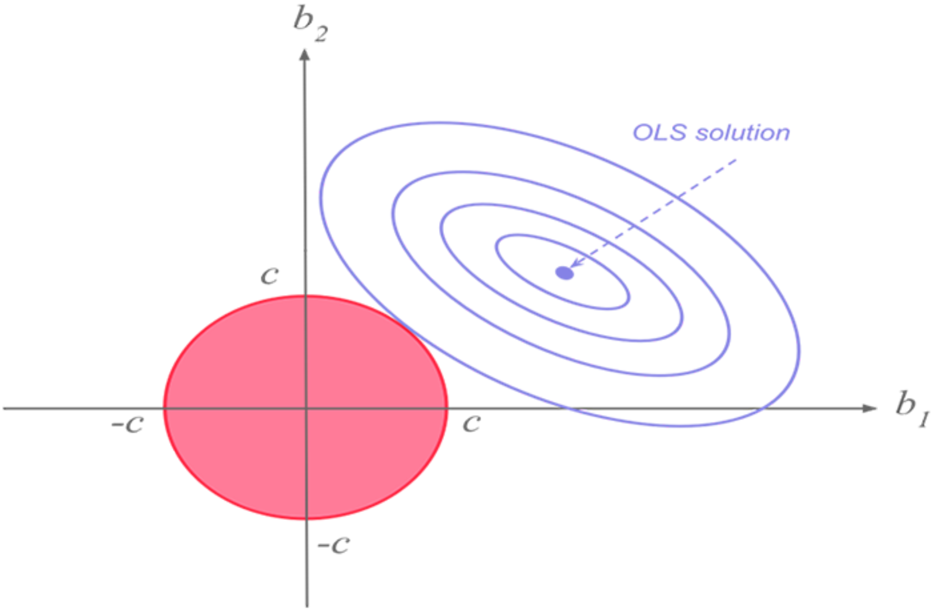
Case 1

02582 - Computational Data Analysis



GROUP

Koumaniotis, Ioannis - s212887
Kapakoglou, Georgios - s223001

March 18, 2023

Contents

1 Introduction/Data description

In this project, we are provided with a dataset consisting of 100 observations of response variable Y and a 100-dimensional feature matrix X. Additionally, we have 1000 new observations. Our goal is to test different model and find the optimal one in order to construct a predictive model for Y using the given data. The given data are available in two text files, case1Data.txt and case1Data Xnew.txt, in which we got access through the course page on DTU learn.

2 Model and method

In order to find the predictions of the new data set, four models were trained on the first dataset. The first model (baseline model) is the simple linear regression, the second model is the lasso regression, the third represents the ridge regression and the fourth model is the elastic net. The last three models are used to control the model complexity.

Regarding the first model, the weights β of the model can be calculated from the next equation:

$$\beta_{\text{OLS}} = \arg\min_{\beta} \left\{ ||Y - X\beta||_2^2 \right\} \tag{1}$$

The weights for the second model are calculated from Equation 2. The equation is the same as for the Ordinary Least Square apart from the last term $\lambda||\beta||_1$. The λ factor introduces a penalty to the weights and makes the solution non-differentiable at 0.

$$\beta_{\text{Lasso}} = \arg\min_{\beta} \left\{ ||Y - X\beta||_2^2 + \lambda||\beta||_1 \right\} \tag{2}$$

The third model is the Ridge regression model. Compared to the previous model, the ridge regression introduces a quadratic penalty to the weights and the latter are calculated as seen below.

$$\beta_{\text{Ridge}} = \arg\min_{\beta} \left\{ ||Y - X\beta||_2^2 + \lambda||\beta||_2^2 \right\} \tag{3}$$

Finally, concerning the last model, it combines L1 and L2 penalties to achieve both sparsity and smoothness in the learned coefficients. Hence, it minimizes the objective function as seen below:

$$\frac{1}{2n} \left\{ ||Y - X\beta||_2^2 \right\} + \lambda\Big(\alpha||\beta||_1 + 0.5(1-\alpha) * ||\beta||_2^2\Big) \tag{4}$$

In all cases, the λ factor is determined from cross-validation as described thoroughly in the subsequent chapters.

In an attempt to decrease the root mean squared error (RMSE) of our predictive model, a linear regression with feature selection was introduced using both forward and backward selection techniques. For sake of knowledge, feature selection is the process of selecting a subset of features from a larger set of available features to be used in a predictive model. The aim of feature selection is to improve the model's performance by reducing overfitting. However, rather than decreasing the

RMSE, we observed that it increased by a factor of 10 after applying these methods. This mainly happen due to the fact that the selected features do not capture all the relevant information needed to make accurate predictions.Thus we moved on without any dimensionality reduction method.

3 Model selection

In order to choose the best model, we have performed two rounds of 20-fold cross-validation to choose the optimal hyperparameter lambda and to select the best model among a baseline, a lasso, a ridge and an Elastic Net.
The first round of 20-fold cross-validation involved tuning the λ hyperparameter for each model, which is used to control the strength of the regularization penalty applied to the coefficients. The λ values ranged between 10^{-9} and 10^{9}.
In the second round of 20-fold cross-validation, we evaluated the performance of the models with the chosen hyperparameter lambda. The best model is chosen based on its performance metric, which in our case is the MSE.

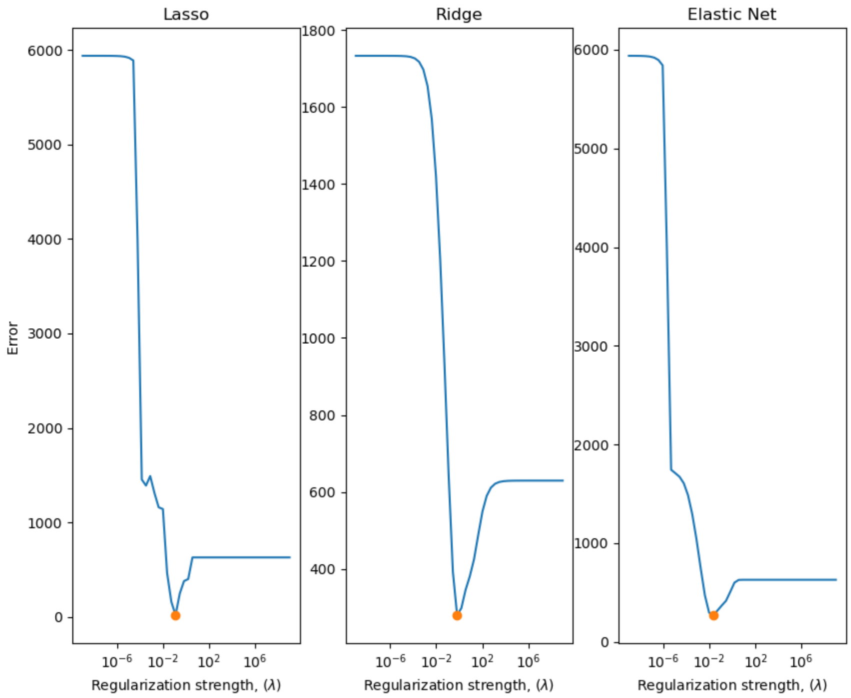


Figure 1: Optimal Lambdas

The above plot presents the errors for different Regularization strength (λ) for Lasso, Ridge and Elastic-Net regression. From the plot, it appears that the Lasso method has a sharper decrease in MSE as lambda increases in comparison with the other two models, as we can see that lasso drops the error at 0. On the other hand, the ridge and Elastic-Net methods have a smoother decrease in MSE as lambda increases. Moreover, one can be seen is that on the Ridge plot, the graph is extremely more soft in compare with the one represents the lasso and the Elastic-Net. This is happening because the Lasso regression model is multiplying the β by λ while in Ridge, β is multiplied with λ^2 which will provide softer curves as they are raised in the power of 2. Additionally, the Elastic-Net method contains both β and β^2, multiplied by & λ. The optimal values of the regularization λ are indicated on the next table.

Model	Lasso	Ridge	Elastic Net
Optimal λ	0.120	0.655	0.022

Table 1: Optimal values of λ

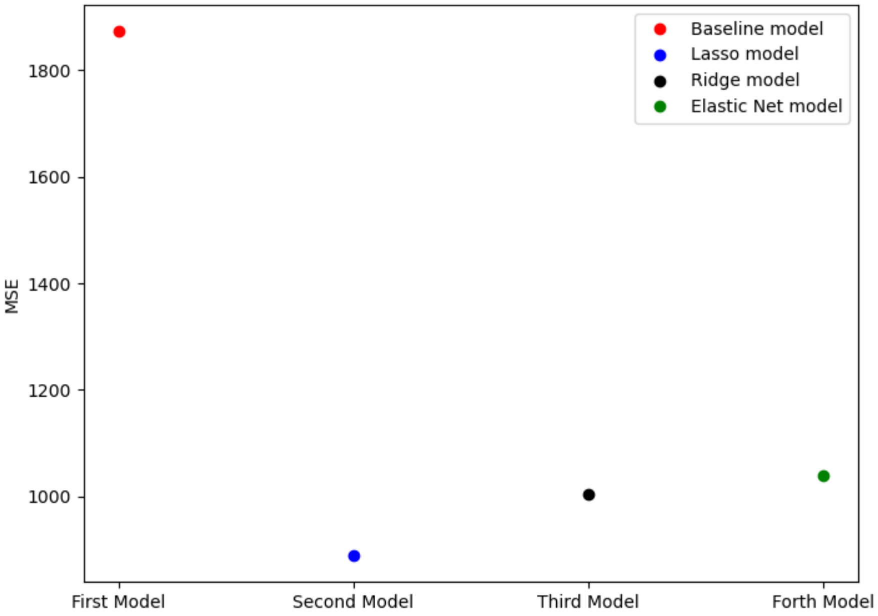


Figure 2: Comparison of four models' performance

The plot in figure 2 shows the mean squared error (MSE) performance of four different models: a baseline model, a lasso model,a ridge model and an Elastic-Net model. The y-axis represents the MSE value, which ranges from 800 to 2000. The baseline model appears to have the highest MSE value, located in the top left corner of the plot. This suggests that the baseline model is the least accurate of the four models. Also, the ridge and Elastic-Net model appear on the right side of the plot with an MSE value of around 1000, indicating that they are more accurate than the baseline model. Finally, The lasso model appears second in the order of the plot, with the lowest value of MSE close to 800. This indicates that the lasso model is most accurate model.

Overall, this plot provides a clear visualization of the MSE performance of the four models, allowing to easily compare their relative accuracy. We used this plot to demonstrate the effectiveness of the regularization methods for reducing MSE compared to a baseline model.

4 Missing values

After an exploratory analysis of the dataset, we observed that the first 95 columns contained numeric values, while the last 5 columns contained categorical values. Also, we observed a plethora of undefined (NaN) values. Regarding the missing values of our x_train data, we didn't fill the ”NaNs” through the cleansing process, as we will do it later inside the cross validation method while taking the mean value of the column and assign it in the undefined value. It is vital to fullfill the empty data inside the cross validation loop in order to prevent data linkage and to give more accurate value to the missing values. Lastly, we extracted the ”Y” column as target variable and the rest columns as ”X” for further analysis.

5 Factor handling

As stated previously, the categorical data on the last 5 coloumns, couldn't be used them for any regression model. To address missing values in the categorical columns, we iterated through each categorical column and replaced any ”NaN” values with the most common categorical value. In order for our model to interpret the data, we used one-hot encoding. In regression models when including dumming variables for a categorical variables, is recommended to only include k-1 of the dummies and not all k of them. The reason for this is that these k dummy variables sum to 1 leading to multicollinearity (linear dependence), which will make it difficult to interpret the coefficients. This was done to represent each category as a separate column rather than encoding them as a single column with numerical values. However, we encountered a challenge when we realized that a significant portion (20%) of our categorical data was missing.

To address this, we decided to use KNN imputation instead of the typical mode imputation to fillin the missing values. We chose to use KNN imputation with 4 as the optimal number of neighbors instead of the mode because KNN imputation takes into account the relationships between variables and uses this information to make better predictions for the missing values. Mode imputation, on the other hand, only fills in the missing values with the most frequent value of the column, which can result in a loss of information and potentially biased results.
Overall, these techniques enabled us to create more robust and accurate models for our data analysis.

6 Model validation

As described previously, the lasso regression model performs better in the train data and represents the model which ends up with the minimum error. Thus, the weights that have been calculated from Equation 2 are used to predict the new unseen data. The latter are the data provided from the second dataset with 1000 observations and 100 features. Each weight is multiplied by the feature together with the optimal value of the regularization parameter. This gives us a set of predictions for the target variable on the new dataset, which we could then evaluate using a suitable metric, such as the mean squared error.

7 Results

Since we don't know the true values of the new dataset, we assumed that the new dataset shares similar characteristics and is drawn from the same underlying distribution as the original dataset. Therefore, the root mean square error is estimated based on the root mean square error from the train dataset. The value of RMSE is 2.933 representing approximately 1.19% of the total range of the true values [-116, 130], which can be considered as a relatively small error and therefore a good prediction model.