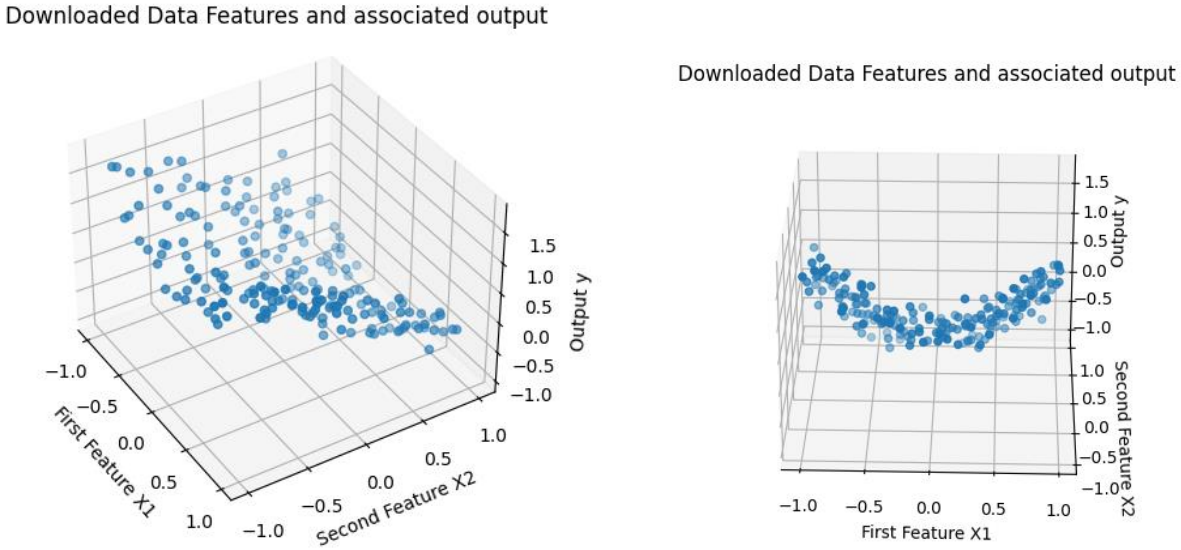
Machine Learning Assignment 3

Alex Kennedy 17328638

(i)(a)

Here is the downloaded data plotted on a 3d scatter plot (two different perspectives).



As one can see, the training data lies on a curve and not a plane. This can be seen by the varying opacity of the points

(i)(b)

The PolynomialFeatures adds additional features by generating all combinations of powers up to the desired degree of 5. After obtaining the polynomial features, I performed Lasso Regression on the data using sklearn with different values of C: 1, 10 and 1000. Here are the parameter values of the model given each C value:

```
lasso regression with C = 1
coefficients:  [ 0. -0. -0.  0.  0. -0. -0. -0. -0. -0.  0. -0.  0.  0. -0. -0. -0. -0.
 -0. -0. -0.]
 intercept:  0.3367133954133785
```

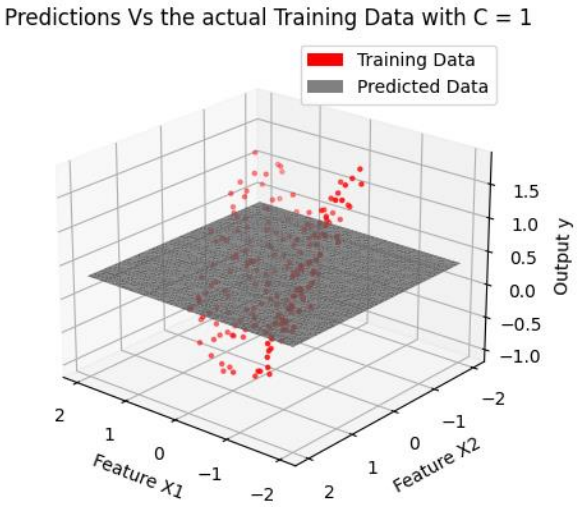
```
lasso regression with C = 10
coefficients:  [ 0.         -0.         -0.80165082  0.37850565  0.         -0.
 -0.         -0.         -0.         -0.          0.         -0.
  0.          0.         -0.         -0.         -0.         -0.
 -0.         -0.         -0.        ]
 intercept:  0.22645295320753336
```

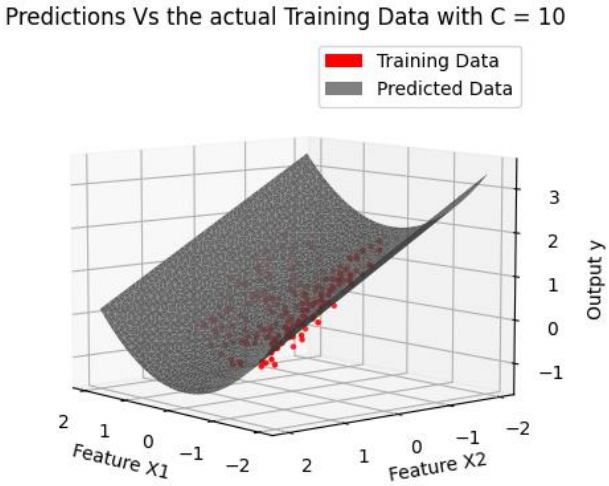
```
lasso regression with C = 1000
coefficients:  [ 0.         -0.         -1.09651715  1.11048619  0.23280451  0.
  0.         -0.00826834 -0.07953245  0.12894172 -0.19253617 -0.25819003
  0.17061248 -0.08032482 -0.17315949  0.03159552  0.         -0.11110935
  0.          0.058711    0.05783482]
 intercept:  0.03040522911890775
```

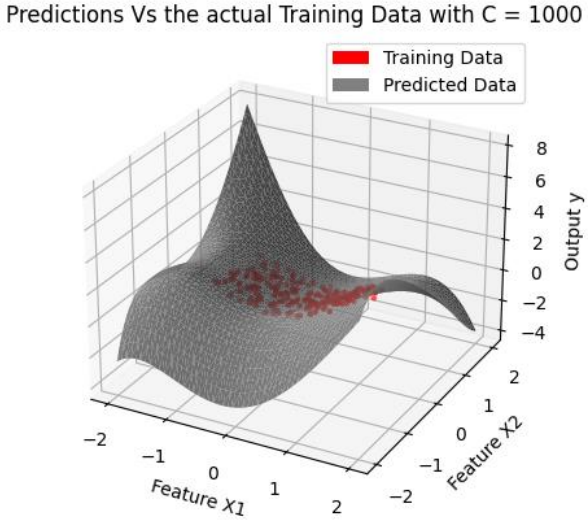
From the lectures we see that the Value of C is used to calculate alpha $\alpha = 1/(2C)$ and so the larger the C value, the smaller the value of alpha.

As seen from the reported parameter values, when C is a low value, all of the parameter values become 0. As the value of C increases, the number of parameter values = 0 decreases. However, there are still quite a few 0's even at C = 1000. This is because Lasso Regression which uses the L1 penalty which tends to make the less important feature values 0 which in turn, removes these features altogether.

(i)(c)
After creating the grid of predictions using the code given in the assignment, in order to compare it with the actual data, we need to get the feature polynomials also. The grid range is between -2 and 2. Using sklearn's predict() function, I created a list of predicted values. I used a tri-surface plot to display the predicted values and a scatter plot for the actual training data. Here are the plots when using different values of C:







From the graphs, C = 1 is a bad value to use. The predictions clearly don't fit around the training data and just produces a flat plane (underfitting). This is due to all the parameter values being 0, so the slope is also 0. When C = 10, the predictions do fit around the points quite well i.e the curvature of the predictions is the same as the training data. The curve is also quite uniform as there are still a lot of values of 0 for the parameter values. When C = 1000, the center of the surface plot does fit around the points. However, many of the points towards the outer ends of the curve are quite skewed (overfitting). It seems that there is a threshold where the C value will work the best for this specific set of training data.

(i)(d)

Underfitting: Underfitting can be referred to a model which cannot model the training data. A model which is underfit will not be effective as it will not have good performance on the training data

Overfitting: Occurs when a model over-learns the detail and noise of the training data to such an extent to where it negatively impacts the performance of the model.

The 3 C values selected illustrate these concepts quite well. A low C value (C = 1) will produce underfitting on the model, this can be seen from the graph that the predicted data is simply a poor representation of the training data. A high C value (C = 1000) will cause overfitting. Although the predicted model fits the training data, it also now has random fluctuations and noise which will hinder performance. A mid-range C value (C = 10) for the predicted model produces a good fit on the training data as there is no random fluctuations and the model fits the training data well.

(i)(e)

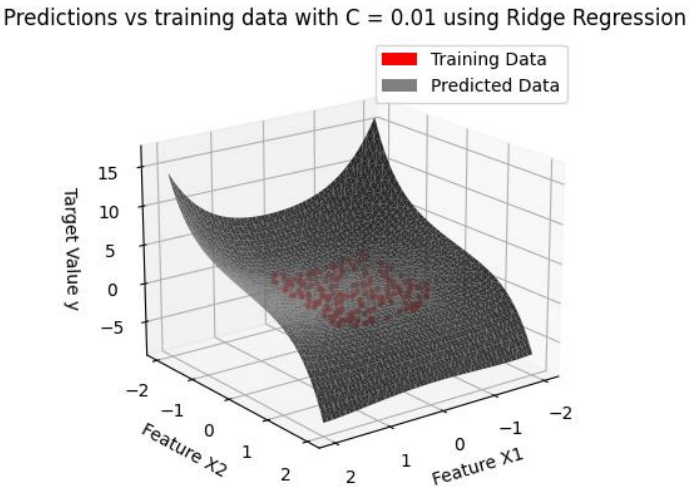
Here we are repeating (b) and (c) but using Ridge Regression, which uses L2 penalty to the cost function.

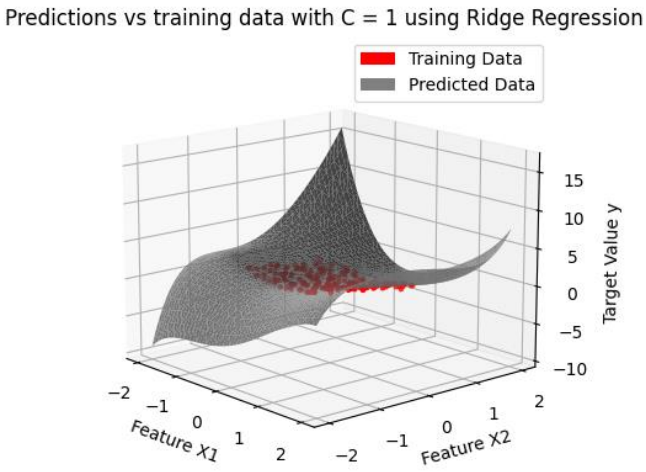
```
Ridge regression with C = 0.01
coefficients:  [ 0.         -0.00129681 -0.40004919  0.19347421  0.01107024 -0.02111683
  0.00426135 -0.12092208 -0.01674764 -0.19731087  0.16739154 -0.00315297
  0.04554318  0.00913901 -0.02312038  0.00303802 -0.05942786 -0.01019052
 -0.0582295  -0.00953488 -0.12704294]
 intercept:  0.2572114616770147
```

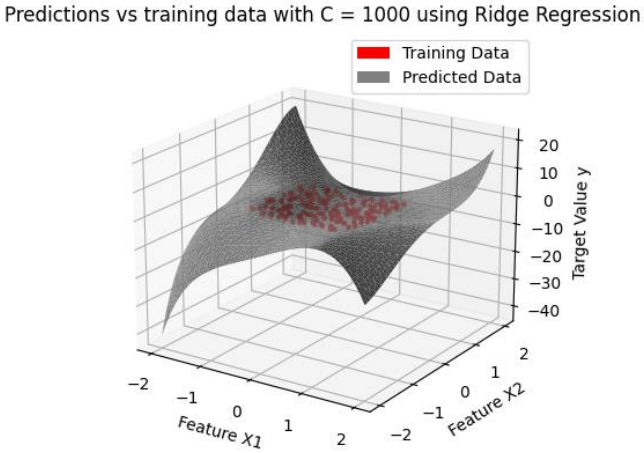
```
Ridge regression with C = 1
coefficients:  [ 0.         -0.01226374 -1.00740094  0.90412239  0.23379548  0.03418444
  0.03482589 -0.22316926 -0.14819136 -0.00930229 -0.00276957 -0.22729904
  0.22442505 -0.11181041 -0.22230471  0.0339685   0.14571049 -0.15798893
  0.09612788  0.1666565   0.12485039]
 intercept:  0.056660049822384206
```

```
Ridge regression with C = 1000
coefficients:  [ 0.00000000e+00 -5.49208336e-03 -1.03100720e+00  1.29665282e+00
  3.89486850e-01  2.05007012e-01  4.01806578e-02 -6.88637256e-01
 -3.95725519e-01  1.36981581e-01 -4.00192499e-01 -3.46410455e-01
  1.64737967e-01 -2.25583522e-01 -4.09009793e-01  8.66354563e-02
  5.53389338e-01 -3.61225006e-01  3.55971411e-01  5.71112432e-01
  8.57322641e-04]
 intercept:  -0.007388558099871101
```

Here we can see that very few of the parameter values here are 0, even for C = 0.01. This suggests that we would need an extremely large alpha value in order to produce underfitting in our model. With C = 1 and C = 1000, the parameter values are much larger than our parameter values using Lasso Regression. This could be due to L2 penalty adding a squared magnitude of the coefficients to the cost function.



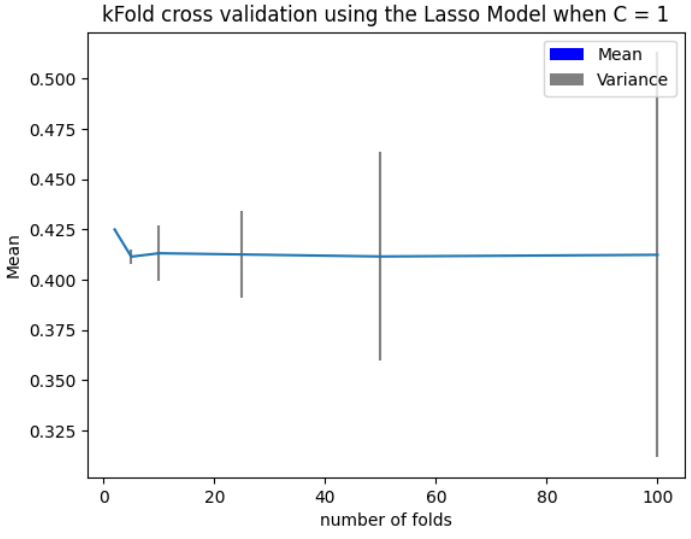




One can see that no underfitting has occurred here. Even with a very small C value, the predicted fits the training well as well as having a good shape. With C = 1 and C = 1000, there is a bit of fluctuation in the predicted models, which could suggest some overfitting. However, the model of C = 1000 here compared to Lasso's model when C = 1000 is much better. There is a lot less fluctuation in the model.

(ii)(a)

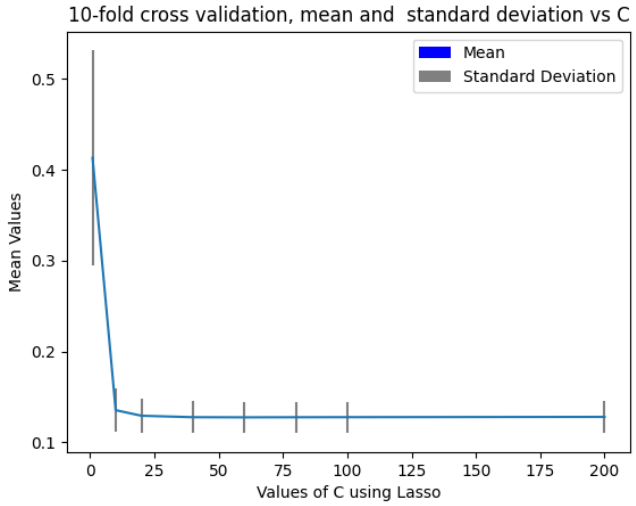
The mean and the variance obtained from the estimates from performing k-fold cross validation was found by calculating the mean and variance of the mean squared error values obtained for every value of k. Here is the plot which shows the values of the mean and the variance as error bars vs the number of folds.



There is a trade-off in the number of folds. The more folds used, the more accurate the mean will be. This can be seen when there is a small number of folds, the mean changes quite drastically. However, it seems that the more folds you use, the higher the variance becomes. This could be due to the higher likelihood of outliers appearing in higher fold numbers. From the above graph, I would pick 10 folds to be ideal, as here is when the mean begins to level out and the benefits of having higher folds begins to decline and the variance begins to increase very rapidly, becoming unreliable.

(ii)(b)

Here is a plot of 10-fold cross validation of the mean and standard deviation of the prediction error vs different values:



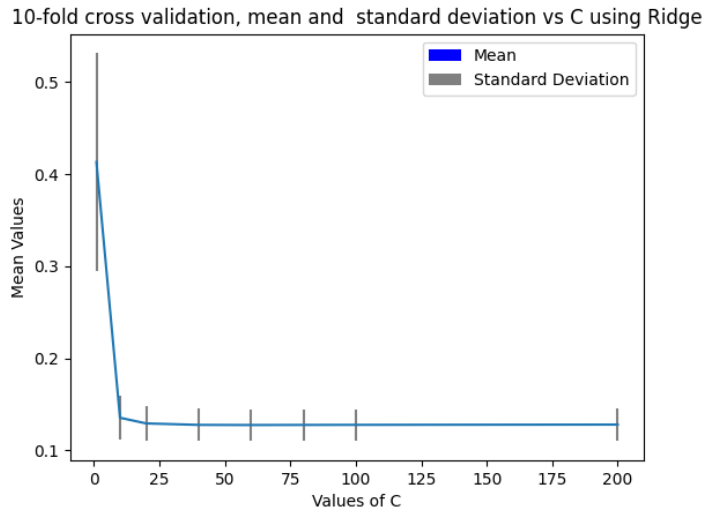
I chose the values of C to be (1, 10, 20, 40, 60, 80, 100, 200) as I believe it gives a good indication of a suitable C value to pick for 10-fold cross validation. It will show the effects of a small and large C value.

(i)(c)

A low C value will produce an inconsistent mean with a very large standard deviation. As the value of C increases, the mean and the standard deviation both seem to level out but once again it seems there is diminishing returns to this once the C value starts to get higher. The value of C I would pick here would be between 30 and 60 as here is when the mean and standard deviation barely differ from higher C values. From the graph C = 30 and C = 200 are essentially the same.

(d)

Here I will repeat what I have done in (b) and (c) but use a Ridge Regression Model.



I chose the same values for C as once again I believe it gives a good indication of a suitable value to pick for C for 10-fold.

The model looks very similar to the Lasso Regression model. Because of this, I would pick a C value between 30 and 60.