ML Assignment 1 Report

Jialun Darren Huang (S3755729)

Table of Contents

Correlations between variables 3
Deciding on my model 3
Regression Metrics Used 3
Linear vs Polynomial Regression 3
K Fold Cross Validation with Linear Regression 3
Regularisation 4
What happens after? 5
Conclusion 5
References 6
Appendices 6

Correlations between variables

Correlations of features is seen in appendix A

Based on the correlation heatmap, we can see that the target (TARGET_LifeExpectancy) has a strong correlation with some features. However, more importantly, it has weak correlations with other variables as well.

Deciding on my model

Regression Metrics Used

Before I begin, I would like to state the regression metrics I used.
I used Mean Square Error (MSE), Mean Absolute Error (MAE), R Squared score (R2Score), Training and Validation Set Score.

Linear vs Polynomial Regression

Firstly, I have to decide whether I want to use Linear or Polynomial Regression. 80% of the data is classified as training data while 20% of the data is classified as validation data

I created a code for polynomial regression that takes in a degree of 1 (linear) to 5.
Any more than 5 takes a lot of processing power and causes my computer to crash. For sure, we could use AWS, however, it still takes awhile to process and it does not make sense to go anywhere more than 5 as the MSE, MAE and R Squared score graphs show that values start getting way off the charts. This is because of overfitting and we are using the validation data to test the model.

While running the whole code multiple times, the training and validation data changes(not including k fold cross validation). The MSE, MAE and R Squared score values changes and at times, favour degree = 2 over degree = 1. However, one thing that did not change was the difference in Training Set and Validation Set score. When looking at this difference, degree = 1 is always seen to be significantly better than degree = 2. Therefore, with this stabilised result, I have decided to use degree = 1 over degree = 2, which is essentially, Linear Regression.

Degree = 1 vs Degree = 2 for Difference in Training and Validation Set score is seen in Appendix B

Logistic regression is not recommended here as logistic regression is more used in binary classifications (0 and 1), while the dataset has more than 2 types of values in the features.

K Fold Cross Validation with Linear Regression

My next step was to use k fold cross validation. The training and validation data may be having bias and this will affect the hyperparameters of the model. This will affect the prediction if I were to use just one set of training and validation data.

With k fold cross validation, I compared all 19 sets of model. I used k = 19 since there is 2071 data in the whole train.csv, I would have a nice cut of 109 data in each fold, which results in 1962 training data and 109 validation data in each cross validation. I calculated all the metrics as said before, and calculated the

average MSE, average MAE and average R Squared score. With that, I displayed the models that is closest to the average MSE, average MAE and average R Squared score.

Table of Average Metrics and Best model is seen in Appendix C

I decided to priortise MSE, and pick the model with the lowest MSE, in this case(picture), it is the 8th linear regression model (index starts from 0). R Squared score is not a good gauge in determining the best model as there are issues, like goodness of fit, it does not measure[1]. Comparing MAE and MSE, I chose MSE as I believe that the data, which comes from Global Health Observatory data repository, has been cleansed from errors, and I assume that the modifications made as stated in the assignment has not modified the data significantly.

Graph of MSE against each fold is seen in appendix D

However, just a note. If there has been huge data changes and many outliers, I would pick the model with the lowest MAE, as it is less punishing with the huge number of outliers.
But as for now, I am sticking with the model with the lowest MSE.

Regularisation

I used Ridge and Lasso Regularisation on my chosen model.

Ridge regression did not go so well. Based on my findings, when alpha increases just slightly, there is a huge spike increase in MSE and MAE, while a huge drop in R Squared score. There does not seem to be any improvement as alpha increases so I decided not to use Ridge Regression.

Regression metrics when alpha changes (Ridge) is seen in Appendix E

I managed to find a value for alpha that would work well in Lasso Regression. With lasso regression, when alpha is less than 0.027, MSE and MAE decreases while R Squared score increases. As alpha increases over 0.027, MSE and MAE starts increasing and R Squared score decreases. With that, there is a nice value for alpha that allows Regularisation to work.

Regression metrics when alpha changes (Lasso) is seen in Appendix F

[1] Clay Ford, *Is R-squared Useless?* (Oct 2015)
https://data.library.virginia.edu/is-r-squared-useless/

Finally, this is my model and the statistics about it

```
Lasso Regression (alpha = 0.027):
y-intercept:
[58.63023815]

coefficients:
[ 0.00000000e+00  0.00000000e+00  2.25582720e+00 -1.92365930e-02
 -0.00000000e+00 -0.00000000e+00 -0.00000000e+00  2.45855056e-02
  0.00000000e+00 -0.00000000e+00  2.48261989e-02 -1.87393813e-03
  4.04402563e-03  0.00000000e+00  1.90226021e-02 -3.54449144e-01
  0.00000000e+00 -0.00000000e+00 -4.35959741e-02 -0.00000000e+00
  9.21201090e+00  1.68405485e+00]

Training set score: 0.71
Valid set score: 0.68

Mean squared error:  26.457044549968742
Mean absolute error:  3.860773684165635
R squared score:  0.6713788156961846
```

What happens after?

I created a .csv file and save the predicted targeted data based on test.csv

```
targetLifeExpectancy = pd.DataFrame(data = predictTestY, columns = ["Target_LifeExpectancy"])
mySolution = testingData["ID"]
mySolution = pd.concat([mySolution, targetLifeExpectancy], axis = 1)

mySolution.to_csv("myExpectedSolution.csv", index = False)
```

Saving my expected targeted values in myExpectedSolution.csv

An example of the .csv file content is seen in Appendix G

Conclusion

My best model is the linear regression with lasso regularisation as said with reasoning from above. This may not be the best model, but I can say that it is one of the better models that has its parameters tuned, aimed all for better prediction all around.

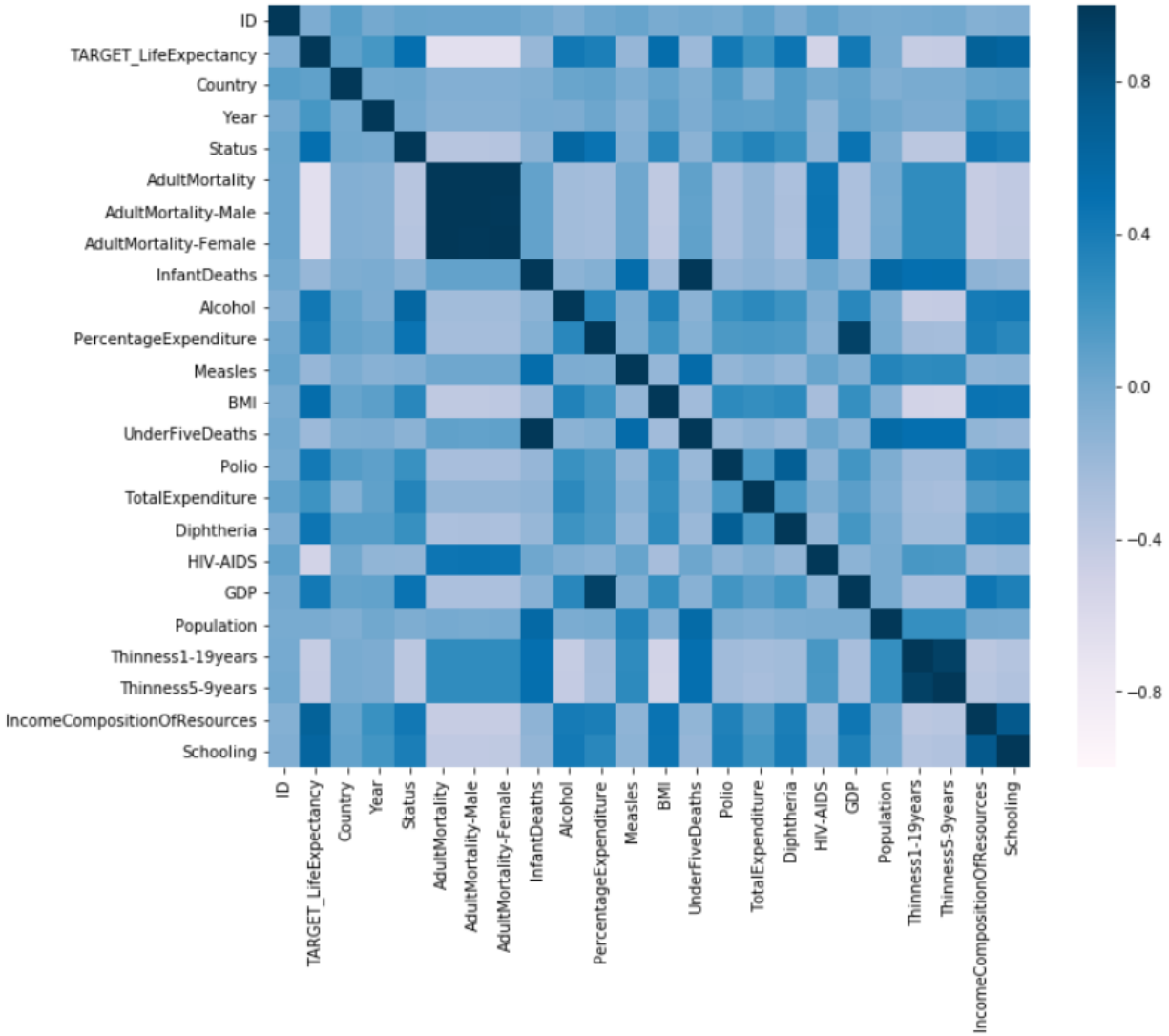
References

[1]: Clay Ford, *Is R-squared Useless?* (Oct 2015) https://data.library.virginia.edu/is-r-squared-useless/

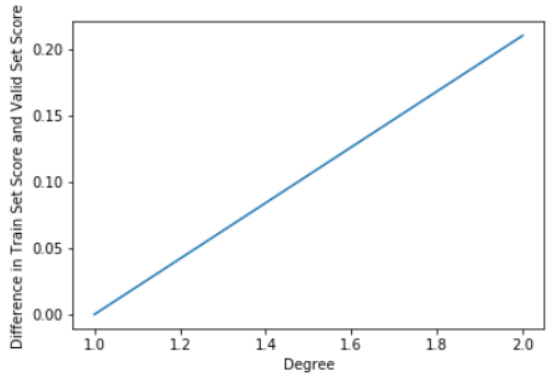
[2]: Scikit-learn, Machine Learning in Python https://scikit-learn.org/stable/

Appendices

Appendix A: *Correlations of features*



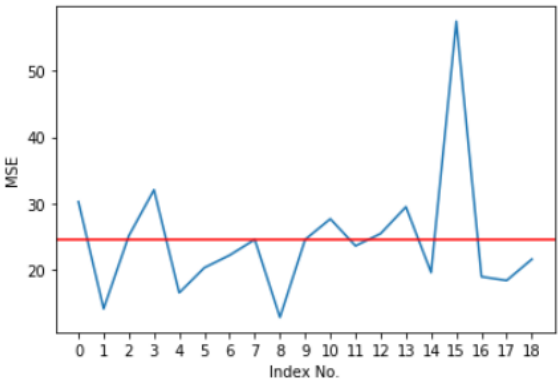
Appendix B: *Polynomial Degree = 1 (linear) is better than Degree = 2*



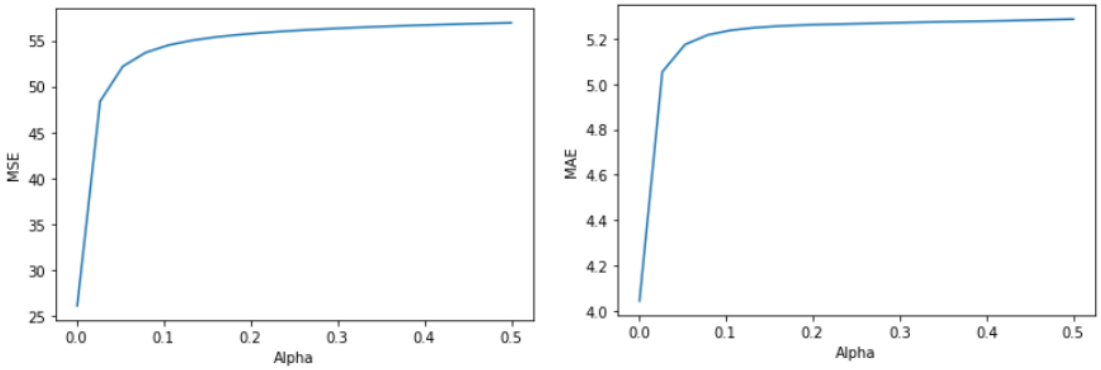
Appendix C: *Table of Average Metrics and Best model*

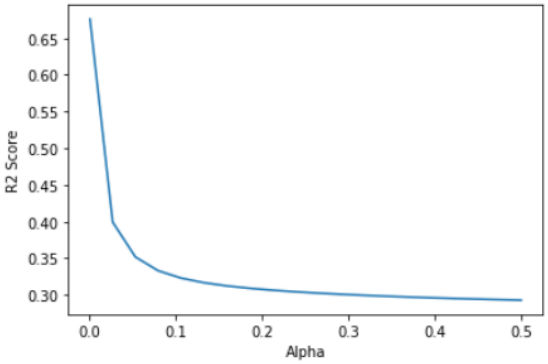
```
---Average Metrics---
Average MSE:  24.451506677396623
Average MAE:  3.7753635308829634
Average R Squared score:  0.6302241117410585

---Best Model based on average metrics---
Best model for MSE (index):  7
Best model for MAE (index):  6
Best model for R2 score (index):  11
```

Appendix D: *Finding the fold(index no.) closest to the average MSE(in red)*

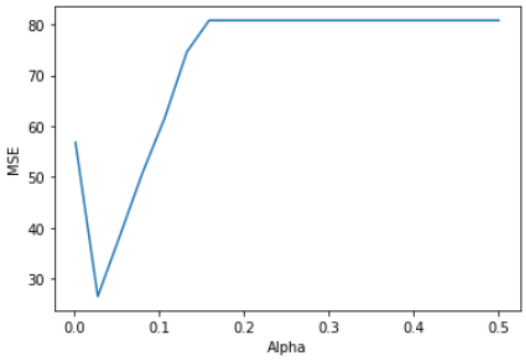


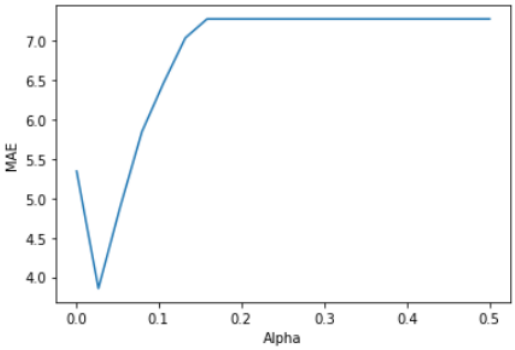
Appendix E: *Regression metrics when alpha changes (Ridge)*

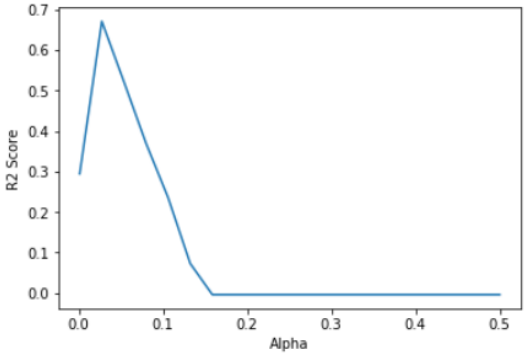




Appendix F: *Regression metrics when alpha changes (Lasso)*







Appendix G: *My solution .csv file*

	ID	Target_LifeExpectancy
1	1	63.52495811685518
2	2	63.46871758652489
3	3	63.401722714928155
4	4	62.782979267908104
5	5	62.28468594769004
6	6	61.690571112524665
7	7	60.366778790999774
8	8	61.24101046843912
9	9	60.98794108847178
10	10	59.67039537564178
11	11	59.79905801381603
12	12	58.6710443608522
13	13	58.39004887489475
14	14	58.98092390418412
15	15	65.41679159083715
16	16	64.71877596196684
17	17	79.97420258781932
18	18	80.9377409440335
19	19	79.79939792872585
20	20	79.74887926193702
21	21	79.61795635514336
22	22	79.53036971865353
23	23	79.38685009172616
24	24	79.34103091944013
25	25	79.27738128515486
26	26	79.47815778605288