Project 3: Ridge Regression

Christian Romeo – Oscar Aleksander Halvorsen

Ridge regression function:

The implementation of the ridge regression function was closely based on the implementation of linear regression that was shown in the course. To implement ridge regression, we created a new function named *ridge_regression.*

Ridge_regression takes in three input values:

- *X:* 3D array of the x values that have already been polynomially feature expand
- *y:* 2D array of the mean centered y values
- *lambda_val:* the regularization parameter

Then we apply the standard formula for ridge regression:

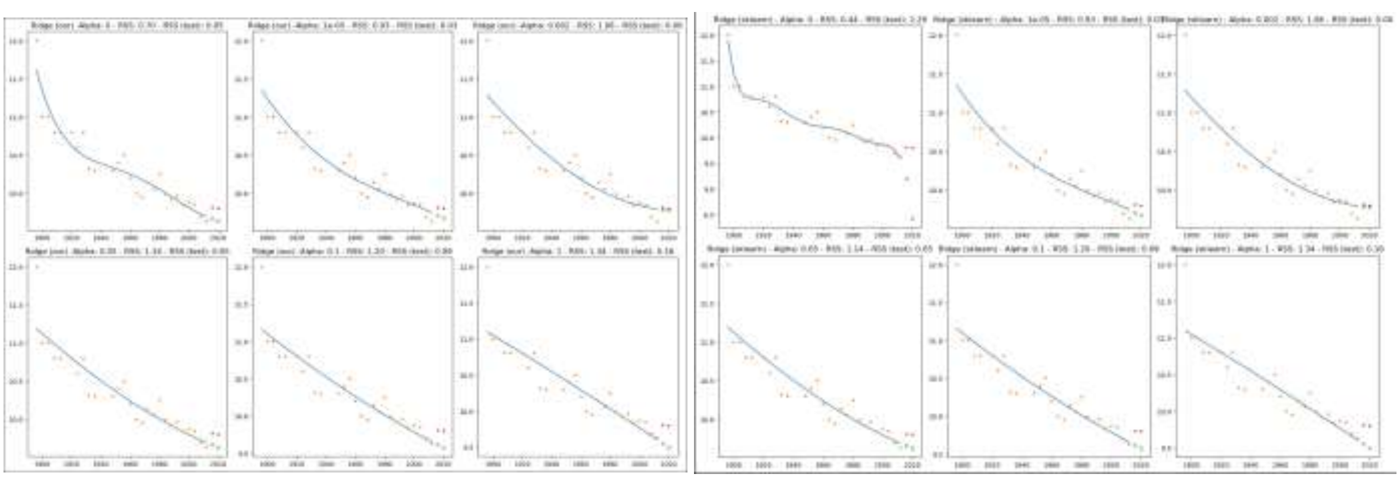
$$w = (X^T X + \lambda I)^{-1} X^T y$$

Where $X = X, y = y, \lambda = lambda\_val$ and I is the identity matrix with the same dimensions as $X[1]$. We also remember to set $I[0][0] = 0$, because all the values in the first column are bias terms and the regularization should not be applied to them. The function then output w which estimated coefficients of the linear model.

Interpolation:

We then test *ridge_regression* on the olypmics_100m data and chose the last two data points to best the test data and the rest to be train data. We used the same six different alpha values as the demo notebook: 0, 0.00001, 0.002, 0.05, 0.1 and 1 and tried to find the one with the RSS, both for the training and testing values. We got the best RSS (testing) result for $alpha = 0.002$, but $alpha = 0.00001$ and $alpha = 0.05$, gave quite good results.

We then compared it to the results we obtained when using *Ridge* from sklearn. *Ridge_regression* which produced similar plots, indicating that our function is performing effectively. RSS and RSS (test) are even completely equal for all the alpha values except $alpha = 0$, where it looks like the sklearn one has a smoother curve. This could be due to differences in the implementation of the algorithm.



Graphs from ridge_regression (left) and and sklearn (right). As one can see, the two are only different when alpha=0.