A0216288X
Report on DSA4211 Project

Summary Page

Upon receiving the data, the first thing I did was to look for any avenues to clean the data. Firstly, there was a variable , X52, that had a significant number of NAs. To fix this, I removed the variable. I checked for collinearity and found 2 highly correlated variables, and I removed 1 of them.

Now that the data was preprocessed, there were several approaches to take. I first made two more datasets to use, by normalising the x train data , and standardising the x train data. Initially I only performed regularisation by putting the datasets into lasso, ridge, elastic net and lasso models. Thereafter, I realised that there were too many variables and thus realised subset selection was necessary. To perform subset selection, I performed recursive feature elimination. In round 1, after selecting 5 variables, I ran lasso , ridge, elastic net and lars on 3 cross validation folds and selected the one with the best cv score, lars. In Round 2, I changed the number of folds as I realised that my dataset was not equally divided when run on 3 folds. I picked 10 folds and ran lasso , ridge , elastic net and lars and selected the one with the best cv score, elastic net.

Before any models and observations could be drawn on the dataset, it was necessary to clean it. Firstly, I checked for rows with NA values and found that the variable X52 had a significant number of NAs, and so I removed it. Next, I checked for variance inflation factor, an indication of multicollinearity. Multicollinearity makes the dataset less reliable as it occurs when several independent variables in a model are correlated. In this dataset, two variables were highly correlated with a VIF of more than 10, X30 and X40. The feature X30 had a vif of 72.08217255849414 and the feature X40 had a vif of 73.22847982915701. I decided to drop feature X30. I checked the VIF after dropping X30, and no new features with high multicollinearity were detected so it was safe to say that there were no more variables with high multicollinearity.

Moving on, I created 4 primary models to use with varying L1 and L2 regularisation. I made models with the LarsCV, LassoCV, ElasticNetCV and RidgeCV from the sklearn.linear_mode package. Lasso and Lars perform l1 regularisation, Ridge performs l2 regularisation, and ElasticNet performs both l1 and l2 regularisation. With the respective CV functions, I was able to perform the models to select the best alpha on multiple cross validation folds in order to get an alpha which did not overfit the data.

With various models, it was important to set a basis of comparison for the best model. I decided to obtain the score and the mean of the cross validation score of the models, using the mean cv score to pick the best model and the score to ensure the model performed relatively well enough to be used.

Initially in round 1, I ran these models on 3 cross validation folds and got the best CV score from using LarsCV with normalised data. In Round 2, I realised my data was not equally divided on 3 folds ; 250/3 would result in 3 datasets of different sizes. Therefore I decided to use 10 cross validation folds instead so as to split the dataset into 10 equal sizes.

With 10 cv folds, I got the values as follows. As seen highlighted from the table below , Lasso CV on normalised data performed best.

		Normal Values	Normalised	Standardised
Lasso CV	Alpha	0.50856474148134	0.08718194389620	0.508564741481346
	Score	0.63552651181165	0.64377722479908	0.635526511811658
	CV Score	0.53230687269910	0.53702228759705	0.532306872699109
Ridge CV	Alpha	10.0	1.0	10.0
	Score	0.77321172130442	0.76495538654782	0.773211721304425
	CV Score	0.32822019478432	0.40808217903708	0.328220194784326
Elastic NetCV	Alpha	0.38294272105505	0.02004760997340	0.382942721055055
	Score	0.70298987492629	0.72353153889856	0.702989874926295
	CV Score	0.50772239083140	0.47526143298538	0.507722390831402
Lars CV	Alpha	0.03595208241364	0.03595208241364	0.035952082413646
	Score	0.62575704085816	0.62575704085816	0.625757040858166
	CV Score	0.53676720967874	0.53676720967874	0.536767209678747

I realised that there were too many features still and subset selection was required. Therefore, I did recursive feature elimination subset selection obtaining X12, X13, X14, X82 and X94. Recursive feature elimination is a feature selection algorithm that fits a model , in this case the ordinary least squares model, and recursively eliminates a small number of features per round based on R square scoring until the optimal number of features is obtained.

I then standardised and normalised the values. With a cv of 10, I ran and computed the various scores as shown below. As seen highlighted on the next page, the Elastic Net ran with normalised values had the highest CV Score of 0.5832717940234157.

		Normal Values	Normalised	Standardised
Lasso CV	Alpha	0.00508564741481	0.00093482340791	0.005085647414813
	Score	0.63987875865439	0.63987847795263	0.639878758654398
	CV Score	0.58192976117351	0.58193449368865	0.581929761173518
Ridge CV	Alpha	10.0	0.1	10.0
	Score	0.63912053862419	0.63978627205387	0.639120538624199
	CV Score	0.58213581470126	0.58265990596969	0.582135814701264
Elastic NetCV	Alpha	0.01777462657962	0.00174363887792	0.017774626579624
	Score	0.63980655767980	0.63939078314798	0.639806557679801
	CV Score	0.58317301317578	0.58327179402341	0.583173013175785
Lars CV	Alpha	0.0	0.0	0.0
	Score	0.63988083680799	0.63988083680799	0.639880836807992
	CV Score	0.58187974991242	0.58187974991242	0.581879749912429

Therefore , for Round 1 I selected LarsCV on 3 folds with the variables X14, X13, X12, X82 and X94,

And for Round 2 I selected Elastic Net CV on 10 folds with the variables X14, X13, X12, X82 and X94.