California housing prices prediction using ridge regression

Riccardo Bona

Department of Computer Science, University of Milan, Via Giovanni Celoria 18,
20133 Milano, Italy

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.

Abstract. The project consists in the analysis of the performance of rigde regression with square loss and its risk estimate dependence on the the regularization parameter *alpha*, evaluated on the prediction of the median house values of the California housing dataset. The dataset contains 20640 observations on a total of 10 variables. The ridge regression algorithm has been implemented without the employment of pre-existing libraries and has been tested using *k-fold cross validation* and *nested cross-validation.* The algorithm performance has been evaluated after a phase of data cleaning and manipulation, testing the impact on the prediction error of outliers removal, features selection based on multicollinearity analysis and *principal component analysis.* The results show no benefit in dropping multicollinear features and minor ones in performing dimensionality reduction through *PCA*. The *cross-validated* results demonstrate no objective *overfitting* in the prediction of the label *medianHouseValue*, highlighting no distinct positive impact of the regularization with both training and testing error significantly increasing with higher values of the parameter *alpha* and with the testing error always resulting slightly higher than the training one.

Keywords: Machine learning · Linear prediction · Regression.

1 Introduction

When performing linear regression for square loss, the predictor will try to fit the *training set* by finding the vector $w \in \mathbb{R}^d$ of coefficients that minimize the sum of squared differences between the model's predictions $h(x) = w^T x$ and the real target values y,

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{t=1}^{m} (w^T x_t - y_t)^2 \tag{1}$$

which in matrix notation translates in:

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left\| Xw - y \right\|^2. \tag{2}$$

Being $\left\| Xw - y \right\|^2$ convex, the gradient $\nabla \left\| Xw - y \right\|^2 = 2X^T(Xw - y) = 0$ is solved for the vectors of weights:

$$\hat{w} = (X^T X)^{-1} X^T y. \tag{3}$$

Thus the model will fit the observation as better as possible by finding the weights minimizing the sum of squared errors. This, however, could lead to potential *overfitting* issues, resulting in an overly specific and complex model with a low training error (low bias) by relying too much on the training data. If the learnt model overfits, it could incur in a high test error (high variance) by fitting new testing data poorly.

Given a learning problem defined by $(D,\ \ell)$ where D is the random draw with respect to the set of training examples $\mathbb{X}$ and ℓ the loss function, we call h^*_A the predictor with the lowest risk $\ell(h^*_A)$ among all predictors in the set H_A, output of an algorithm A. We can define the loss of a generic predictor h with respect to the random draw D on the training set $\mathbb{X}$ as

$$\ell_D(h_{\mathbb{X}}) = \ell_D(h_{\mathbb{X}}) - \ell_D(h^*_A) + \ell_D(h^*_A) - \ell_D(f^*) + \ell_D(f^*). \tag{4}$$

Noting that f^* is the *Bayes optimal predictor* given D and ℓ defined as

$$f^*(x) = \underset{\hat{y} \in Y}{\operatorname{argmin}} \mathbb{E}[\ell(Y, \hat{y}) | X = x] \tag{5}$$

which in the particular case of the square loss function $\ell(y, \hat{y}) = (y - \hat{y})^2$, can be written as

$$f^*(x) = \mathbb{E}[Y | X = x] \tag{6}$$

$\ell_D(h_{\mathbb{X}})$ can be decomposed in three parts:

1. variance error: $\ell_D(h_{\mathbb{X}}) - \ell_D(h^*_A)$, responsible for overfitting;
2. bias error: $\ell_D(h^*_A) - \ell_D(f^*)$, responsible for underfitting;
3. Bayes error: $\ell_D(f^*)$, which solely depends on D and ℓ, meaning it's uncontrollable.

To reduce a model's high variance, due to its instability, a regularization parameter can be added to the standard linear regression solution. This model, called *ridge regression*, exploits a bias injection in the prediction by penalizing the weights, shrinking them towards zero as the regularization parameter α increases, as a mean to counterbalance the instability of the model due to the potential multicollinearity of the predictors that could lead to a singular or nearly-singular $X^T X$ matrix, thus susceptible to errors in the training data. This way, the *bias-variance* trade-off can be balanced by finding a weight $\hat{w}_\alpha$

leading to higher bias but lower variance than the weight $\hat{w}$, effectively mitigating eventual overfitting issues.
The $\hat{w}_\alpha$ that minimizes the loss function can then be written as:

$$\hat{w}_\alpha = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left\| Xw - y \right\|^2 + \alpha \left\| w \right\|^2 \tag{7}$$

with $\alpha \geq 0$ (obtaining the linear regression solution for $\alpha = 0$) controlling the amount of bias introduced in the solution. The gradient $\nabla \left\| Xw - y \right\|^2 + \alpha \left\| w \right\|^2 = 2X^T(Xw - y) + 2\alpha w = 0$ is then solved by

$$\hat{w}_\alpha = (X^T X + I\alpha)^{-1} + X^T y \tag{8}$$

resulting in a reduction of the instability of the $(X^T X)^{-1}$ matrix.

1.1 Ridge regression implementation

Given the aforementioned premises, the ridge regression has been implemented as a function calculating the weights $\hat{w}_\alpha$ using the formula described in equation 8. Thus the prediction $\hat{y}_t$ is obtained as follows

$$\hat{y}_t = \hat{w}_\alpha^T x_t \tag{9}$$

2 Preliminary work and methodology

2.1 Dataset

The *dataset* employed in the project, consists in 20640 observations on a total o 10 variables:

- *longitude*: a measure of how far west a house is; a higher value is farther west;
- *latitude*: a measure of how far north a house is; a higher value is farther north;
- *housingMedianAge*: median age of a house within a block; a lower number is a newer building;
- *totalRooms*: total number of rooms within a block;
- *totalBedrooms*: total number of bedrooms within a block;
- *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;
- *medianIncome*: median income for households within a block of houses (measured in tens of thousands of US Dollars);
- *medianHouseValue*: median house value for households within a block (measured in US Dollars);
- *oceanProximty*: location of the house w.r.t ocean/sea.

The dataset contains 207 missing values for the column *totalBedrooms* and *oceanProximty* is a categorical feature. The rest of the features are numerical.
The target feature for the predicition is *medianHouseValue*.

2.2 Anomalies handling

Missing values imputation. As mentioned before, the column *totalBedrooms* contains 207 missing values. By plotting an histogram of the data it is clear how skewed to the right the distribution of *totalBedrooms* is. For this reason the missing values of the feature have been imputed using the median of the non-null values of the column, as a meaningful centrality index.

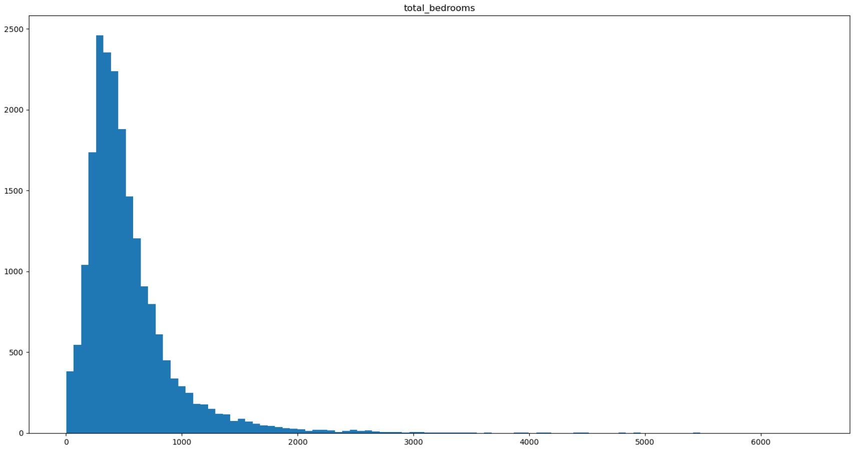

Fig. 1: Histogram of the *totalBedrooms* column.

Categorical features analysis and encoding. The feature *oceanProximity* is categorical and its unique values are: *INLAND*, *1<H OCEAN*, *NEAR BAY*, *NEAR OCEAN* and *ISLAND*. By plotting the frequency table of *oceanProximity* it is observable that the label *ISLAND* is barely represented in the dataset with only 5 observations in relation to the other labels belonging to at least 2000 rows in the data. Even if misrepresented in the dataset, the label has been kept in order not to lose information.

The column *oceanProximity* has been *one-hot* encoded, creating 5 new columns for each level of *oceanProximity* containing the respective binary values. The reason for choosing one-hot encoding over label encoding is the uncertain ordinality between the labels of *oceanProximity* which could be encoded on a given set of 5 integers, assigning the smallest value to the level representing the geographical area closer to the ocean and the biggest value to the label relative to the farthest one. However, by plotting the *oceanProximity* values in relation to *latitude* and *longitude* it is unclear for example, whether houses situated in the *NEAR BAY* area are closer to the ocean as opposed to the ones in the *1<H OCEAN* zone (Fig. 3), leading to an ambiguous ordinality between the labels. Even if this method involves adding four more dimensions to the dataset, a total of 14

features is acceptable. For this reason, in order to prevent the algorithm from interpreting an inaccurate ordinal relation, *one-hot encoding* has been chosen over *label encoding*.

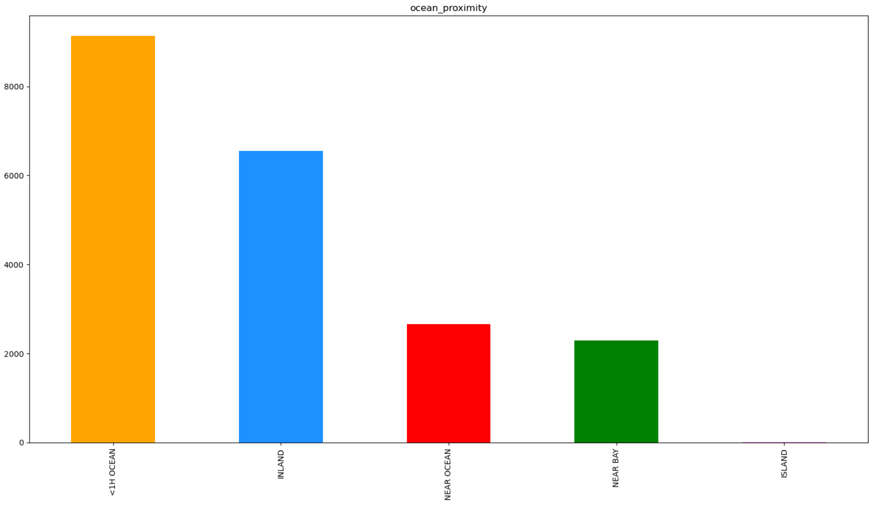

Fig. 2: Barplot for the *oceanProximity* frequencies.

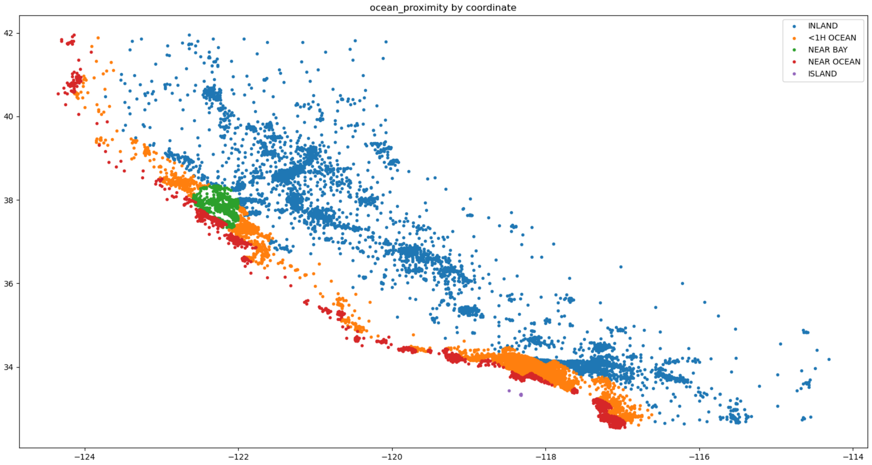

Fig. 3: *oceanProximity* values for each respective coordinate.

Outliers detection. Dataset outliers have been detected using the interquartile range rule counting, for each feature, the number of observations for which $x < Q1 - (1.5)IQR \ \vee \ x > Q3 + (1.5)IQR$.
6 columns out of 9 (excluding *oceanProximity*) contain outliers.

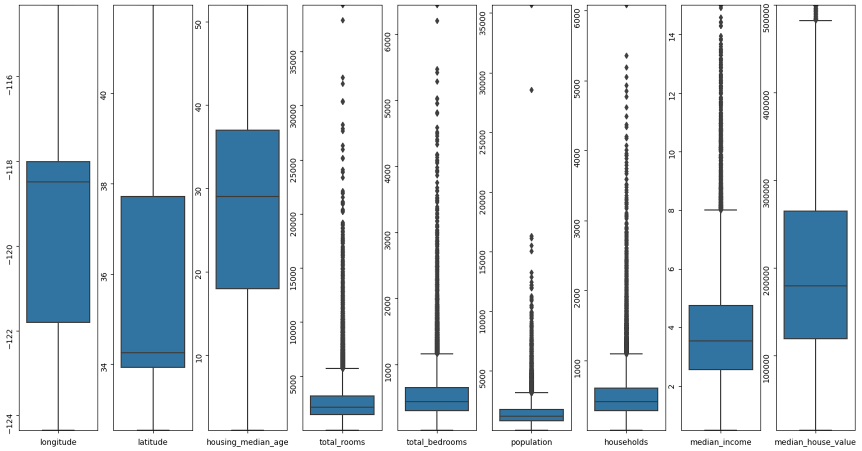

Fig. 4: Boxplots for each feature (excluding *oceanProximity*).

Dropping all outliers, reduces the dataset down to a size of 16898 observations.

Considerations on capped attributes. By looking at the histograms of all numerical attributes

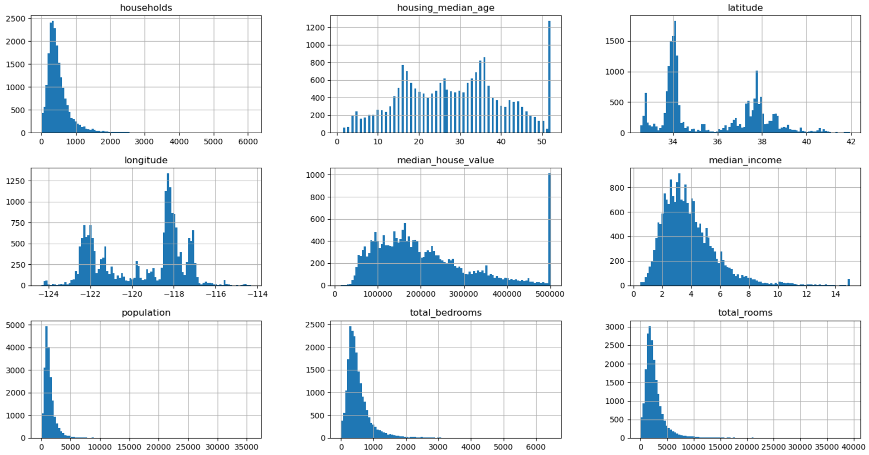

Fig. 5: Histogram for each feature (excluding *oceanProximity*).

it is evident that the attributes *housingMedianAge* and *medianHouseValue* have been capped for values higher than 52 and 500001 respectively. Since *medianHouseValue* is the target variable for the project, the capping of its values could result in a negative impact on the model's performance by leading to erroneous predictions.

Correlation and multicollinearity. By calculating the correlation map for all attributes (Fig. 6) can be observed that the only promising feature, in terms of correlation with the target variable *medianHouseValue*, is *medianIncome* with a Pearson correlation coefficient of 0.69, while all other attributes appear to be uncorrelated.

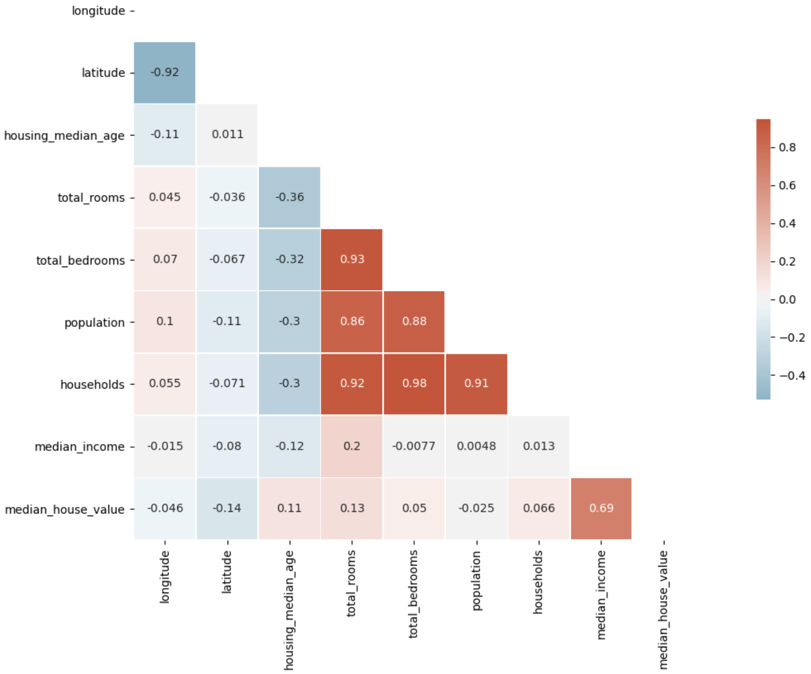

Fig. 6: Correlation heatmap for all attributes (excluding *oceanProximity*).

The correlation map shows, however, the presence of multicollinearity between some of the predictor variables, namely *latitude* and *longitude* with a strong negative linear relatioship and *totalBedrooms*, *population* and *households* with a high positive correlation coefficient among each other. Multicollinearity can lead to imprecise predictions and redundancy in multiple regression models, by using ridge regression and introducing a regularization parameter the effects of multicollinearity could be mitigated. The algorithm has also been tested on the dataset without multicollinear features. The results of this approach are described in subsection 3.3 of this report.

Final processing. As a final step, excluding the target variable, the dataset has been rescaled and centered by performing standardization. Since the dataset attributes present very different value ranges, standardization is a mean to ensure that the weights of the *rigde regression* solution are coherent and on the same scale. In particular, since regularization involves a shrinkage of the weights, standardization prevents cases in which a weight is overly penalized only due to the value scale of its corresponding attribute.

In addition, an extra column of values equals to 1, has been added to the dataset in order to represent the intercept term in the regression.

3 Experimental results

Prior to testing the algorithm, the dataset has been preprocessed as follows: the missing values of *totalBedrooms* have been imputed using median, all attributes except *medianHouseValue* and *oceanProximity* have been standardized, *oceanProximty* values have been one-hot encoded and the intercept term has been added. The rigde regression algorithm has been tested using *K-fold* and *nested cross-validation* on three different versions of the dataset, namely:

1. Dataset (a): No values removed;
2. Dataset (b): *medianHouseValue* capped values removed (i. e. those ≥ 500001);
3. Dataset (c): outliers removed for all attributes (using the previously described interquartile range rule).

The principal metric used to evaluate the model is *Mean Squared Error*. *Root Mean Squared Error* is taken in account as a more readable result metric. *Adjusted* R^2 score is also computed and reported in the test results.

3.1 Cross-validation

The algorithm performance has been tested using *K-fold cross-validation* in order to observe the dependence of the prediction error on the parameter *alpha*, employed in the ridge regression solution. In addition *nested cross-validation* has been implemented and used to evaluate the algorithm without the need of choosing the value for *alpha*. The number of folds K for the external-cross validation is set to 5, as well for the number of folds in the internal-cross validation for the nested version. The different tested values for *alpha* are $\alpha = \{0, 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 50, 100, 200, 300, 400, 500\}$. For the *nested cross-validation*, the best α value for each internal-cross validation is chosen by picking the one leading to the lowest *Mean Squared Error* value.

In order to avoid unwanted influences on the model's results due to how the data has been collected, the dataset is randomly shuffled before performing the two cross-validation techniques. Since the cross-validated results change slightly with different shuffling instances, the final results have been obtained by averaging the results of multiple test runs (in the case of the results reported in this paper, the number of test runs is 10).

K-fold cross-validation. *K-fold cross-validation* has been implemented as following:

1. The dataset is randomly shuffled and subsequently partitioned into K equal parts (the default value is $K = 5$);
2. For each fold $k = 1,...,K$: k is retained as the test set, while the rest of the dataset minus the fold k is used as the training one;
   2.1. For each α value: the ridge regression algorithm is run on the training part and the resulting model is applied to the testing set;
   2.2. *MSE* and *adjusted* R^2 are calculated for both training and test sets;
3. Each α value will be associated with K different training and testing scores, which averaged give the estimated performance of the algorithm with each tested α value.

Nested cross-validation. *Nested cross-validation* has been implemented as following:

1. The dataset is randomly shuffled and subsequently partitioned into K equal parts (the default value is $K = 5$);
2. For each fold $k = 1,...,K$: k is retained as the test set, while the rest of the dataset minus the fold k is used as the outer training set;
   2.1. The outer training part, is further split into J equal parts (the default value is $J = 5$);
   2.2. For each fold $j = 1,...,J$: j is retained as the validation set, while the rest minus the fold j is used as the inner training set;
      2.2.1. For each α value: the ridge regression algorithm is run on the inner training part and the resulting model is applied to the validation set;
      2.2.2. *MSE* is calculated for the validation set;
   2.3. The α value leading to the lowest average *MSE* over the J validation sets is chosen;
   2.4. the ridge regression algorithm is run with the chosen α value on the entire outer training part and the resulting model is applied to the test set;
   2.5. *MSE* and *adjusted* R^2 are calculated for both training and test sets;
3. The mean scores over all K folds are calculated.

3.2 Model's performance

In this section are displayed and analyzed the results for both *K-fold* and *nested cross-validation* on the three aformentioned variants of the dataset. For *K-fold cross-validation* are listed the average results for all tested α values and also the scores for the α values leading to the best results. The tables show the results for both training and test parts.

Case (a): Dataset with no values removed. The dataset with no values removed contains a total of 20640 observations.
By observing the results of the *K-fold cross-validation* for all tested α values, the average test error is higher by a small degree in relation to the training one. By looking at the *MSE* trend as α increases, as shown in Fig. 7, even if the distance between the two errors slightly shrinks with the increase of α, the test error remains higher than the training one. Since the difference between the training and test error is relatively small (even with $\alpha = 0$) and they increase almost equally with the regularization parameter, no objective *overfitting* can be observed in the prediction of the target variable. Moreover the best results in the cross-validated risk estimate are for low α values, with the best results for the training part obtained with $\alpha = 0$ and $\alpha = 0.4$ for the test one (Tab. 1). Tab. 1 also shows the training and test results for the *nested cross validation* using dataset (a), achieving slightly better test results than the average results for the *K-fold cross-validation*.

K-fold cross-validation

	MSE	**RMSE**	**Adjusted R^2**
Training average results	4728549203.94	68764.44	0.6446
Test average results	4751205714.57	68928.99	0.6418
Training best α: 0.0	4718662746.13	68692.52	0.6453
Test best α: 0.4	4742845269.91	68868.31	0.6424

Nested cross-validation

	MSE	**RMSE**	**Adjusted R^2**
Training results	4727635295.93	68757.80	0.6446
Test results	4750507626.68	68923.92	0.6420

Table 1: Model's results for the *K-fold* and *nested cross-validation* on the dataset with no values removed. $K = 5$.

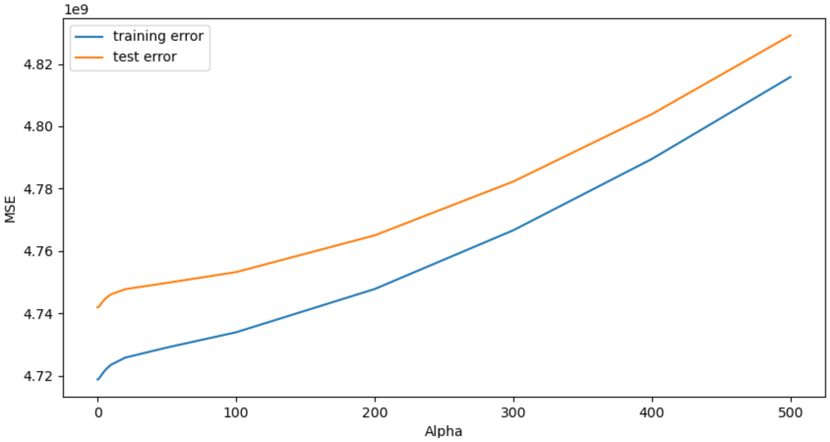

Fig. 7: *MSE* trend of the *K-fold cross-validation* results as *alpha* increases. Case in which the model is tested on the dataset with no values removed.

Case (b): Dataset with *medianHouseValue* capped values removed. The removal of the *medianHouseValue* capped values reduces the dataset down to a total of 19675 observations.

The results of the cross-validated risk estimate with dataset (b) show that removing the capped *medianHouseValue* values yields a significant reduction of the overall error in the prediction, at the cost of a smaller dataset (Tab. 2). As for the dataset (a), the increase of the regularization parameter α results in an increase of the *MSE* for both training and test parts in the *K-fold cross-validation*. As for the precedent case, the best results are achieved with small α values, in particular with $\alpha = 0$ for the training error and $\alpha = 0.4$ for the test one. In general, removing the capped values of the target variable doesn't change the relation between the α parameter and the cross-validated prediction error, while lowering it by a significant amount. However, the removal of the capped values also leads to a smaller *adjusted* R^2 value for both cross-validation methods.

K-fold cross-validation

	MSE	**RMSE**	**Adjusted R^2**
Training average results	3712989709.07	60934.30	0.6107
Test average results	3728426760.48	61060.84	0.6080
Training best α: 0.0	3703389544.57	60855.48	0.6117
Test best α: 0.4	3719896473.74	60990.95	0.6088

Nested cross-validation

	MSE	**RMSE**	**Adjusted R^2**
Training results	3708455981.48	60897.09	0.6112
Test results	3726544395.88	61045.42	0.6082

Table 2: Model's results for the *K-fold* and *nested cross-validation* on the dataset with *medianHouseValue* capped values removed. $K = 5$.

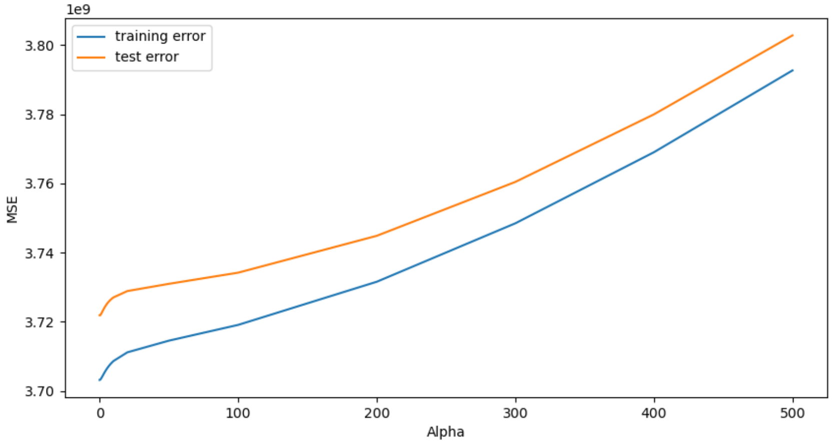

Fig. 8: MSE trend of the K-fold cross-validation results as *alpha* increases. Case in which the model is tested on the dataset with only the *medianHouseValue* capped values (≥ 500001) removed.

Case (c): Dataset with outliers removed. Removing all outliers results in a dataset of 16898 observations.
Employing dataset (c) leads to the overall best results between all three tested datasets. However, removing all outliers results in a much more smaller dataset consisting in 16898 observations, compared to the original size of 20640 observations. With dataset (c) *K-fold* and *nested cross-validation* achieved the lowest MSE values, while the MSE trend remained similar to the one of the two precedent datasets, with the MSE value increasing as α increases and the best results obtained using $\alpha = 0$ for the training error and $\alpha = 0.2$ for the test one. As for the precedent case the *adjusted* R^2 for case (c) is lower than the one for case (a) but which, however, results higher than the one for case (b).

K-fold cross-validation

	MSE	**RMSE**	**Adjusted R^2**
Training average results	3173577533.71	56334.51	0.6255
Test average results	3181462127.14	56404.45	0.6232
Training best α: 0.0	3162777556.36	56238.57	0.6266
Test best α: 0.2	3171105163.47	56312.56	0.6243

Nested cross-validation

	MSE	**RMSE**	**Adjusted R^2**
Training results	3164863700.68	56257.12	0.6264
Test results	3174171096.15	56339.78	0.6241

Table 3: Model's results for the *K-fold* and *nested cross-validation* on the dataset with outliers removed. $K = 5$.

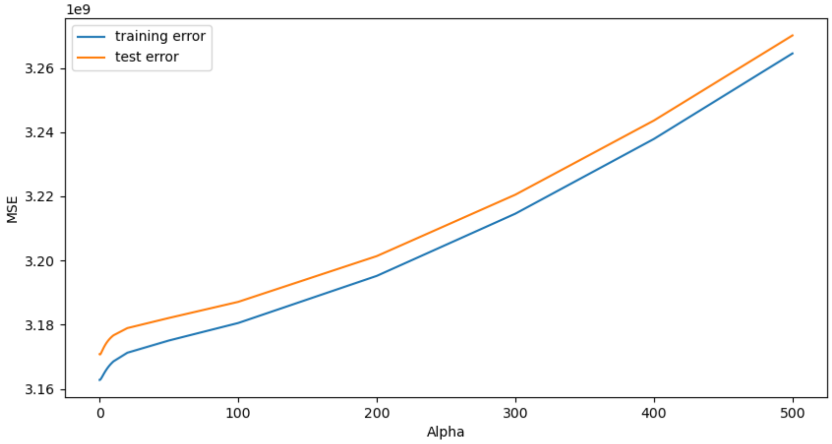

Fig. 9: MSE trend of the K-fold cross-validation results as *alpha* increases. Case in which the model is tested on the dataset with all outliers removed.

3.3 Impact of collinear features on prediction

As mentioned in subsection 2.2, the dataset presents five collinear features (Fig. 6) which may impact the model's performance. For the purpose of analyzing the influence of multicollinearity on the prediction, the ridge regression algorithm has been tested on the dataset without collinear features, removing those with the highest *variance inflation factor (VIF)*, measuring the amount of variance increase in a regression coefficient estimate due to mulicollinearity:

$$VIF = \frac{1}{1 - R^2}. \tag{10}$$

The dataset has been previously standardized and the columns have been dropped by removing the attribute with the highest *VIF* greater than a threshold set equals to 5, then subsequently recalculating the *VIF* for all remaining features and repeating the process until no attributes with *VIF*> 5 remain (Tab. 4).

Attribute	**VIF**	**Removed**
longitude	18.028444	No
latitude	19.925764	Yes
housingMedianAge	1.321927	No
totalRooms	12.349114	Yes
totalBedrooms	27.040073	No
population	6.342122	No
households	28.315383	Yes
medianIncome	1.740468	No
<1H OCEAN	1.352098	No
INLAND	2.227693	No
ISLAND	1.002389	No
NEAR BAY	1.432499	No
NEAR OCEAN	1.248673	No

Attribute	**VIF**
longitude	1.363410
housingMedianAge	1.301877
totalBedrooms	4.373679
population	4.306311
medianIncome	1.109646
<1H OCEAN	1.109962
INLAND	1.117049
ISLAND	1.000668
NEAR BAY	1.352339
NEAR OCEAN	1.006490

Table 4: On the left: *Variance inflation factor* value for each dataset attribute (excluding the target variable). On the right: *VIF* value for each remaining dataset attribute, after performing the removal.

After removing the most collinear features, the overall correlation between the dataset features is significantly lower (Fig. 10) , exception made for *population* and *totalBedrooms* with a correlation coefficient of 0.87. However, since their *VIF* values are acceptable, no further removal has been performed. The ridge regression algorithm performance on this particular instance of the dataset is to be compared to the one of case (a) (subsection 3.2) since no other droppings were performed. The algorithm has been tested using both *K-fold* and *nested cross-validation* with the same parameters of the testings described in subsection 3.2. The results show a net decrease in performance of the algorithm on the dataset without collinear features, implying a relevance of the removed features in the prediction of *medianHouseValue*, despite their contribution to the increase in multicollinearity and in variance inflation of the respective coefficients.

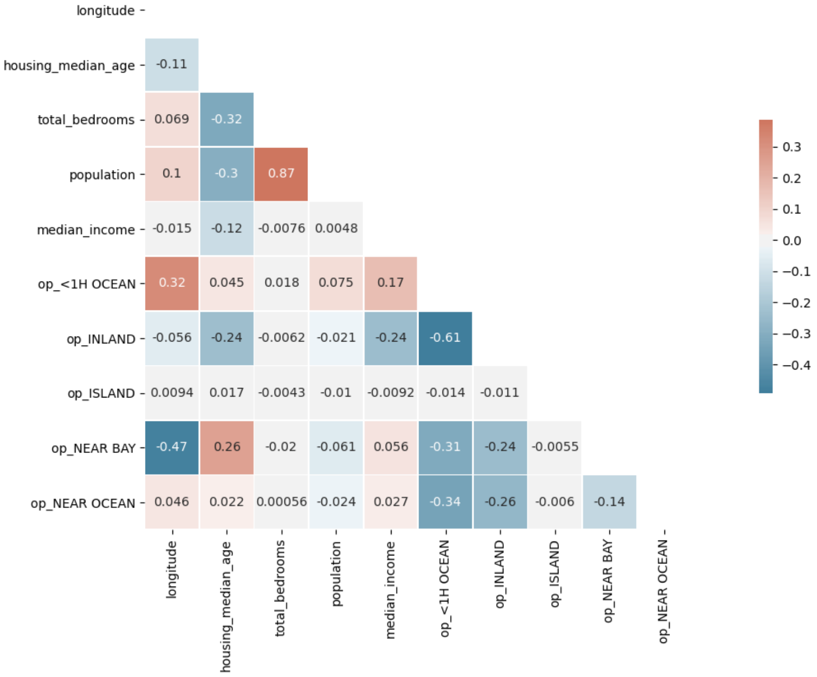

Fig. 10: Correlation heatmap for all remaining attributes, after performing the removal.

|***K-fold cross-validation***|

	MSE	**RMSE**	**Adjusted R^2**
Training average results	4935380757.92	70252.26	0.6291
Test average results	4949474902.38	70352.50	0.6272
Training best α: 0.0	4926286644.53	70187.51	0.6297
Test best α: 0.2	4941088845.92	70292.87	0.6278

|***Nested cross-validation***|

	MSE	**RMSE**	**Adjusted R^2**
Training results	4928520362.03	70203.42	0.6296
Test results	4943344773.50	70308.92	0.6277

Table 5: Model's results for the *K-fold* and *nested cross-validation* on the dataset with collinear features removed. $K = 5$.

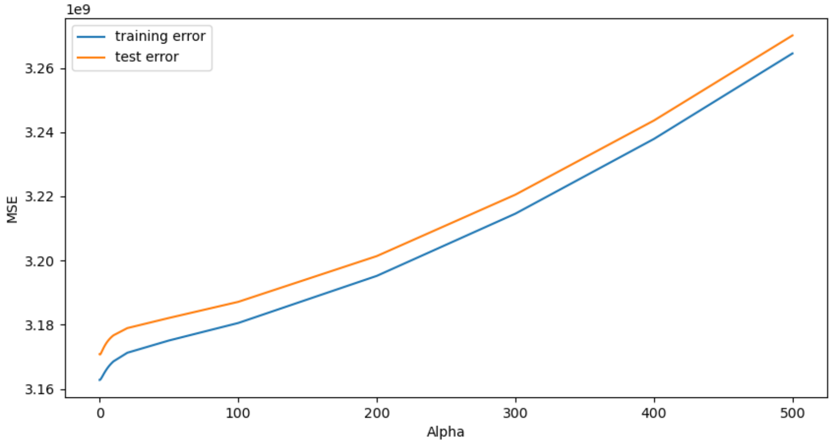

Fig. 11: MSE trend of the K-fold cross-validation results as *alpha* increases. Case in which the model is tested on the dataset with collinear features removed.

3.4 Impact of principal component analysis on prediction

As a last step in the project, *principal component analysis* (*PCA*) has been applied to the aforementioned three dataset instances, with the intent of improving the risk estimate. Being X the NxD matrix representing the dataset and Σ being the covariance matrix of X, *PCA* computes the *principal components* by finding the M eigenvectors with the largest eigenvalues of Σ, with M being at maximum the number of variables D in the dataset. Prior to computing the covariance matrix Σ, the data is centered by mean subtraction. Thus *PCA* allows for a representation of the orignal data matrix with fewer dimension while preserving the data dispersion. This is achieved by choosing subsequent M eigenvectors, with eigenvector i being orthogonal to eigenvector $i-1$, each one maximizing the explained variance of the original data.

PCA has been applied to the dataset with no values removed (dataset (a)), with capped values removed (dataset (b)) and with all outliers removed (dataset (c)). For each case, the performance of the ridge regression algorithm has been tested using *nested cross-validation*. As shown in table 6, the highest cumulative *explained variance ratio* values are achieved as more components are added. With the addition of the fourth component, the ratio value is around 0.9 for all three datasets, meaning that with at least 4 components, an acceptable approximation of the original datasets should be achieved. By looking at the results in table 7, it can be seen how the best test results are obtained in all three instances of the dataset with 12 or 13 components. However fig. 12 shows that really similar results can be achieved even using a number of components comprised between 6 and 13, for all three datasets.

With the optimal number of components *PCA* allows to achieve slightly better results, however, this means reducing the dimensions of the original dataset

Dataset	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
(a)	0.449	0.678	0.802	0.904	0.939	0.961	0.977	0.988	0.995	0.997	0.999	1	1
(b)	0.452	0.680	0.809	0.905	0.940	0.961	0.978	0.988	0.994	0.997	0.999	1	1
(c)	0.423	0.654	0.788	0.893	0.929	0.955	0.975	0.986	0.993	0.997	0.999	1	1

Table 6: For each dataset: cumulative contribution of each principal component to the *explained variance ratio*.

	Dataset (a)	Dataset (b)	Dataset (c)
Num. components	12	13	12
Train MSE	4720634008.72	3703928227.84	3163584898.21
Train RMSE	68706.87	60859.90	56245.75
Train adjusted R^2	0.6451	0.6116	0.6266
Test MSE	4740045053.03	3720144270.79	3170673911.23
Test RMSE	68847.98	60992.98	56308.73
Test adjusted R^2	0.6422	0.6087	0.6245

Table 7: For each dataset: best results (lowest test *MSE*) of the *nested cross-validation*, depending on the number of components used by *PCA*.

from 13 to at best 12 components. On the contrary, if the aim is reducing the dataset dimensions the performance will drop accordingly. However, with at least 6 components, the dataset can be reduced making computations more efficient without sacrificing too much prediction accuracy.

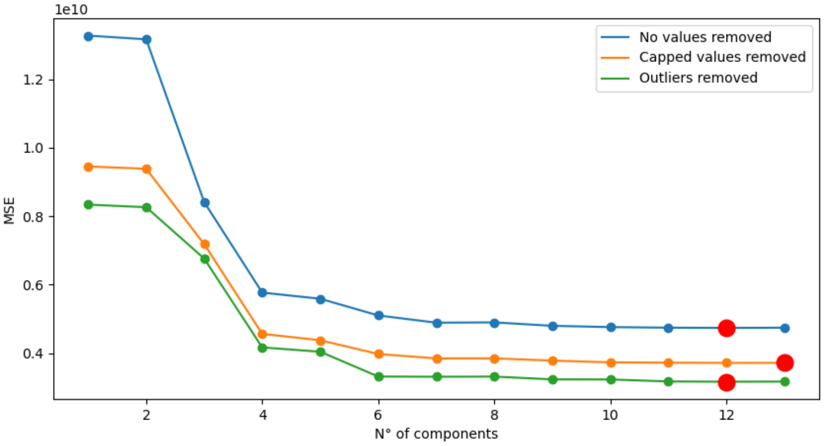

Fig. 12: For each dataset: MSE trend for the testing part of the nested cross-validation as more principal components are added. The red dots represent the lowest MSE values achieved.

4 Conclusion

In conclusion, the results of the *cross-validated* risk estimate show no objective overfitting since, in all tested cases the training error is always larger by a small amount than the testing one and both errors grow equally as the parameter α increases. This also underlines the fact that the best results are achieved with values of α close to 0. It can be concluded that the introduction of the regularization parameter has a negligible positive impact on the prediction. Using *nested cross-validation* doesn't improve the results but gives a more robust representation of the risk estimate and eliminates the need of choosing the value for α.

Overall, dropping the capped values of *medianHouseValue* improves both training and testing error but the best results are achieved by removing all outliers in the dataset at the cost of approximately 18% of the dateset observations.

Removing multicollinear features worsens both training and testing errors, implying the importance of the removed attributes in the target variable prediciton. *PCA* achieves minor improvements of the risk estimate with 12 or 13 components. If aiming to reduce the dataset dimensionality and improving computational efficiency, both errors increase but, with at least 6 components, a good approximation of the original dataset can be reached at the cost of a slighter larger errors in prediction.

References

1. Shai S.-S. e Shai B.-D., Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
2. Mehryar M., Afshin R. e Ameet T., Foundations of Machine Learning (second edition), MIT Press, 2012.
3. Thompson, B., Canonical correlation analysis. Encyclopedia of statistics in behavioral science, 2005.
4. Guyon, I., & Elisseeff, A., An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182, 2003.
5. Ross S. M., Introduction to Probability and Statistics for Engineers and Scientists (fifth edition), 2014
6. Tan P.-N., Steinbach M.,Kumar V., Introduction to Data Mining, (First Edition), Addison-Wesley Longman Publishing Co., Inc., 2005