignment). MSA is the unit used in this assignment.*

1. Download the dataset named *Concrete\_samples.csv* into a Pandas DataFrame. Confirm that all of the data is numeric and that there are no missing data values. Then, scale the data in the DataFrame using the sklearn “Standard Scaler” as discussed in class. Be sure to retain the column headings. **<4pts>**
2. You are going to create Linear Regression, Ridge and Lasso machine learning models for three different training/test splits of the dataset. In addition, you are to try three different values for each split. Specifically, for each of the three split percentages identified below, you are to use alpha values of 0.1, 0.01, and 0.001. **<18 pts>**

So, to be clear, one Python output is required for **each** of the following and the output is to list the coefficients **AND** the accuracy score for **each** of the three machine learning models (so 9 printed outputs):

1. 20% test size – 80% training size split Linear Regression   
   20% test size – 80% training size split Lasso with alpha = 0.1  
   20% test size – 80% training size split Ridge with alpha = 0.1
2. 20% test size – 80% training size split Linear Regression   
   20% test size – 80% training size split Lasso with alpha = 0.01  
   20% test size – 80% training size split Ridge with alpha = 0.01
3. 20% test size – 80% training size split Linear Regression   
   20% test size – 80% training size split Lasso with alpha = 0.001  
   20% test size – 80% training size split Ridge with alpha = 0.001
4. 30% test size – 70% training size split Linear Regression   
   30% test size – 70% training size split Lasso with alpha = 0.1  
   30% test size – 70% training size split Ridge with alpha = 0.1
5. 30% test size – 70% training size split Linear Regression   
   30% test size – 70% training size split Lasso with alpha = 0.01  
   30% test size – 70% training size split Ridge with alpha = 0.01
6. 30% test size – 70% training size split Linear Regression   
   30% test size – 70% training size split Lasso with alpha = 0.001  
   30% test size – 70% training size split Ridge with alpha = 0.001
7. 40% test size – 60% training size split Linear Regression   
   40% test size – 60% training size split Lasso with alpha = 0.1  
   40% test size – 60% training size split Ridge with alpha = 0.1
8. 40% test size – 60% training size split Linear Regression   
   40% test size – 60% training size split Lasso with alpha = 0.01  
   40% test size – 60% training size split Ridge with alpha = 0.01
9. 40% test size – 60% training size split Linear Regression   
   40% test size – 60% training size split Lasso with alpha = 0.001  
   40% test size – 60% training size split Ridge with alpha = 0.001

As an example, ONE of the 9 printed outputs may look like that shown on the next page. **Be sure to identify the split AND the alpha for each run to not lose points!**

1. **<5 pts>**Which of the 9 models provides the highest accuracy score for a training data set?   
   \_\_\_All accuracy scores were similar, but the models that used 20% test were a little better than the others, and all within a fraction of each other\_\_\_\_\_  
   **Why do you think this is so?** I’m really not sure, I thought that the 40% test would be the best because of possible overfitting issues, but honestly since there is such a small difference between the different models, I’m really struggling to make an accurate assumption. I’m not sure if I did something wrong in my code, but I followed the examples we did in class exactly, and I don’t understand why 40% isn’t the right answer.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If this model were chosen as the machine learning model to predict an MPA for a set of inputs, what is the linear equation defined for the model?  
\_\_\_\_\_It would be all of the coefficients that are associated with that model, multiplied by each individual input, plus the intercept. But because I’m not sure which model to use since they are all so similar for the model 1 lasso the equation is .12a + .1b + .095c - .16d + .2e + .02f + .02g + .11h - 30.19.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **<7 pts>**Which of the 9 models provides the highest accuracy score for a test data set? \_\_\_All of the models were similar, but the models that used 20% test wielded the best output. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   Why do you think this is so? \_\_\_\_\_\_Like I said for the question above, all of the answers were really similar, but I would assume that this was the best answer because the overfitting issues weren’t as bad since we used a smaller test percentage compared to the other two. But if I had to make an suggestion as to why the accuracy scores are so bad, I would suggest that we need to use less variables, since them all combined are clearly bad predictors.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   If this model were chosen as the machine learning model to predict an MPA for a set of inputs, what is the linear equation defined for the model?  
   \_\_\_\_\_It would be the same answer as above, which is .12a + .1b + .095c - .16d + .2e + .02f + .02g + .11h - 30.19\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

How confident should a builder be using this machine learning model?  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Not confident at all, the best accuracy score was at a 62% so that means we most likely wouldn’t get any accurate predictions if we had to use this in a real world application.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print your Python code file (you need only do so for one of the models) and all 9 outputs and submit along with this sheet for credit. In addition, upload your Python code in a zip file with the data file to the course Canvas site.

A sample output for ONE run is provided on the next page 🡪

|  |
| --- |
| Run: 20%-80% alpha = 0.1  Linear coefficients:  The coefficient for cement is 0.7715730472648976  The coefficient for slag is 0.559141398160738  The coefficient for flyash is 0.37803878293835097  The coefficient for water is -0.20952053946693538  The coefficient for superplasticizer is 0.0768213973973341  The coefficient for coarseaggregate is 0.10727182754620543  The coefficient for fineaggregate is 0.12602433035781768  The coefficient for age is 0.4391983741100994  Ridge coefficients:  The coefficient for cement is 0.7700818459435413  The coefficient for slag is 0.5576938568055403  The coefficient for flyash is 0.37675181906248134  The coefficient for water is -0.21054606313847268  The coefficient for superplasticizer is 0.07679659013019605  The coefficient for coarseaggregate is 0.10624619778364926  The coefficient for fineaggregate is 0.12471166673449596  The coefficient for age is 0.43907526542161046  The intercept is 0.0007804922070795538  Lasso coefficients:  The coefficient for cement is 0.36683692484132313  The coefficient for slag is 0.14869795227789595  The coefficient for flyash is 0.0  The coefficient for water is -0.15945851692584412  The coefficient for superplasticizer is 0.16775062920604297  The coefficient for coarseaggregate is -0.0  The coefficient for fineaggregate is -0.0  The coefficient for age is 0.2830669661530609  The intercept is 0.001521470581385161  Linear model accuracy:  0.596472150557698  0.6245250808411705  Ridge model accuracy:  0.5964718337626582  0.6245877174311083  Lasso model accuracy:  0.5030576157517481 0.5339044320014084 |