Housing prices

Statistical methods for machine learning's project

Aggiudicato Irene - 960261

May 2021

Contents

1 Abstract

The aim of this paper is to predict the median value of houses in California. In order to do that, we are going to perform Ridge regression with square loss, one of the best way to increase the bias in linear regression making the model more stable. For proving the predictors, we are going to perform a cross validation on our dataset, splitting it in K-folds, for checking the mean square error. We are going to see how the result actually changes taking into consideration 100 different values of α Once done this, we are going to perform principal components analysis (PCA), a technique which allows us to project data onto a lower-dimensional space without loosing too much information. In the end we are going to make a brief paragraph about the conclusions that we can get analysing the work that has been done.

2 Dataset

The dataset is composed by 20640 observations that we can find in the rows and of 10 features which constitutes the columns of the data. The attributes are the following: - Longitude: A measure of how far west a house is; a higher value corresponds to farther west. The mean of this variable appears to be negative make us thinking that the major part of the houses are on the east. This is something confirmed by the minimum value which is equal to -114.31. - Latitude: A measure of how far north a house is; a higher value corresponds to farther north. In this case we have a mean equal to 35 and a range between 32 and 41. - Housing Median Age: Median age of a house within a block; a lower number corresponds to a newer building. - Total rooms: total number of rooms within an house. - Total bedrooms: total number of bedrooms within a block. This variable presents some missing values. In fact, out of 20640 observations we have 207 missing values. During the computation we will see how to deal with this issue. - Population: Total number of people residing within a block - Households: Total number of households, a group of people residing within a home unit, for a block. - Median income: Median income for households within a block of houses, measured in tens of thousands of US Dollars. - Median House Value: Median house value for households within a block, measured in US Dollars. This is the value of interest, in fact we are going to apply the ridge regression for predicting this value. - Ocean Proximity: Location of the house with respect to ocean. This is the only categorical variable contained in our dataset. Also in this case, we are going to see during the computations how to deal with this issue. For the analysis we are going to keep the column of median house value apart since we will need it for predicting the labels. So in the computations, we have created an object containing just the target value and another one containing all the explanatory variables.

3 Preprocessing and preliminary observations

The first thing that we have done beginning our analysis was of course importing the dataset of California housing. The result that we obtained is shown below in Figure 1:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

Figure 1: How the dataset looks like

The first thing that we have noticed was that one of the variable in our dataset actually contained some missing entries. More in details, the variable `total_bedrooms` contained 207 missing values. We have decided to drop those records since, having deleting them would have not be caused any dramatically changes in the overall analysis. Moreover, one of the variable appeared to be categorical and this could have also caused some problems. This variable was `ocean_proximity` which explained the proximity of the houses to the oceans according to four different levels. Thanks to the function `get_dummies` we were able to transform the variable into a dummy one. In this way, just the level corresponding to the observation takes a value equal to 1 and all the others 0. Once having done this, we have plotted the median house value variable on a

scatterplot in which latitude was plotted in the vertical axis, whereas longitude on the horizontal one. In this way we could have a clear representation of the shape of California and where the most expensive houses were located. As we can see from the picture in Figure 2, luxury houses are near the ocean in particular two areas have the most richer market for dwellings: San Francisco (the northern part) and Los Angeles (the southern part). Not surprisingly, the houses costs more in the big cities of California and near-by the ocean.

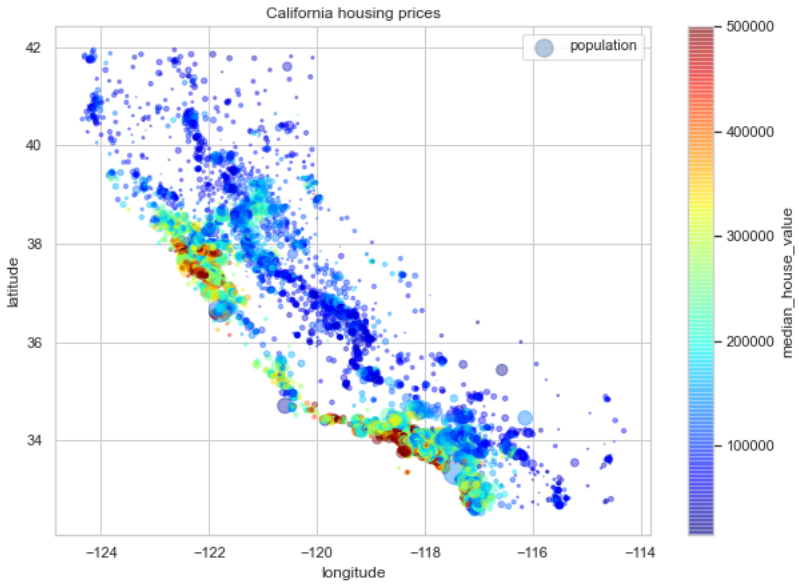

Figure 2: Location of houses in California according to their value

Moreover, we can also plot the distribution of the median house value, the variable that we will use for predicting the labels. The outcome is shown in Figure 3. As we can see there is a shrinking on the left, having a mode between 100000 and 200000 but we have also different values higher than 500000. Since these can be seen as outliers for our analysis, we have decided to not take into consideration those values. Since this variable, as already said will be used for predicting the labels, we have dropped it from the original dataset, storing it in a new object called y.

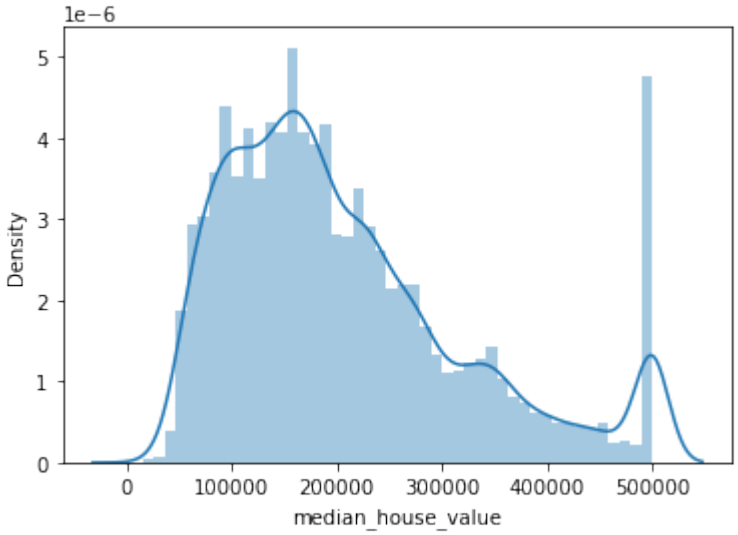

Figure 3: distribution of median house value

At this point of the analysis, we can standardize our data. This is extremely important especially when we are going to deal with the principal component analysis. In fact, if we don't standardize our data, the risk is that the principal components will put more weight on some data, making the analyses misleading themselves. Besides of the principal component analysis, it is really important to standardize the data, making their mean equal to zero and variance equal to unity. Once done this, we can begin the analysis part, but before it is better to recall some notions.

4 Ridge Regression

Before entering into the details of our analysis, it is better to understand further and better what ridge regression is and how it works. When we deal with linear regression, it can occur to have the need of increasing the bias in order to reduce the variance. This method will also produce a more stable model. Ridge regression works in this sense and could help in stabilizing the model, decreasing the variance at the cost of increasing the bias. For doing this, in ridge regression we introduce a regularizer in the empirical risk minimization (ERM). Ridge regression takes this form:

$$\hat{w} = argmin_{w \in R^d} ||Sw - y||^2 + \alpha ||w||^2$$

where α is the regularizer parameter, which takes value just greater than zero. Needless to say, when α is equal to zero, we are computing a linear regression solution. The parameter can be used to control the bias of the algorithm

as when it takes large values, the solution $\hat{w}$ becomes the zero vector.

Similarly to what happen with the linear regression, we will have that the minimizer satisfying the condition $\nabla F(w) = 0$ will thus be:

$$\nabla\Big(\, Sw - y \,^2 + \alpha \, w \,^2 \, \Big) = 2S^\top (Sw - y) + 2\alpha w$$

And consequently, the gradient vanishes for $w = \big(\alpha I + S^\top S\big)^{-1} S^\top y$ In this case we don't have to worry about the invertibility of $S^\top S$ as $\alpha I + S^\top S$ is invertible whenever $\alpha > 0$

Ridge regression is also useful when we are dealing with a dataset that present multicollinearity: as to say a situation where the variables on which we have to work present a strong correlation between each other.

5 Multicollinearity

As we have previously said, Ridge regression is particular useful when we are dealing with a dataset in which different variables appear to be highly correlated between each others. In this part of the analysis we are going to see if this is the case also for the California houses' dataset. First of all, we have dropped one of the five levels of the categorical variable that we have transformed in dummy. This is because of course one of them will be collinear to the others. In other words, knowing the value of four out of five of them, could bring us also to understand which will be the value of the fifth one. For this reason, randomly, we have decided to drop the level 'inland'. Then we have studied the correlation between all the other numerical values thanks to the use of an heatmap. The result is shown in Figure 4. As we can see through the outcome, we have an highly correlation between households, total rooms, total bedrooms and population. When we are talking about highly correlation we means that their Pearson's coefficient (which takes values between zero, when variables are not correlated, and one when they are strongly correlated) is higher than 0.5.

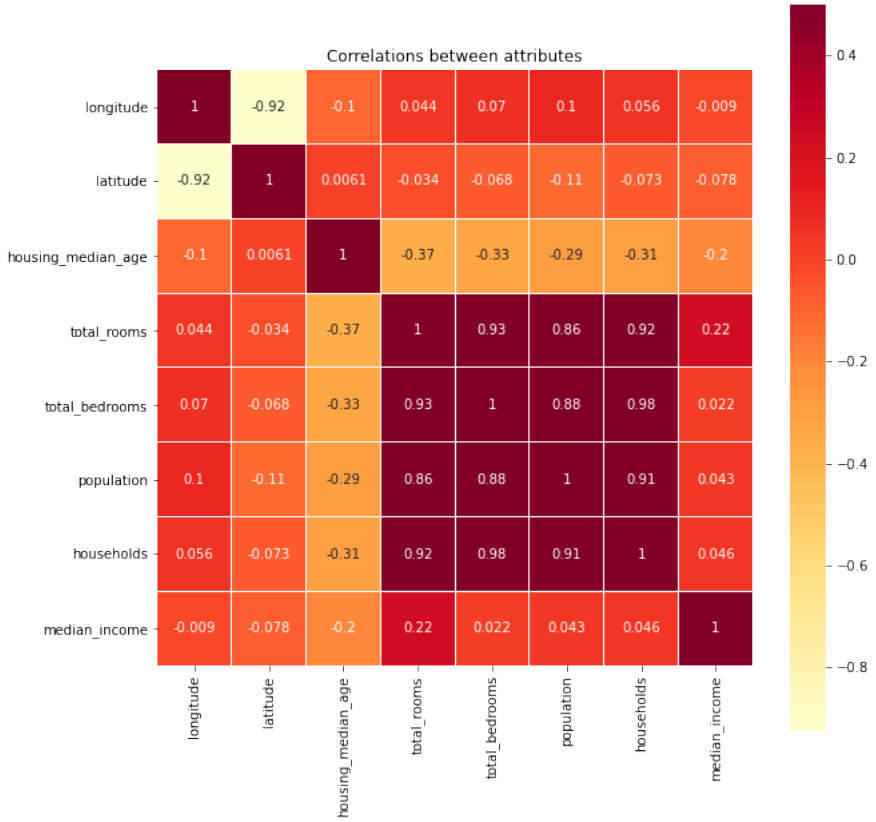

Figure 4: heatmap showing the correlations between the variables in our dataset

The result seems to make sense, in fact if we have an higher number of rooms, of course we will have also an higher number of bedrooms and households. The same is true for the population. When we are dealing with highly correlated variables, we usually deal with correlated pairs and so we can get along with our analysis dropping all the correlated variables with except of one. For this reasons we have decided to drop households, total bedrooms and population. There is no a particular reason for dropping exactly these variables, the important notion here is to keep just those that are uncorrelated between each other. Moreover, we can see through Figure 4 that there is a strongly correlation also between longitude and latitude. This is because of the particular shape of California who brings this type of outcome. In the code that we have used, we have created an heatmap for each time that we have dropped a variable. For summary reasons, we have decided to present here just the first one, representing the multicollinearity of the variables, and the last one. In Figure 5 we can see the definitive heatmap in which, as we can see, there are no more correlated

variables.

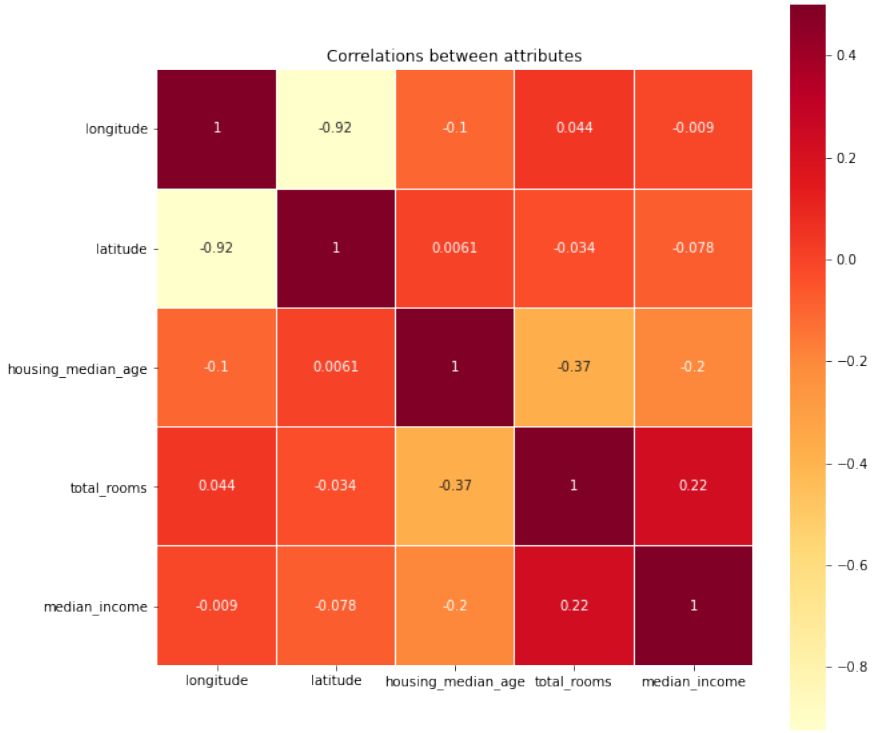

Figure 5: heatmap having dropped the correlated variables

6 Analysis

We are now able to perform the analysis on our clean data, having analysed the collinearity and excluded the highly correlated variables. First of all we have applied the Ridge regression algorithm in order to find out . At the beginning we have fixed a value of α equal to 0.2 for finding out the coefficient of the regression. Here however, we are working on the entire sample, as to say on the training set, without taking into consideration a part of the data on which we can "prove" our algorithm by knowing the error. For doing this part of the analysis we can actually choose between two strategies: dividing our dataset into a portion of training set and a portion of test set (e.g. 70% as training set and 30% as test set) or we can use the external cross validation. We have decided to use the latter approach since it could bring different advantages. In this case we are partitioning the dataset into 5 different folds, taking just one of it as test set while the remaining as training set. We, then, perform this computation taking once all the folds as testing parts, calculating the mean square error of the algorithm. We have, then, average those errors and taken them for 100 different values of α. The meaning of this last passage is to check how much the error change due to different values of . The plot in Figure 5

shows the trend of the error. Since errors' values are very close one to the other, we don't have a clear picture of it. In fact, for all the different values of α, we have an estimator that range between 0.4212 to 0.4213.

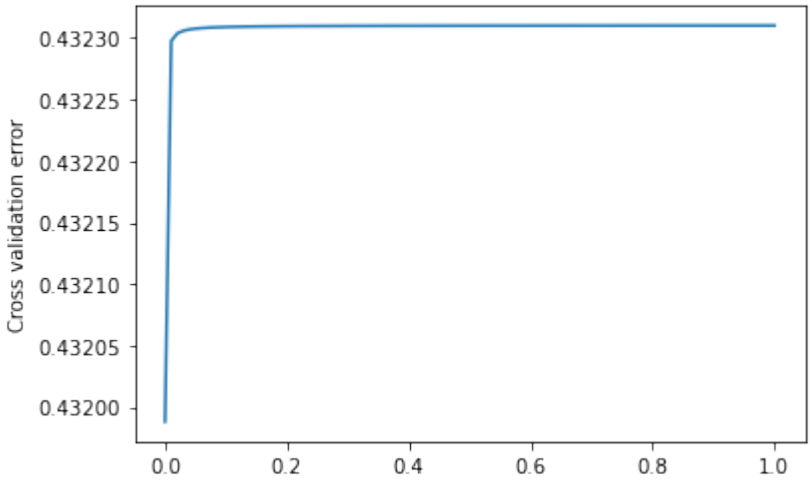

Figure 6: Cross validation error given different values of α

7 Principal component analysis

We now turn into a new phase of the analysis: The principal component analysis (PCA) This technique allows to project data onto a lower-dimensional space without loosing too much information. In order to do that, we are going to compute the singular value decomposition thanks to the function `linear algebra (LA)` contained in the Python's library `Numpy`. The singular value decomposition (SVD) is the generalization to non-square matrices of the spectral decomposition. The formula defined the SVD is specified as follow:

$$X = U\Sigma V^\top = \sum_{i=1}^{d} \sigma_i u_i v_i^\top$$

Where U and V are orthonormal matrices. The columns of U are the eigenvectors of $XX^\top$. Whereas the columns of V are the eigenvectors of $X^\top X$. The singular values are the eigenvalues of $\sqrt{XX^\top}$ or, equivalently, of $\sqrt{X^\top X}$. Coming back to our practical part, we have re-introduced in the dataset the

correlated variables that we previously left apart for performing the ridge regression algorithm. Now, we have to pop up those columns who belong to the categorical value, that we have transformed into dummies, this is because they

appear to be supplementary for this kind of analysis. So we have used the object `housing` which was the first one used for initialize all our computations. There were still some missing variables in that dataset, we don't know exactly why since we have dropped them at the beginning, but anyway we had to drop them, since it's extremely important to not have any missing variables in this part of the analysis as otherwise we cannot compute the singular value decomposition. Once again, we have created a new variable, named `f` containing just the variable `median house value` which is the one that we use for predicting labels. At this point we were able to compute singular value decomposition. We have then looked at the matrix of eigenvectors of the correlation matrix (`Vh` in the project). We then performed and saw how much variance was explained by each principal component. For doing that we have computed the explained variance as follow: $cumsum(s)/sum(s)$. So basically, we have taken the eigenvalues resulting from the singular value decomposition. We have computed the cumulative sum of them which was then divided by the sum of the eigenvalues. We have then plotted the result in Figure 7. As we can see through the picture, we have that the first four principal components explain a good amount of variance. In fact, the first one explains 0.315 of the variance, the second 0.224, the third 0.167 and the last one 0.140 portion of the variance. So, the first four principal components explains an overall variance of 84%.

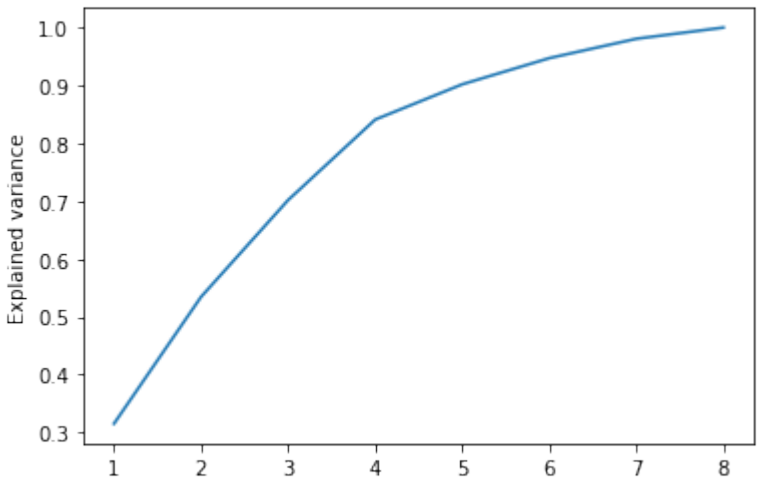

Figure 7: Scatterplot of the explained variance by the principal components

Now that we know how many principal components do we need in order to project data onto a low dimensional space without losing too much information, we can take just the first four principal eigenvector, since four are the principal components that we have decided to use. We have transposed them in order

to be able to multiply them with the original matrix, as to say the dataset containing the variables that we are studying. In nutshell, at this point of the analysis, we are taking our original data but we are "transforming" them in order to be expressed in an other way. Now we were able to performed the ridge regression predictor on the new matrix composed by the original one multiplied by the first four eigenvectors. Of course now, the result is a four components one as we are reducing the dimension of the analysis. We have then computed the cross-validation error in this case too. Recall that we have computed the cross validation error in 5 different folds using once one of those folds as test set. Then, we have averaged the results obtained and calculate it for 100 different values of α. Once again we have plotted the result of the error and this can be seen in Figure 8. The error in this case takes a wider range considering the different values of α with respect to the result that we had in the case of the original dataset. Moreover, the result shows that as much as α increase, the error increase too. This is because α gives more stability to the predictors at the expense of an higher error. Instead, if we compare this result to the one that we obtained with the original dataset, we can see that the error is bigger also because even if we are catching as much variance as possible, we are still missing some information.

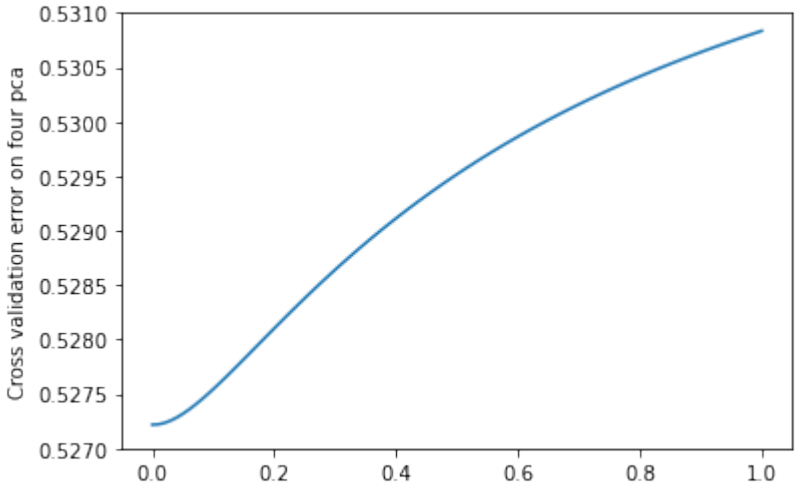

Figure 8: Cross validation error for the first four principal components

8 Conclusions

The aim of this project was to implement from scratch the Ridge regression algorithm for square loss for predicting the median house value of California. The project began with a previous look of the dataset in which we have seen that the more expensive houses were located on the seaside and close to big cities. Looking at the distribution of the target variable, we have recognize a not indifferent amount of luxury houses, which we have decided to drop since could have caused problems in the analysis. We have also dropped the missing values and transformed a categorical variable into a dummy one. We have then standardized the values which constitutes the pre-processing phase of the work. We have then studied the correlation between variables, leaving apart those that appeared to be highly correlated between each others. We have then applied the ridge regression and evaluated its performance trough a 5-fold cross validation which appeared the best solution to be applied, according to 100 different values of the regularizer term α. Once done this, we have plotted our result, viewing that beside having computed the error among various values of α, the former appeared to be not so different. We have then performed the principal component analysis, in order to visualize our work into a low-dimensional space. We have used the singular value decomposition for finding out the eigenvalues and taking them for calculating how much variance each principal component explained. We have seen that the first four principal components were those that explained an enough amount of variance: exactly equal to 84%. For this reason we have used them to create a new matrix starting from the original one called `M`. Once done this, we have applied the ridge regression on the matrix `M` and computed the cross validation error. Also these errors according to 100 different values of α have been plotted. For making a conclusion we can say that the estimation of the ridge regression actually does not change so much given different values of α. Moreover, the cross validation errors estimated on the new matrix composed by our data expressed by the four principal components, are higher than those estimated on the original matrix. This is because, even if we were careful and we took the principal components which explained as much variance as possible, we are still loosing information and so the error will be bigger. Moreover, in this case, as the level of α increases, the error increases as well as this parameter gives more stability to the predictors at the expense of an higher error.

9 Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it,

has not been previously submitted by me or any other person for assessment on this or any other course of study.