Machine Learning Report 2

(Data set used: # id:21-21-21)

(i) (a)

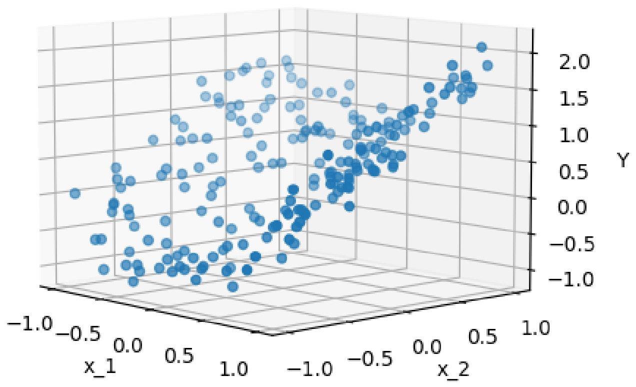

Figure 1

The training data in **Figure 1** looks like it fits on a curve.

(i)(b)

C Value vs features

	1	10	25	100	125	250	500	625	1000
1	0	0	0	0	0	0	0	0	
x_1	0	0	0	0	0	0	-.01370524	-.01782308	-.03303159
x_2	0	0.89613782	0.98017811	1.02132335	1.02318503	1.02732304	1.04412517	1.05884158	1.08320075
x_1^2	0	0.57957654	0.89054241	1.0453247	1.05459757	1.07217481	1.07978309	1.08153468	1.0845079
x_1x_2	0	0	0	0	0	0.00604762	0.01666095	0.01872079	.02164245
x_2^2	0	0	0	0	0	0	0	0	0
x_1^3	0	0	0	0	0	0	0	0	0
$x_1^2x_2$	0	0	0	0	0	0	-.01542768	-0.0280051	-.04179604
$x_1x_2^2$	0	0	0	0.01109124	0.0224322	0.04303377	0.07919888	0.09114382	0.11119726
x_2^3	0	0	0	0	0	0	-0.0158045	-.03226912	-.06022501
x_1^4	0	0	0	0	0	0	0	0	0
$x_1^3x_2$	0	0	0	0	0	0	0	0	0
$x_1^2x_2^2$	0	0	0	0	0	0	0	0	0
$x_1x_2^3$	0	0	0	0	0	0	0	0	-.04772305
x_2^4	0	0	0	0	-.00517646	-.02887712	-.04151264	-.04331075	0.01792853
x_1^5	0	0	0	0	0	0	0	0	-.00872407
$x_1^4x_2$	0	0	0	0	0	0	0	0	0
$x_1^3x_2^2$	0	0	0	0	0	0	0	0	0
$x_1^2x_2^3$	0	0	0	0	0	0	0	0	0
$x_1x_2^4$	0	0	0	0	0	0	0	0	0
x_2^5	0	0	0	0	0	0	0	0	0

Figure 2

Figure 2 shows a table of the weights associated with each engineered feature and its corresponding hyperparameter C. As you can see, as C increases, less and less of the weights are 0. The features x_2 and x_1^2 have the highest influence on the output because they are the only weights that have a value over 1. Firstly, the values in the training data mostly have a value less than 1 so features with increasing exponents will have less and less influence on the output. Secondly, a high C value reduces the influence the L_1 penalty has on the model. Third, the higher hyperparameters have exposed some features with higher exponents to have a weight albeit extremely small. These weights must have a small influence over what the model outputs. This indicates to me that the use of additional features may not be necessary and a simpler model might produce similar results. Maybe using a different method of producing our model might justify the usage of features with higher polynomials.

(i)(c)

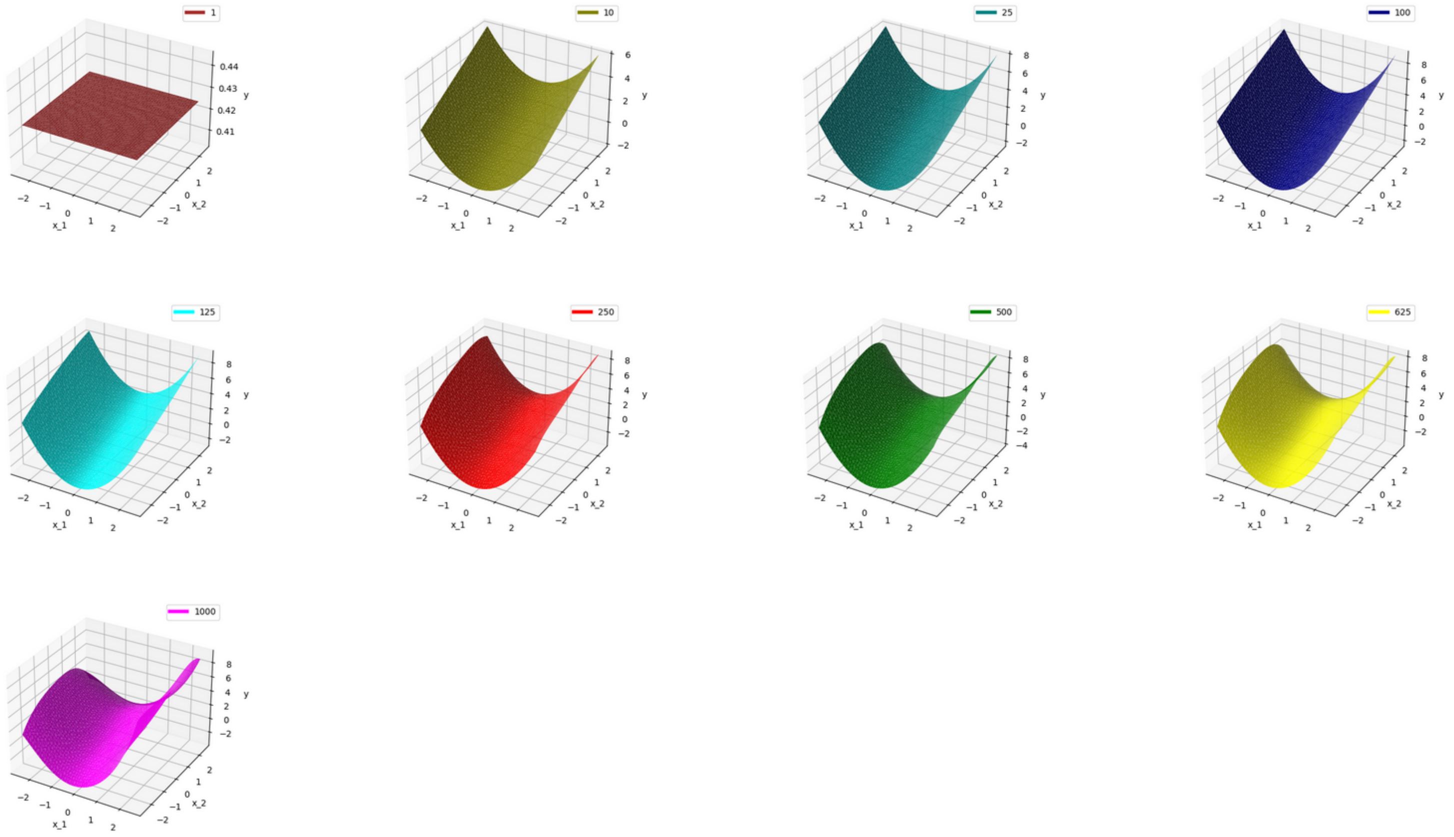

Figure 3

As C increases the plot looks more and more similar to the curve shown in **Figure 1** but after a certain point, the curve takes on a “horseshoe” shape which indicates that the model is fitting in the “noise” of the data.

(i)(d)

Underfitting is when there are too few parameters in the model so it fails to capture the trend of the data. Undefit models perform well on training data but poorly on testing data. With regards to **Figure 2** and **Figure 3**, we can see if the hyperparameter C is too low, it fails to capture the trend of the data which is shown visually as a flat plane and expressed numerically as all the weights set to zero.

Overfitting is when there are too many parameters in the model which is caused by training the model too well. When overfitting occurs, the noise of the training data is incorporated into the model. Overfit models perform well on testing data but poorly on new data. With regards to **Figure 2** and **Figure 3**, we can see if the hyperparameter C is too high, it over captures the trend of the data which is shown visually as a horseshoe despite the data taking on the the shape of a curved plane.

(i)(e)

	1	25	100	250	625	1000
1	0	0	0	0	0	0
x_1	-0.03302909977	-0.07656136407	-0.09454478552	-0.09945197304	-0.1015789118	-0.1021264534
x_2	1.046476523	1.299517735	1.345948691	1.35739176	1.362225939	1.36345893
x_1^2	0.8747034844	1.208440925	1.238908519	1.245804747	1.24865627	1.249377944
x_1x_2	0.05839186991	0.01859195676	0.008731309129	0.006365851917	0.005374266922	0.00512208597
x_2^2	0.002492798493	0.1260062285	0.1376190455	0.1402200447	0.1412928722	0.141564151
x_1^3	-0.03362700444	-0.060117596	-0.03327540614	-0.02500985496	-0.02132933387	-0.02037286356
$x_1^2x_2$	0.005918951434	-0.315796138	-0.4034343053	-0.4259756859	-0.4356021645	-0.4380671234
$x_1x_2^2$	0.1127098862	0.5240488012	0.628027575	0.6543709978	0.6655769898	0.6684422891
x_2^3	-0.02148940466	-0.7404797747	-0.8851452618	-0.9206397165	-0.9356128549	-0.9394297503
x_1^4	0.1923742174	-0.09572750936	-0.1230962702	-0.1293227543	-0.1319008486	-0.132553649
$x_1^3x_2$	-0.03476079683	0.02054082695	0.03176690204	0.03447638691	0.03561477735	0.03590455327
$x_1^2x_2^2$	0.06747887726	-0.07515657317	-0.08708484505	-0.08969504615	-0.09076413019	-0.09103374072
$x_1x_2^3$	-0.02285023096	-0.01521250373	-0.01325668066	-0.01287641689	-0.0127275601	-0.01269070232
x_2^4	-0.08319549369	-0.147161262	-0.1517558163	-0.1527209426	-0.1531124144	-0.1532108006
x_1^5	0.05216976641	0.1177200011	0.1084572719	0.1049663647	0.1033524176	0.1029276684
$x_1^4x_2$	-0.06452237348	0.05210725685	0.1055625516	0.1196917963	0.1257647226	0.1273233154
$x_1^3x_2^2$	0.03916296124	-0.1953187124	-0.2733971335	-0.2936218744	-0.3022719514	-0.3044880372
$x_1^2x_2^3$	0.01173305663	0.3958410075	0.4820935758	0.503785734	0.5129977548	0.5153518237
$x_1x_2^4$	-0.03185513158	-0.3494874869	-0.4230978176	-0.44153214	-0.4493507549	-0.4513477999
x_2^5	8.63E-05	4.65E-01	5.67E-01	5.91E-01	6.02E-01	6.04E-01

Figure 4

(**Note:** The range of C values used in **Figure 4** is the same in **Figure 2** but only a select set of values were shown for convenience.)

Figure 4 shows a table of the weights associated with each engineered feature and its corresponding hyperparameter C. Just like in Figure 2, x_2 and x_1^2 have the largest influence over the output because they are the only weights that are greater than one.

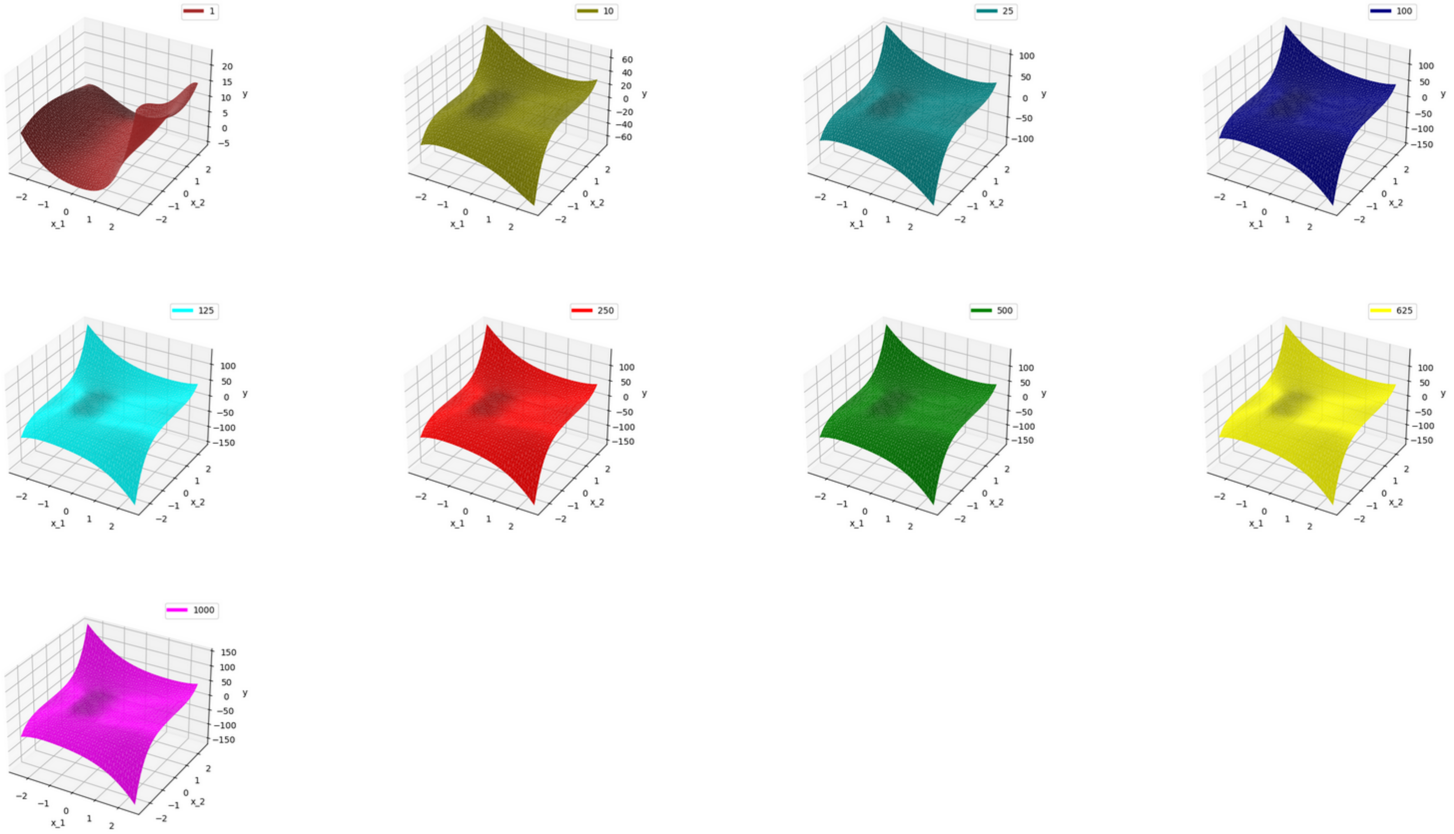

Figure 5

The model best represents the data when its L_2 penalty is at its maximum i.e when the hyperparameter value C is 1. As C increases, the model doesn’t seem to change.

The impact of changing C values on the weights has a greater effect on the Lasso model as opposed to the Ridge Regression model because the weights remain relatively constant as C increases for the Ridge Regression model but not for the Lasso model. This can also be seen visually for the different values of C in **Figure 3** and **Figure 5**.

(ii) (a)

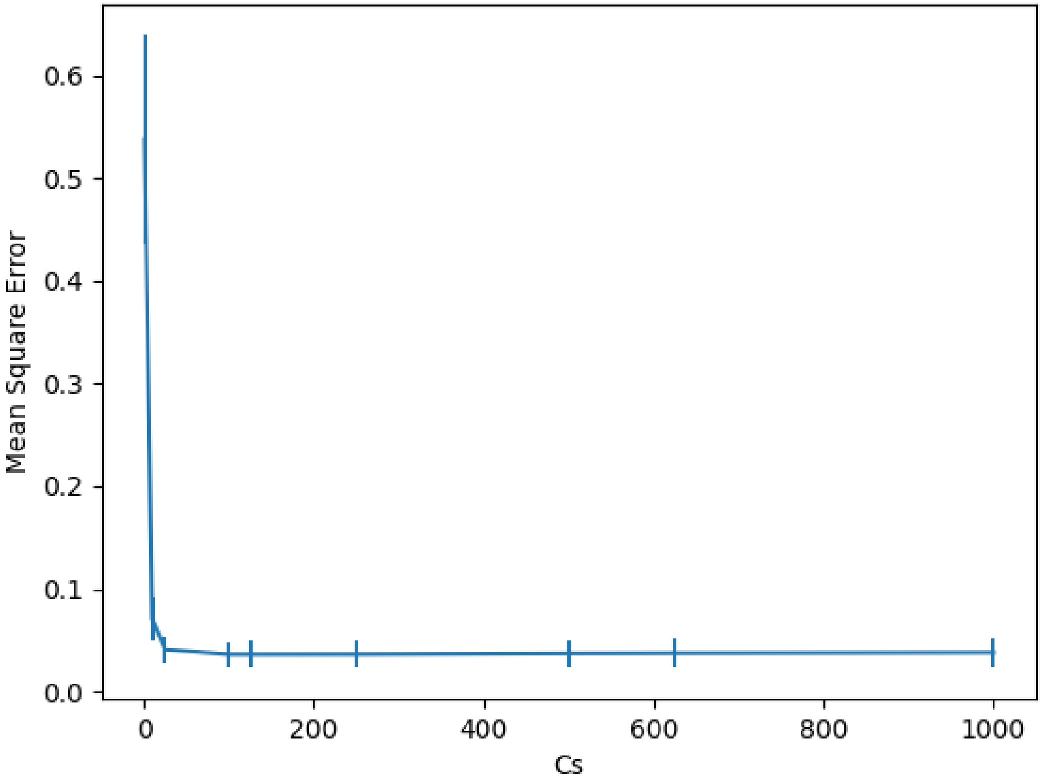

Figure 6

The range of values used are [1, 10, 25, 100, 125, 250, 500, 625, 1000]. The extremes of the range (1 and 1000) are chosen to have a maximum and minimum effect of the L_2 penalty. The values within the range are factors of 5 and 10 with some of the values removed to avoid unnecessary computation because some of the values were close to each other and would therefore produce similar results.

(ii) (b)

According to the graph, a C value of 100 would be ideal because it has the lowest mean square error. Any value after that doesn’t reduce the mean square error so it’s just overfitting the model.

(ii)(c)

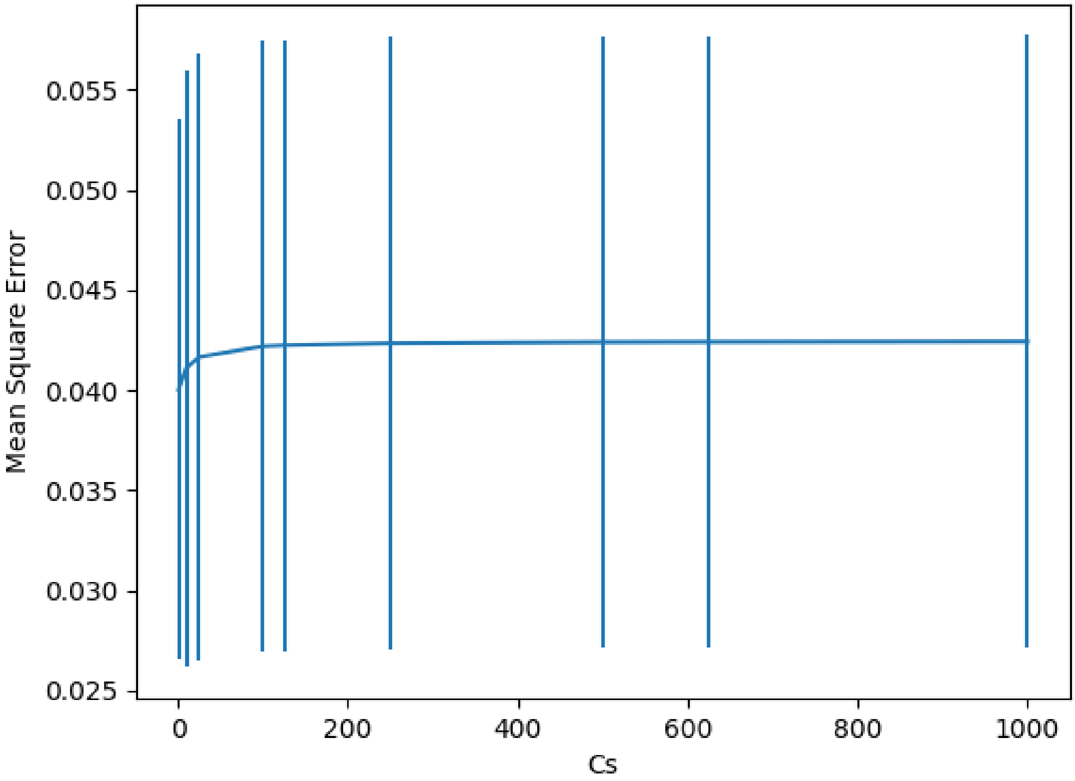

Figure 7

The range of values used in **Figure 7** are the same in **Figure 6** and use the same rationale.

According to **Figure 7**, the value for C that should be used is 1 because it has the lowest mean square error.