D209 Performance Assessment Task 2

November 5, 2021

**Part A**

1. Using the medical dataset, a question that an organization could ask is to ask what factors within the dataset are affected by Age and how does Age affect other factors within the dataset.
2. The main goal of this data analysis is using a Lasso regression to determine how predictable other factors of the dataset are when given ages as well as the age of a patient when presented with factors.

**Part B**

1. I have chosen to use lasso regression because it would help eliminate predictors in the dataset since it is mainly used to help eliminate predictor variables that have higher correlations to each other. The hope is that we can have a result of as close to 0 as possible since it would show that there is very little correlation between variables and would then be able to perform a regression on the data since one of the biggest problems is making sure that the variables are not directly related or correlated in any way for the analysis to work. I hope that I will be able to predict 2 things, a patient’s age given the correct data, and the correct data given a person’s age.
2. One of the big assumptions that is needed in a lasso regression analysis is that the dataset being analyzed has a high level of multicollinearity. This means that there are factors working with each other that show correlations.
3. For this analysis I chose to use python because that’s the best language I can use, plus it has a wide arrange of libraries that allow for various machine learning, data mining, and analysis tools and visualizations. Pandas was selected since it’s needed in order to work with databases. Numpy was used for calculations and numerical functions such as absolute values. Matplotlib was used in order to create figures and save them into a jpg format. Scipy was used for calculating zscores and standard deviations to remove outliers. Seaborn was imported to add graphical capabilities to the charts being created. Sklearn imported train\_test\_split, LassoCV, mean\_squared\_error, and RepeatedKFold all to help train, test, and fit the model that I was creating. I also imported the warnings library because I would get a lot of false warnings that were annoying so I could hide them.

**Part C**

1. One of the first things that I had to do to help clean the data was to replace all categorical variables with numerical variables in order to perform an analysis on them. I created a dictionary and replaced certain values in columns with numerical ones in order to analyze them. I also created a heatmap in order to see different relationships between certain variables and target the variables I wanted to use as my predictor variables in my analysis for Age.
2. For my analysis, I decided to throw everything but the kitchen sink into my analysis in order to really see what was operating under the hood. I threw in City, State, County, Zip, Lat, Lng, Population, Area, TimeZone, Job, Children, Age, Income, Marital, Gender, ReAdmis, VitD\_levels, Doc\_visits, Full\_meals\_eaten, vitD\_supp, Soft\_drink, Initial\_admin, HighBlood, Stroke, Complication\_risk, Overweight, Arthritis, Diabetes, Hyperlipidemia, BackPain, Anxiety, Allergic\_rhinitis, Reflux\_esophagitis, Asthma, Services, Initial\_days, TotalCharge, and Additional\_charges. I checked for duplicates, outliers, and then used a dictionary to replace certain categorical columns with numbers to allow for a proper analysis. I then created a heatmap and saw what factors appeared to have the highest correlation to something within the dataset and reduced the table into Zip, Lat, Lng, Population, Age, Initial\_days, TotalCharge, and Additional\_charges, which were all continuous variables, while also taking away ReAdmis and HighBlood as categorical variables for my analysis.
3. For my analysis, I first created a data file with all the columns that I wanted to check. I then checked for duplicates or null values, followed by removing all rows that were outside 3 standard deviations, or 99.7 percentile. Next, I had to create a histogram to check the distribution of all the variables followed by a heatmap to see all variables compared to every other variable in a bivariate analysis. Using the heatmap, I took out the categories that were highlighted and then used them as my dependent variables and stuck to using Age as my dependent variable. I then trained my data using a train\_test\_split method. Next, I used a LassoCV function in order to build a lasso regression model and then fitted it with my 2 datasets containing the Ages and the other containing the predictor variables. I then scored my function to test the variation of my function which came out to 0.889 which is not bad at all, but not very good. It demonstrates that my model was able to predict the correct age of the patient roughly 90% of the time using a lasso regression, which is impressive, but had an alpha variable of 0.02 which showed that there wasn’t much correlation between the variables which is necessary for the use of a lasso regression.
4. The cleaned dataset is attached

**Part D**

1. This analysis split the training and prediction data into 2 csv files which have been attached.
2. The analytical technique I used for this analysis is a lasso regression because I originally wanted to use a random forest, but the calculations started getting so large that it took way too long to calculate. I still think a random forest is the better option, but the sacrifice for time of the generation of the trees makes it almost impossible. Since I was mainly focusing on lasso regression, I would be focusing on a linear equation. The whole goal with the lasso regression is to reduce the sum of R2 values as much as possible. The basic structure would be a *y = mx + b* but with lasso regression you’re using cross products so you get a slightly different function of *y = w[0] X x[0] + w[1] X x[1] + … + w[n] X x[n]* where we are checking the cross product of various factors in accordance to age and other predictors in order to find out our lasso regression results. Further calculations can be seen in the attached file.
3. See attached file for code implementation.

**Part E**

1. For my analysis the accuracy was approximately 89% which is pretty good. We had an alpha value of 0.02 which wasn’t very good since it describes the discrepancy between the data and the estimation model. This value demonstrates that there is not much correlation between the variables which is needed for lasso regression, so these results are questionable. However, being able to predict the ages given the other factors roughly 90% of the time still bodes rather well. The mean squared error that was calculated was 46.986 which the variation of squares in this analysis. This means that the estimated values vs true values were distant given the ages. Its showing that the differences between the values for age swung wildly all over the place depending on the age range. This level of error is strange given that our regression predicted the accuracy roughly 90% of the time.
2. After going through this full analysis, I have determined that while we were able to predict the ages of the patients well, this is more than likely a fluke since there wasn’t a high level of significance in our results given an alpha value of 0.02 and a high MSE value of 46.986. While I would personally say these results are decent enough to use as a general guide toward predicting ages, saying that it was reliable would be a little bit of an overestimate.
3. One major limitation of this analysis is the data not being heavily correlated. For a lasso regression to work, there needs to be high levels of multicollinearity in the data, but after running the heatmap on the data it didn’t appear to have too much that was correlated. The ones that were selected seemed to be correlated with other variables, but not all together which would hinder the ability of the lasso regression to accurately assess the fit of our model in order to create our predictions.
4. The recommended course of action for this analysis would be to try using other forms of regression. While a lasso regression here may have given us decent predictions, the use of other methods would help verify that our predictions held any water. The next step I would say is to perform a ridge regression and compare the models and their results. Followed by that, I’d then use a random forest in order to do a triple check to increase the accuracy even further.

**Panopto video:** <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=c2db3e42-18b4-4794-b5e9-adca01729d2b>

**References:**

<https://www.statisticshowto.com/lasso-regression/> Lasso Regression: Simple Definition, Statistics How To

<https://www.statisticshowto.com/multicollinearity/> Multicollinearity: Definition, Causes, Examples, Statistics How To

<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html#sklearn.linear_model.LassoCV> sklearn LassoCV, scikit-learn documentation

<https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b> Ridge and Lasso Regression: L1 and L2 Regularization, towards data science

<https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/> A Complete Tutorial on Ridge and Lasso Regression in Python, Analytics Vidhya

<https://www.geeksforgeeks.org/python-mean-squared-error/> Python | Mean Squared Error, GeeksforGeeks