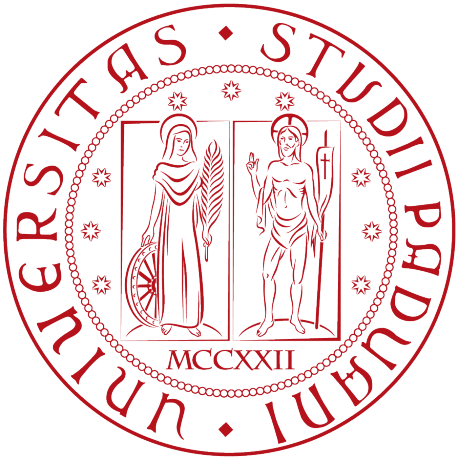
Predicting Diamonds Price

Statistical Learning Mod. B, Final project

Bernardi Alberto, Cracco Gianmarco

23rd June, 2020



Abstract

Our main goal for this project is to build an appropriate model in order to predict effectively the prices of diamonds. Using a reasonably large dataset with both categorical and numerical features such as the three spatial dimensions, the color, the carats and others, we try different approaches to the problem: in particular we apply *Feature Selection*, *Shrinkage Methods* and finally *Principal Component Analysis* in addition to the classical linear regression.

For this report we used as main references the notes and the material provided and suggested during the Statistical Learning Course hold during the academic year 2019/2020 by Professors Roverato Alberto and Scarpa Bruno at the University of Padua.

Contents

1 Diamonds Dataset[1]

For this final project we have chosen a dataset regarding diamonds coming from `https://www.kaggle.com/shivam2503/diamonds`. It contains roughly 54.000 features each one with 10 attributes, both numerical and categorical:

- Carat: a numerical feature that gives us the weight of a diamond, with 1 carat equivalent to 0.2 g.
- Cut: a categorical factor with 5 levels, that are, in ascending order: *Fair*, *Good*, *Very Good*, *Excellent*, *Ideal*.
- Color: a categorical factor with 7 levels: *J*, *I*, *H*, *G*, *F*, *E*, *D*. We reported them in ascending order, with *J* corresponding to a diamond near colorless and D to a colorless one.
- Clarity: a categorical factor with 8 levels, each classifing the type of inclusions and blemishes a diamond has: *I1* (inclusions),*SI2*, *SI1* (small inclusions), *VS2*, *VS1* (very small inclusions), *VVS2*, *VVS1* (very very small inclusion) and *IF* (internally flawless).
- Depth: a numerical feature which describes the percentage calculated by dividing the total height of a diamond by its average width.
- Table: a numerical feature which describes the percentage calculated dividing the width of the table (i.e. the top facet of a diamond) by the total width of the diamond.
- Price [$]: a numerical feature.
- x, y, z [mm]: three numerical features encoding the dimensions of a diamond along its three axis.

Clarity is a measure of how many imperfections are in the diamond, like dark spots, gas bubbles, white spots, cracks, or cloudiness. The cleaner a diamond is (the less imperfections), the rarer it is and the more it will cost even if the imperfections are barely visible to the naked eye.
The first four features are also called the $4C$: together they represent the universal method for assessing the quality of a diamonds. Given this fact, we tried to keep these features as important as they are (see Section 2.2.2).

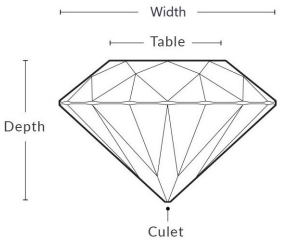


Figure 1: Diamonds structure

1.1 Data Preprocessing

We started exploring our data trying to capture meaningful structures and connections between them, focusing in particular on the possible relationship between variables and our target variable, the price of the diamonds.

[1]For the sake of brevity we report in this paper just a summary of our work; to have deeper insights on all the models and plots, take a look at the **R** code provided.

	X	carat	cut	color	clarity	depth	table	price	x	y	z
1	1	0.23	Ideal	E	SI2	61.50	55.00	326	3.95	3.98	2.43
2	2	0.21	Premium	E	SI1	59.80	61.00	326	3.89	3.84	2.31
3	3	0.23	Good	E	VS1	56.90	65.00	327	4.05	4.07	2.31
4	4	0.29	Premium	I	VS2	62.40	58.00	334	4.20	4.23	2.63
5	5	0.31	Good	J	SI2	63.30	58.00	335	4.34	4.35	2.75
6	6	0.24	Very Good	J	VVS2	62.80	57.00	336	3.94	3.96	2.48

Table 1: First rows of the dataset

Printing some lines of our dataset (see Table 1), we can observe how the first column has to be removed since it only encodes the indices of our instances, and the three categorical features will have to be handled via one-hot-encoding.
After a check for *Null Values* (that were not detected), we inspected the summary of our data (Table 2): the very first thing to be noticed is that x, y and z have minimum values equal to zero, which is an absurd value for a solid object, so we decided to remove all the instances with one of those three variables equal to zero from our dataset. Notice that we discarded those values instead of imputing them since they were just 20 instances in a dataset of almost 54000 elements, definitely a negligible number.

carat	cut	color	clarity	depth
Min. :0.2000	Fair : 1610	J: 2808	SI1 :13065	Min. :43.00
1st Qu.:0.4000	Good : 4906	I: 5422	VS2 :12258	1st Qu.:61.00
Median :0.7000	Very Good:12082	H: 8304	SI2 : 9194	Median :61.80
Mean :0.7979	Premium :13791	G:11292	VS1 : 8171	Mean :61.75
3rd Qu.:1.0400	Ideal :21551	F: 9542	VVS2 : 5066	3rd Qu.:62.50
Max. :5.0100		E: 9797	VVS1 : 3655	Max. :79.00
		D: 6775	(Other): 2531	

table	price	x	y	z
Min. :43.00	Min. : 326	Min. : 0.000	Min. : 0.000	Min. : 0.000
1st Qu.:56.00	1st Qu.: 950	1st Qu.: 4.710	1st Qu.: 4.720	1st Qu.: 2.910
Median :57.00	Median : 2401	Median : 5.700	Median : 5.710	Median : 3.530
Mean :57.46	Mean : 3933	Mean : 5.731	Mean : 5.735	Mean : 3.539
3rd Qu.:59.00	3rd Qu.: 5324	3rd Qu.: 6.540	3rd Qu.: 6.540	3rd Qu.: 4.040
Max. :95.00	Max. :18823	Max. :10.740	Max. :58.900	Max. :31.800

Table 2: Summary of data

From the histograms of the numerical variables in Fig. 2, it seems like all of them are symmetrical and uni-modal, except for price and carat which are really right skewed. Looking at the relative boxplots, on the other hand, we can see that data are spread out and there are many "outliers": because of their numerousness, we did not take any further action, except in the case of those values for which y and z seemed completely out of distribution: we selected the instances for which y and z were greater than 30 mm and, looking at these three diamonds (Table 3), we observed how their shape was very elongated along one axis, which made us suspect of an error in the data.

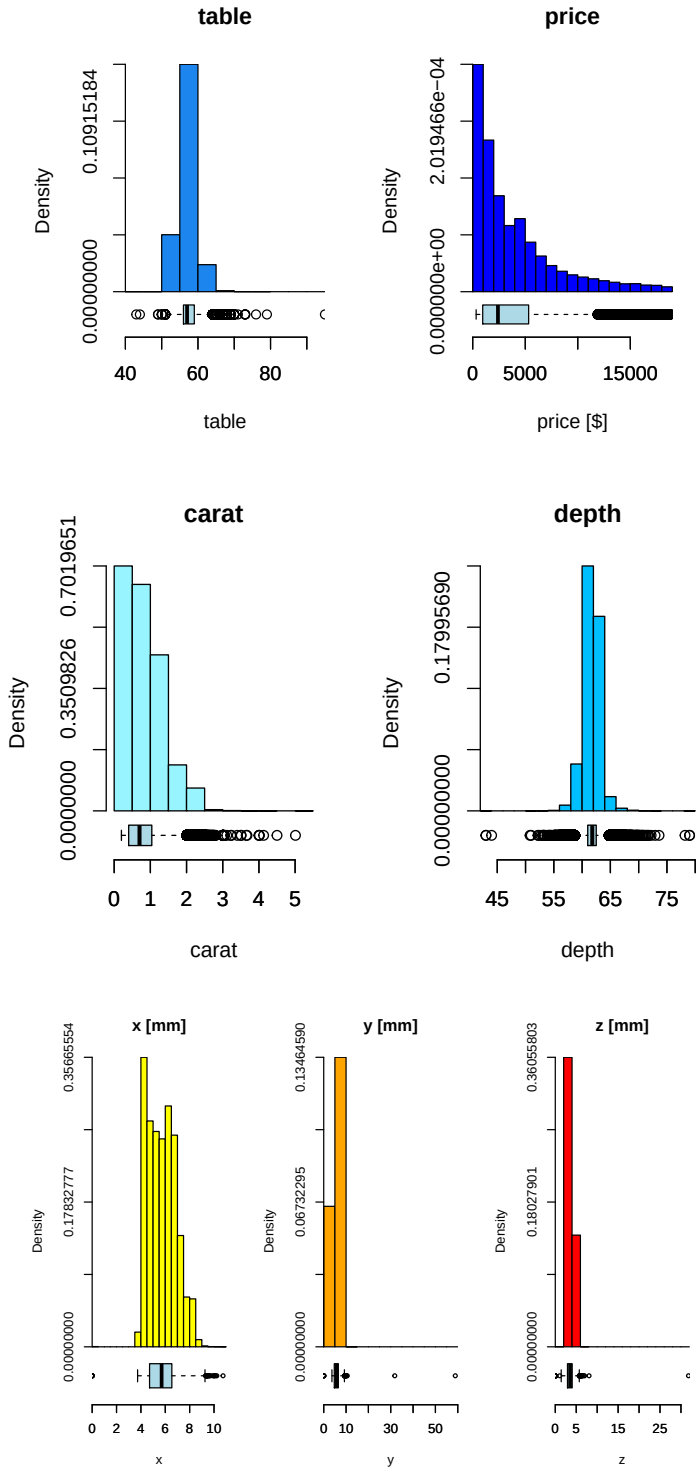


Figure 2: Numerical variables histograms and boxplots

	carat	cut	color	clarity	depth	table	price	x	y	z
24068	2.00	Premium	H	SI2	58.90	57.00	12210	8.09	58.90	8.06
48411	0.51	Very Good	E	VS1	61.80	54.70	1970	5.12	5.15	31.80
49190	0.51	Ideal	E	VS1	61.80	55.00	2075	5.15	31.80	5.12

Table 3: Outliers

Despite these values, we still decided to keep them: in fact, there is a too high variability associated with all our data to justify such a removal.

Looking at the distribution of the categorical features, instead, nothing of concern was noticed: the majority of diamonds presents fine *cut* and *color*, while *clarity* is concentrated among mid levels (Fig. 3).

We now dive deeper inside our dataset, trying to figure out the different dependencies between *price* and the other features.

First of all we considered the correlation matrix for our numerical features: we can observe that the price seems highly correlated to the carats and the dimensions of the diamond, while it seems almost independent of table and depth: observing also the scattered plots in Fig. 4, we can find a visualization of these facts, with table and depth presenting, for the same values, a huge range of different prices, while the other numerical variables seem to increase the price steeply. We are not surprised by these behaviours: in fact, typically an higher number of carats correspond to a higher price and similarly for the dimensions. Instead, table and depth, as said above, should generally assume a value around 60, depending also on the shape of the diamond, so it is natural that the admit a wide range of prices for the same value.

Looking instead at the boxplots relating categorical variables and price

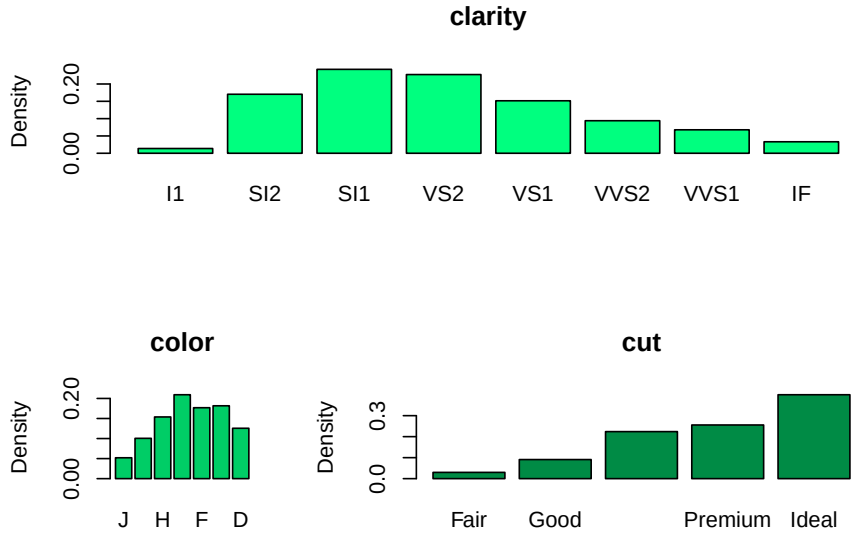


Figure 3: Categorical variables histograms

(Fig. 5), we can observe that the median is low for every class with respect to the prices that reach also very high values, a sign that, whichever category may be considered, there is a large quantity of diamonds still very expensive. This may suggest how the contribution of numerical features is fundamental and

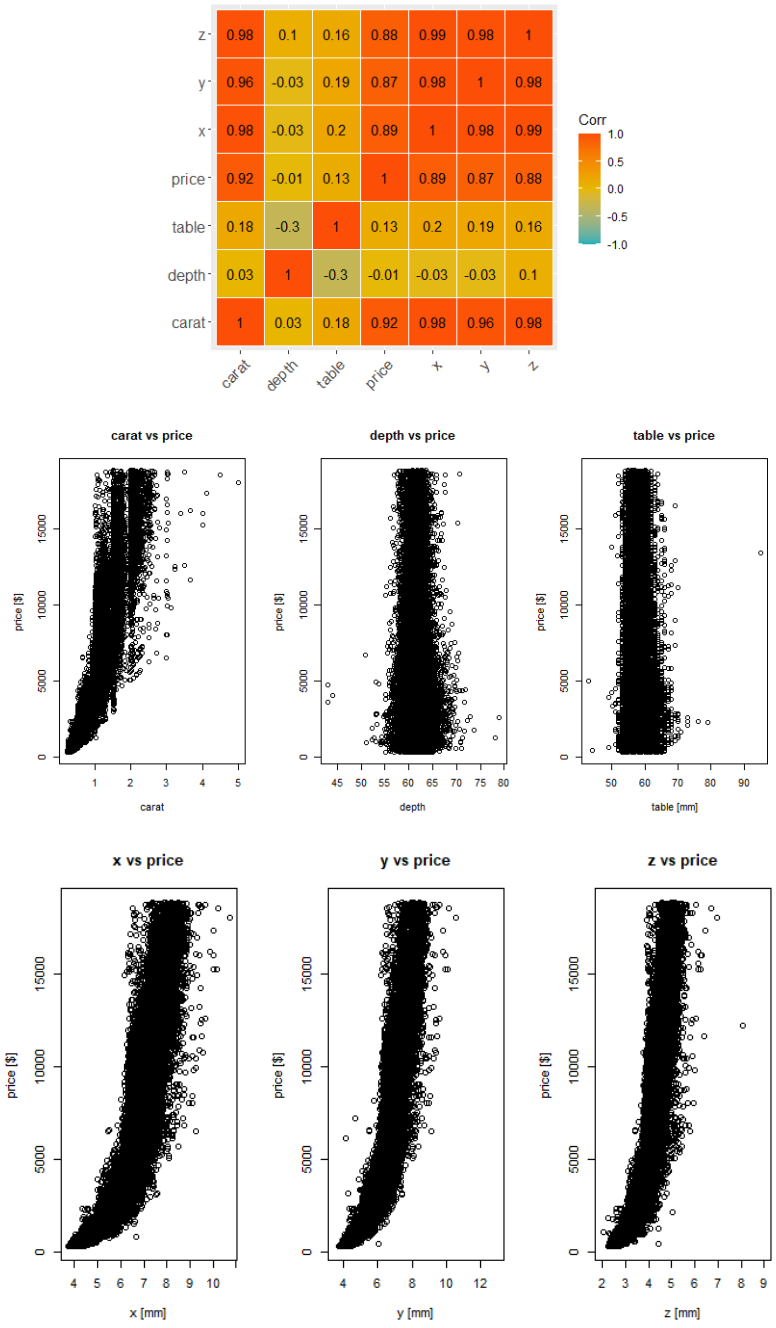


Figure 4: Price relationships: numerical features

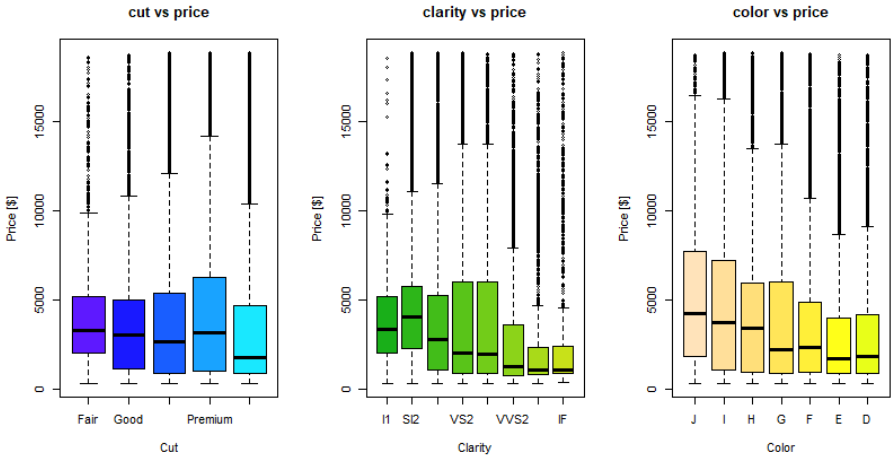


Figure 5: Price relationships: categorical features

maybe even more important than categorical ones (see Section 5 for a further evaluation of this hypothesis).

To have some fun we also tried to represent the joint relationship between *price* and x,y, z inside a 3D plot: Fig. 6 is taken from a interactive plot realized thanks to `library(plotly)`; to fully appreciate it, look at the **R** code.
What this graph basically tells us, is how the price seems to really increase linearly along all the three axis.

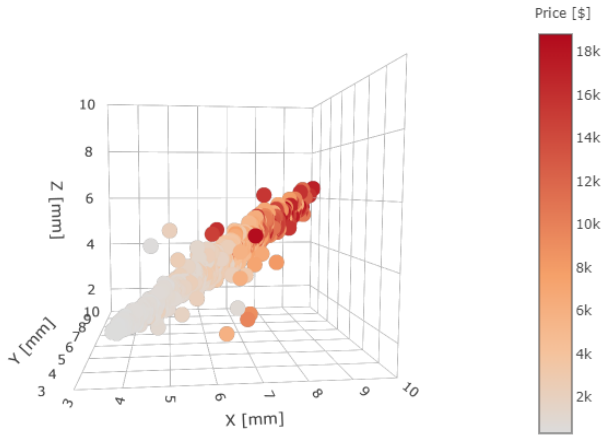


Figure 6: x, y, z vs price

2 Model Exploring

2.1 Naive Approach

As a first, brutal approach to our regression problem, we normalized our data (except for the target variable) and we tried to fit a linear model: the coefficients of the resulting one, listed in Table 4, suggest how increasing carat and the quality of cut, color and clarity, leads to an higher price of a diamond, while table and depth do not have such an impact on it; both of these aspects were expected, but what is harder to understand is the negative correlation between the dimensions of the diamonds and its price: indeed, from what we have seen so far, it seems very unlikely that larger diamonds cost less than smaller ones. The R^2 for this model, which measures how much our model can explain the variability of the data, is equal to 0.920, an impressive results. Going more in depth with the study of the model, we went through the analysis of residuals: they are clearly not uniformly distributed, especially for the largest predicted values (Fig. 7).

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-2574.9411	53.3615	-48.25	< 2e-16
carat	5460.8086	24.4621	223.24	< 2e-16
cutGood	574.3906	33.5265	17.13	< 2e-16
cutVery Good	717.3414	32.1782	22.29	< 2e-16
cutPremium	753.0040	32.1644	23.41	< 2e-16
cutIdeal	824.8785	33.3388	24.74	< 2e-16
colorI	905.8127	26.2719	34.48	< 2e-16
colorH	1396.3077	24.8354	56.22	< 2e-16
colorG	1898.8910	24.2635	78.26	< 2e-16
colorF	2108.6636	24.7638	85.15	< 2e-16
colorE	2167.1630	24.8638	87.16	< 2e-16
colorD	2376.0662	26.0711	91.14	< 2e-16
claritySI2	2716.6898	43.8015	62.02	< 2e-16
claritySI1	3677.7624	43.6168	84.32	< 2e-16
clarityVS2	4276.3465	43.8315	97.56	< 2e-16
clarityVS1	4587.0437	44.5193	103.03	< 2e-16
clarityVVS2	4951.8223	45.8192	108.07	< 2e-16
clarityVVS1	5004.0100	47.1197	106.20	< 2e-16
clarityIF	5340.2771	50.9678	104.78	< 2e-16
depth	-93.2116	6.6443	-14.03	< 2e-16
table	-59.0778	6.4910	-9.10	< 2e-16
x	-1232.1280	39.1716	-31.45	< 2e-16
y	29.5669	22.1718	1.33	0.1824
z	-80.7775	26.5836	-3.04	0.0024

Table 4: Results of naive linear model

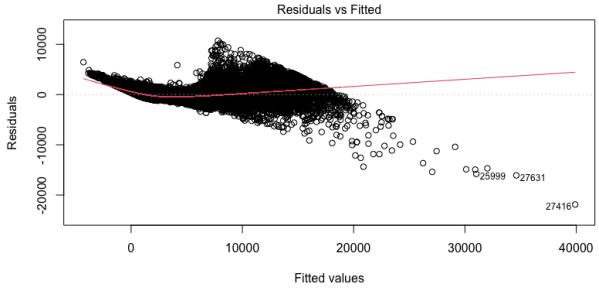


Figure 7: Plot of residuals for the naive approach

2.2 Variable Transformation

2.2.1 log(price)

From the previous analysis of the residuals, we decided to look for a transformation that could improve the quality of our model, removing that strange distribution shape in the residuals plot. Recalling the distribution of the *price* variable plotted above, we opted for a logarithmic transformation of our response variable, which gives our variable a more normal distribution (Fig. 8).

This time, the quality of the residual plots really improved (Fig 9), and all the variables now have an high significance expressed by the low p-values and the dimensions of the diamonds are positively correlated with price, as we expected (Table 5). Also the R^2 significantly increased to 0.981. But what is strange, in this case, is the coefficient for carat: indeed it is negative, while we know how important and positively correlated this feature is in estimating the

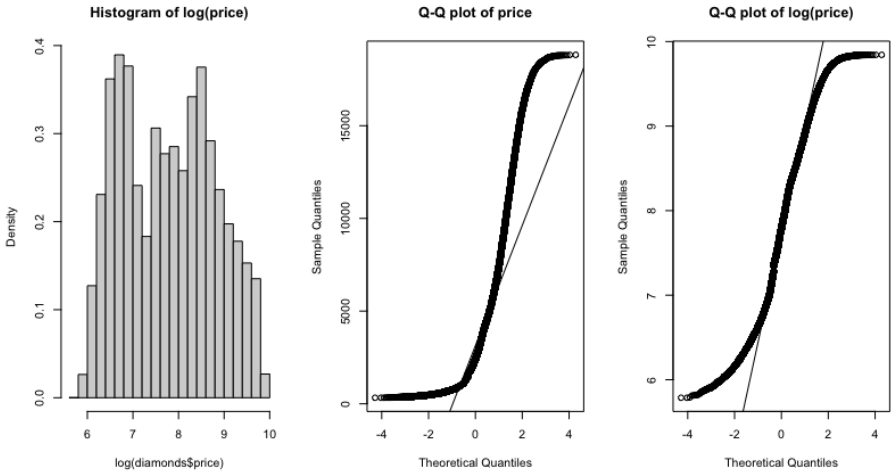


Figure 8

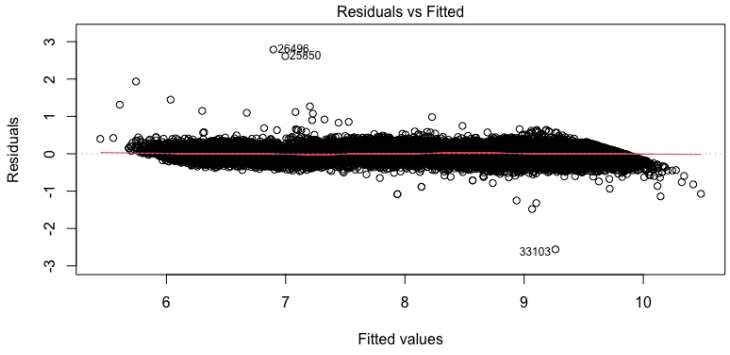


Figure 9: Plot of residuals log(*price*)

price of a diamond.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.6031	0.0066	1007.31	< 2e-16
carat	-0.4746	0.0030	-157.94	< 2e-16
cutGood	0.0992	0.0041	24.09	< 2e-16
cutVery Good	0.1376	0.0040	34.80	< 2e-16
cutPremium	0.1210	0.0040	30.63	< 2e-16
cutIdeal	0.1668	0.0041	40.73	< 2e-16
colorI	0.1362	0.0032	42.19	<2e-16
colorH	0.2554	0.0031	83.72	< 2e-16
colorG	0.3522	0.0030	118.16	< 2e-16
colorF	0.4182	0.0030	137.47	< 2e-16
colorE	0.4567	0.0031	149.51	< 2e-16
colorD	0.5136	0.0032	160.35	< 2e-16
claritySI2	0.4225	0.0054	78.53	< 2e-16
claritySI1	0.5903	0.0054	110.16	< 2e-16
clarityVS2	0.7374	0.0054	136.96	< 2e-16
clarityVS1	0.8065	0.0055	147.46	< 2e-16
clarityVVS2	0.9367	0.0056	166.42	< 2e-16
clarityVVS1	1.0081	0.0058	174.16	< 2e-16
clarityIF	1.1024	0.0063	176.06	< 2e-16
depth	0.0838	0.0008	102.62	<2e-16
table	0.0202	0.0008	25.35	< 2e-16
x	1.4966	0.0048	311.02	< 2e-16
y	0.0224	0.0027	8.21	< 2e-16
z	0.0404	0.0033	12.38	< 2e-16

Table 5: Results of linear model using log(*price*)

2.2.2 log(carat)

As a consequence of the last observation, we wondered whether to apply a transformation also to carat. Resembling what was done for *price*, we took into account the explorative plots, which seemed to suggest that applying a logarithm transformation to *carat* may have helped. Also in this case we obtained a better distribution, vaguely bi-modal and pretty similar to the logarithm of the price (Fig. 10).
Moreover, if we plot the two transformed variables one against the other, a linear relationship seems clear (Fig. 11).

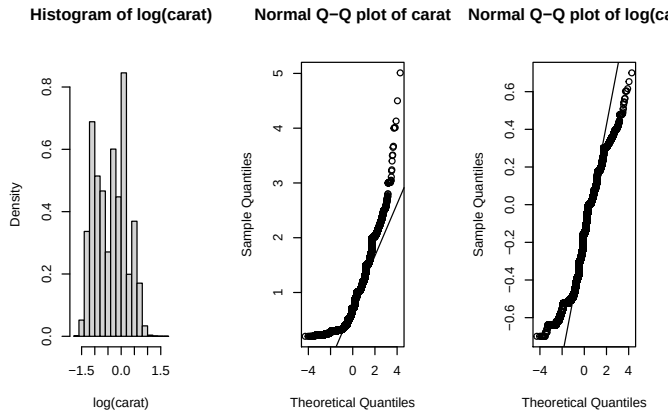


Figure 10

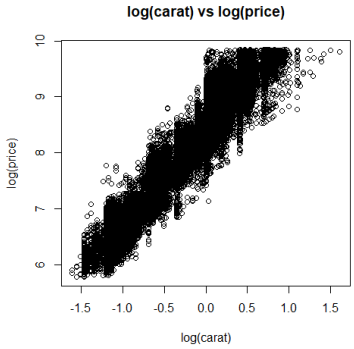


Figure 11: log(carat) vs log(price)

The new linear model fitted to our normalized data gave the results reported in Table 6

We had a slight increase of the R^2, now equal to 0.983 and finally we obtained a positive correlation between carats and price. Positive are also the correlations with all the most significant variables, while it seems like *table*, y and z are not so relevant for our task. For the former variable, this behaviour is understandable since, as stated above a bigger table does not necessarily increase the quality of a diamond; similarly for its height: it gives a better diamond only if it is well balanced with all the other dimensions. To explain the coefficient of y, right now we just report how much its value depend on x, as also the correlation matrix displayed: in fact, together they determine the shape of a diamond.

The residuals, not reported here for brevity, were not visibly affected by this transformation.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.5954	0.0063	1045.66	< 2e-16
carat	0.9947	0.0056	176.48	< 2e-16
cutGood	0.0800	0.0040	20.18	< 2e-16
cutVery Good	0.1158	0.0038	30.43	< 2e-16
cutPremium	0.1338	0.0038	35.14	< 2e-16
cutIdeal	0.1580	0.0039	40.07	< 2e-16
colorI	0.1398	0.0031	45.00	< 2e-16
colorH	0.2630	0.0029	89.64	< 2e-16
colorG	0.3561	0.0029	124.24	< 2e-16
colorF	0.4225	0.0029	144.48	< 2e-16
colorE	0.4622	0.0029	157.38	< 2e-16
colorD	0.5168	0.0031	167.76	< 2e-16
claritySI2	0.4316	0.0052	83.35	< 2e-16
claritySI1	0.5991	0.0052	116.23	< 2e-16
clarityVS2	0.7471	0.0052	144.21	< 2e-16
clarityVS1	0.8174	0.0053	155.34	< 2e-16
clarityVVS2	0.9503	0.0054	175.41	< 2e-16
clarityVVS1	1.0208	0.0056	183.17	< 2e-16
clarityIF	1.1154	0.0060	185.03	< 2e-16
depth	0.0047	0.0008	5.62	< 2e-16
table	0.0007	0.0008	0.90	0.3698
x	0.1076	0.0065	16.44	< 2e-16
y	-0.0014	0.0026	-0.55	0.5808
z	0.0014	0.0031	0.44	0.6605

Table 6: Results of linear model using log(*price*) and log(*carat*)

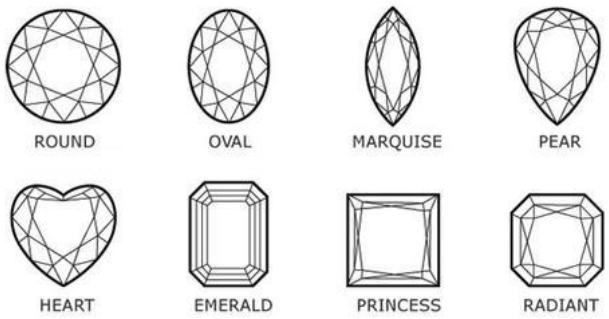


Figure 12: Different shape a diamond can assume.

2.2.3 A new variable: $r \coloneqq \frac{x}{y}$

As stated above and observed also in the correlation matrix, x and y are tightly correlated. Their values give us information about the dimensions of our diamonds, but they can also provide another piece of information: the shape of the diamond. The shape of a diamond, or more precisely its cut (not in the sense of the variable in our dataset, ed.), is another very important feature: the most common are round and princess cut diamonds but there are plenty others, as shown in Fig. 12. What is interesting to notice is we may roughly distinguish different shapes by the ratio between x and y: if it is equal to 1, we can expect to have a round or princess diamond, while another ratio may let think about some other shapes. Hence, we decided to encode this information into a new column of our dataset $r \coloneqq \frac{x}{y}$. We also tried to fit a new model to the normalized dataset containing also this feature and the results are reported in Table 7.

We can see that this addition does not overturn the precedent results, but still fix the possible issue with the sign of the coefficients of y and z and the significance of r seems pretty good. For what concern the R^2, it is the same

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.5846	0.0063	1041.87	< 2e-16
carat	1.0100	0.0057	177.56	< 2e-16
cutGood	0.0909	0.0040	22.71	< 2e-16
cutVery Good	0.1288	0.0039	33.32	< 2e-16
cutPremium	0.1368	0.0038	35.99	< 2e-16
cutIdeal	0.1671	0.0040	42.13	< 2e-16
colorI	0.1395	0.0031	45.03	< 2e-16
colorH	0.2617	0.0029	89.39	< 2e-16
colorG	0.3549	0.0029	124.15	< 2e-16
colorF	0.4214	0.0029	144.46	< 2e-16
colorE	0.4610	0.0029	157.37	< 2e-16
colorD	0.5155	0.0031	167.78	< 2e-16
claritySI2	0.4344	0.0052	84.09	< 2e-16
claritySI1	0.6022	0.0051	117.09	< 2e-16
clarityVS2	0.7504	0.0052	145.16	< 2e-16
clarityVS1	0.8212	0.0053	156.38	< 2e-16
clarityVVS2	0.9551	0.0054	176.57	< 2e-16
clarityVVS1	1.0258	0.0056	184.35	< 2e-16
clarityIF	1.1217	0.0060	186.27	< 2e-16
depth	0.0022	0.0008	2.59	0.0096
table	-0.0002	0.0008	-0.27	0.7908
x	0.0507	0.0073	6.94	< 2e-16
y	0.0350	0.0033	10.46	< 2e-16
z	0.0065	0.0032	2.07	0.0380
r	0.0143	0.0008	17.51	< 2e-16

Table 7: Results of linear model using $\log(price)$, $\log(carat)$ and r

obtained previously, 0.983.

3 Feature Selection

So far we have dealt with our models through some naive approaches, but now we really want to understand which features seem to be really significant for our purpose of price prediction. In order to do it, we take as a reference the last introduced model, which takes into account the logarithm of price and carat and also the new variable r, and we apply methods for best subset selection based mainly on the *Adjusted* R^2*,* the *Mallow's* C_p *coefficient* and the *Bayesian Inference Criterion* (BIC) criterion, but for some methods we will also consider the *Akaike Information Criterion* (AIC).

The basic idea behind model selection is, given the different level of significance of the variables, try to understand which of them contributes substantially to our model in order to simplify it as far as it can still perform well, decreasing overfitting and increasing the generalization of the model. To measure such contribution, we focus on different metrics:

- Adjusted R^2: it is equal to $1-\frac{\frac{RSS}{n-d-1}}{\frac{TSS}{n-1}}$ $(\in [0,1])$, where RSS is the residual sum of squares, TSS is the total sum of squares, n is the total number of features and d is the number of feature selected for the specific model. Our goal is to find the subset of d features that maximizes this quantity.
- Mallow's C_p coefficient: it is computed as $\frac{1}{n}\left(RSS + 2d\hat{\sigma}^2\right)$, where $\hat{\sigma}^2$ is an estimate of the residual variance based on the full model containing all the predictors. Our goal, in this case, is to minimize this quantity.
- BIC: the Bayesian Information Criterion is computed as $-2\ell(\hat{\theta}) + d\log(n)$, with $\ell(\hat{\theta})$ the maximized value of the log-likelihood function for the estimated model. In our specific case, since we are assuming that our errors are normally distributed, the above formula becomes $n\log\left(\frac{RSS}{n}\right) + d\log(n)$. Like for the Mallow's C_p, we aim to minimize it, but BIC will generally suggest models with a smaller number of features because the logarithm in the second term of the definition penalizes high number of features.
- AIC: the Akaike Information Criterion is computed as $-2\ell(\hat{\theta}) + 2d$ but, in the case of normally distributed errors it becomes $n\log\left(\frac{RSS}{n}\right) + 2d$. AIC is very similar to BIC, but it tends to penalize less the complexity of our model. It may also be proved that it is equivalent to the Mallow's C_p, in the sense that they both suggest the same complexity of the models.

Notice that the introduction of these new criteria is due also to the fact that the R^2 and the MSE do not provide a reliable feature selection criterion since they typically privilege the largest model, while we want to achieve a good trade-off between fit and simplicity.

3.1 Exhaustive Best Subset Selection

The trivial approach to solve the best subset selection problem is to try fitting all the models with all the possible combinations of features and see which of

these optimizes the above metrics. Clearly, this approach is unfeasible for a large numbers of predictors, since the number of models to fit grows exponentially with the number of features. In our specific case, anyway, we did not face this issue, thing that allowed us to easily solve our problem via the `regsubsets` command with `method = "exhaustive"`.

By the above plots (Fig. 13) we can see, as we expected, that the BIC suggests a model with one feature less than the Mallow's C_p and the Adjusted R^2: in particular, the C_p and Adjusted R^2 proposes the model with all features except for *table*, while BIC suggests to exclude also *z*. Now, to decide whether to pick as the reduced model the one suggested by C_p or BIC, we observed that, printing the values of the Mallow's C_p coefficient for the different subsets of variables, we can see that the one with 22 features, that is exactly the same proposed according to the BIC criterion, has a very slightly higher value than the best one and moreover, fitting the model with 23 features we can see how the significance of *z* is not that high; as a consequence, and also for privileging the simplicity of the model, we chose as best model the one without *table* and *z*.

For the sake of curiosity, we tried to perform best subset selection also using

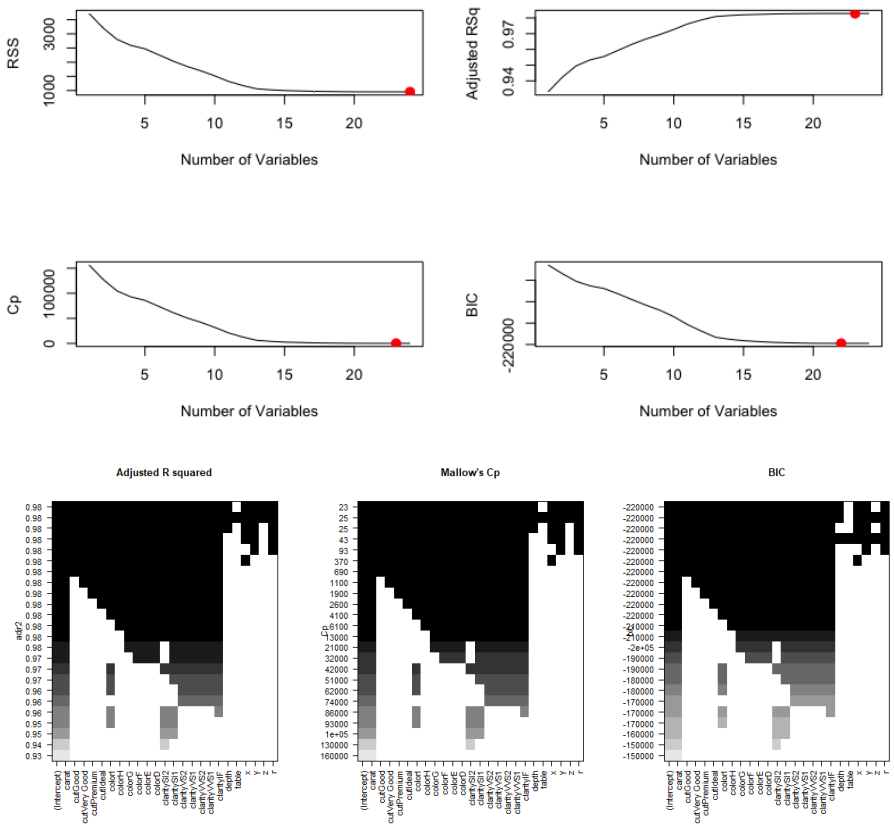


Figure 13: Best subset of features for each criterion

the *Greedy Search* approach: instead of analyzing all the possible models for all the combinations of features, we started either from the full or the null model and we progressively removed or added the feature that guarantees the local optimum for our optimization criterion. This approach does not guarantee the find of a global optimum, but it requires the confrontation of a much smaller number of models compared to the exhaustive subset selection and hence in general it is a more scalable approach[2].

3.2 Forward Selection

The first implementation of the Greedy approach is the *Greedy Forward Selection*: we started from the null model, i.e. the one with no predictors, and, at each iteration, we added the feature that minimizes the selected criterion (in this case we considered only the Mallow's C_p and the BIC). This procedure led to the same models suggested by the *exhaustive* selection. The limited number of predictors helped the local optimum obtained via Greedy search to coincide with the global optimum.

3.3 Backward Selection

The inverse approach if compared to Forward Selection is the Backward Selection: it starts from the model with all the possible predictors and remove one variable at a time, trying to optimize our criterion or, in the worst case, to worsen them as little as possible. Also this time the results coincided with those from the exhaustive search for both criteria.

3.4 Both Selection

Both Subset Selection is basically a combination of forward and backward selection: starting from the null model we sequentially add the most contributing predictors and, after each addition, we remove those variables that do not provide any improvement to the model fit. Also the results of this procedure are aligned to the previous ones.

3.5 Linear Model

Once *Feature Selection* was performed, we used the obtained results to build a linear model.
This time we randomly divided the dataset into two parts, 70% of it was used as a training subset while the remaining as test subset: doing so we can compute the *Mean Square Error* on the latter, which will be, from now on, our main metric to compare different models given that our goal is to predict prices of diamonds.

In Table 8 we report the results obtained using both just the variables selected by best subset selection:

As we can see now, all the variables included have a p-value$< 10^{-6}$: this implies a meaningful relationship is likely to connect the predictors and the

[2]We do not report detailed results of this different approach since the model suggested are the same obtained with the exhaustive selection. For deeper insight, once again, take a look at the **R** code

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.5767	0.0070	943.50	< 2e-16
carat	1.0097	0.0063	160.08	< 2e-16
cutGood	0.0931	0.0044	21.11	< 2e-16
cutVery Good	0.1292	0.0042	30.73	< 2e-16
cutPremium	0.1349	0.0041	32.58	< 2e-16
cutIdeal	0.1666	0.0041	40.65	< 2e-16
colorI	0.1392	0.0035	40.08	< 2e-16
colorH	0.2606	0.0033	79.39	< 2e-16
colorG	0.3555	0.0032	111.04	< 2e-16
colorF	0.4210	0.0033	128.93	< 2e-16
colorE	0.4607	0.0033	140.52	< 2e-16
colorD	0.5146	0.0034	149.61	< 2e-16
claritySI2	0.4418	0.0058	76.29	< 2e-16
claritySI1	0.6106	0.0058	105.87	< 2e-16
clarityVS2	0.7605	0.0058	131.18	< 2e-16
clarityVS1	0.8304	0.0059	140.98	< 2e-16
clarityVVS2	0.9661	0.0061	159.16	< 2e-16
clarityVVS1	1.0339	0.0062	165.89	< 2e-16
clarityIF	1.1313	0.0067	167.98	< 2e-16
depth	0.0034	0.0008	4.45	9e-06
x	0.0514	0.0075	6.90	5e-12
y	0.0413	0.0035	11.74	< 2e-16
r	0.0178	0.0010	18.13	< 2e-16

Table 8: Coefficients for linear model after Feature Selection.

response variable. In addition to that, all the coefficients has become positive and all the $4C$ has obtained the importance we expected. In this case, the MAE and RMSE[3] are reported below:

	MAE	RMSE
Linear model	418	874

3.5.1 Leverage & Cook's points

If we look at the residual plots for the reduced model and in particular at Fig. 15), we can see that there are some point of high leverage, that are points that could have a large influence when fitting the model and, in some cases they could lead to the identification of outliers. Typically, they are considered points of high leverage those that have a leverage greater than the double of the average; anyway, in this dataset, because of the high variability of the variables, we decided to consider just those points with a leverage over 0.01, that is much

[3]Recall that MAE $= \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|$ and RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}$. From now on, every time an error is computed we will perform the inverse transformation applying the exponential on *price*, doing so the MAE and RMSE are measured in actual dollars and are comparable to the real prices.

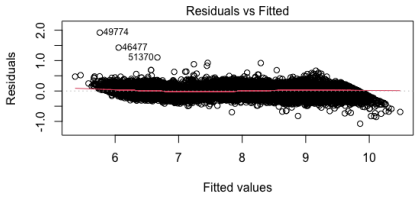


Figure 14

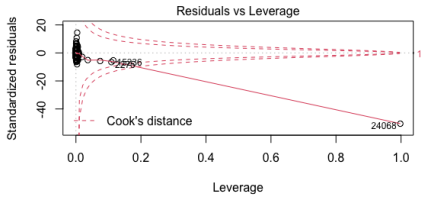


Figure 15

more than the double of the average (0.0006). In this way we identified six points, but to really access the influence of them on the model, we need not just to consider their leverage but also if they have high residuals. In order to do this influence, therefore, we computed the *Cook's Distance* of the points in our dataset, that is defined as $\frac{1}{p} r_i \frac{h_i}{1-h_i}$ where p is the number of predictors, r_i is the residual for the i-th point and h_i its leverage. An observation is generally considered as influential when its Cook's Distance is larger the $\frac{4}{n}$, with n the length of the dataset. Performing this computations, we retrieved only one instance that can be considered influential for our model.

Considering the removal of this point could be an interesting way of proceeding in further developments of our model. Anyway, since it is just one over a huge dataset, we decided to focus on other methods for optimizing our model rather than proceeding in this direction.

4 Shrinkage Methods

If best subset selection methods directly control model complexity and variance by forcing some coefficients to be zero, *Shrinkage Methods* try to shrink in an automated way some coefficient toward zero, in order to increase the generalization of our model and, in some cases, to bring in specific properties (convexity for the Ridge Regression and sparsity for Lasso Regression).

4.1 Ridge Regression

Ridge regression applies a quadratic shrinking to our parameters by adding to the objective function for linear models a term with the squared $\mathcal{L}^2$ norm of the coefficients scaled by a properly chosen constant λ: it is a hyperparameter of our model that has to be tuned in order to decide the amount of shrinkage to bring in.

The importance of this additional quadratic term is capital also by the optimization point of view: indeed, it allows our objective function to become

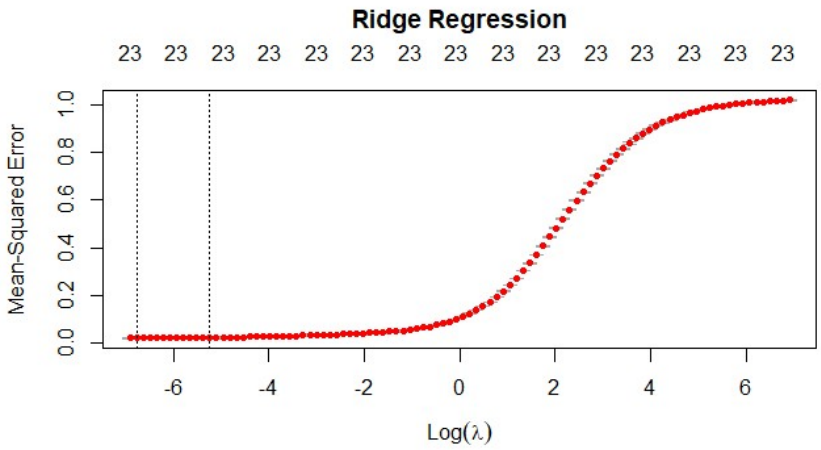


Figure 16: MSE for Ridge Regression varying λ

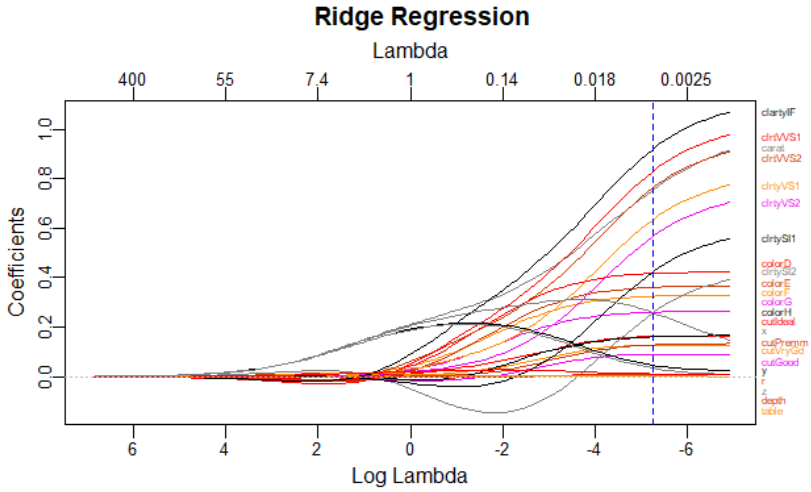


Figure 17: Coefficient Shrinking Ridge Regression

convex, a property that really helps finding an optimal solution.

We applied this method to a linear model with the logarithm of the price as target variable and all the other variables as predictors (we recall that we are considering the logarithm of carat and the r variable). In order to select the proper λ, we chose within a family of possible values ranging in $\left[10^{-3}, 10^{3}\right]$ by applying the *10-fold cross validation* to a subset of our dataset that was selected as a training set. In Fig. 16 we can see how the estimated mean squared error changes depending on $\log(\lambda)$ and we chose as best λ 0.0053: it is not the one that gave the smallest estimated error, but the least complex one within one standard error (highlighted in the figure by the right most dashed line). To visualize the effect of the shrinkage coefficient, have a look at Fig. 17: we can see how the different coefficients change their values according to different λ values; in particular, the selected one correspond to the vertical dashed line.

Once we identified the optimal value of our hyperparameter we fitted a model over the whole training set and then we used it to predict the price on the test set:

	MAE	RMSE
Ridge regression	472	1109

4.2 Lasso Regression

The other shrinkage method we applied is the Lasso Regression. This one differs from the Ridge Regression because of the regularization term it adds to our objective function: instead of the $\mathcal{L}^2$ squared norm, this time we add the $\mathcal{L}^1$ norm scaled by the shrinkage coefficient λ.

The underlying idea of this method is the same of the previous one, we add a term in order to penalize the coefficients in order to keep them toward zero. Anyway, this time the objective function is not quadratic and hence convexity is

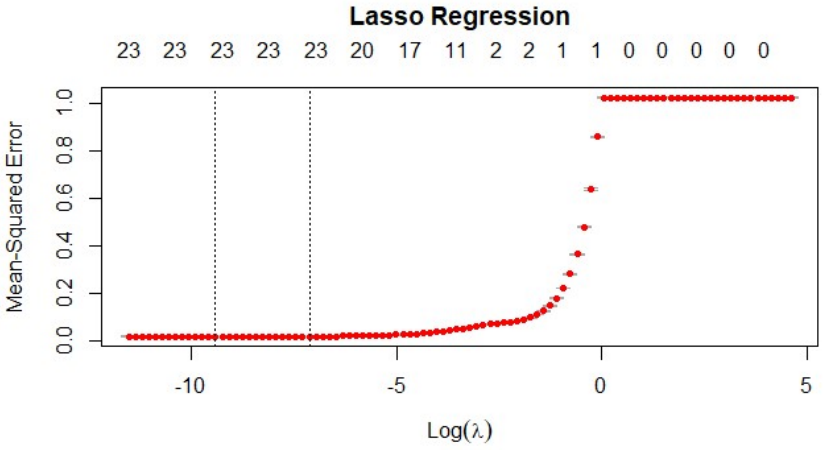


Figure 18: MSE for Lasso Regression varying λ

not guaranteed, but, with this penalization, we obtain a bigger shrinking toward zero, which generally leads to a sparser final model.

The applied procedure was the same described for the Ridge Regression: we picked a proper subset of values for λ (this time $\lambda \in \left[10^{-5}, 10^{2}\right]$) and we applied the 10-fold cross validation. We selected the proper value for $\lambda = 0.0008$ according to the "one-standard-error" rule and we predicted on the test set, this time obtaining:

	MAE	RMSE
Lasso regression	448	953

Something that is really important to notice is how the different values of λ affect the parameters: as we said, Lasso Regression tends to force a larger

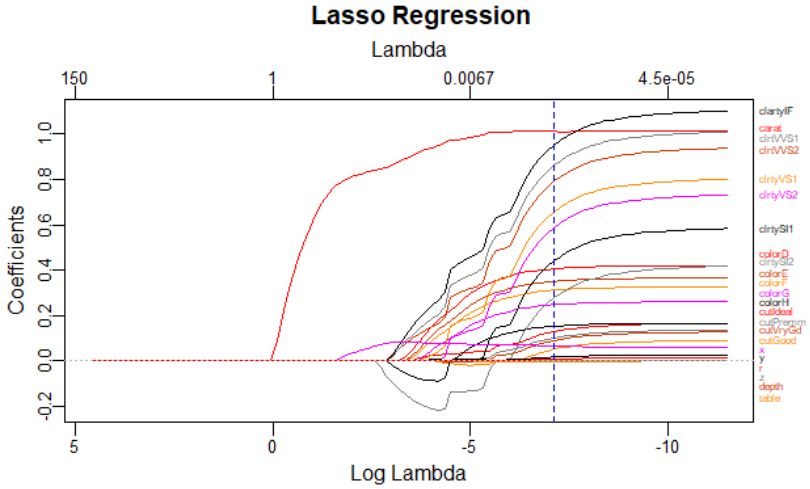


Figure 19: Coefficient Shrinking Lasso Regression

number of parameters toward zero and, indeed, in Fig. 19 we can see that the parameters start increasing for smaller values of λ, if compared to the Fig. 17, and there are more coefficients keeping their values around zero.

5 Principal Component Analysis (PCA)

In this last section we try a different approach from before: instead of selecting a subset of the variables, we decided to keep them all (normalized as before), initially without using any variable transformation beside the addition of r. Starting from here, we applied PCA, an algorithm for dimensionality reduction: by doing so we hoped to summarize the information contained in all our variables in a smaller number of components obtained as a linear combination of the original ones.
Since PCA can be applied only to numerical features, we worked in two different settings: firstly we used a dataset containing only numerical predictors (also to question the hypothesis on the relevance of categorical features introduced at the beginning of this paper) and we computed the MAE and RMSE on the test set (using the same model defined in Section 3.5). Then, we applied PCA to the numerical variables and we combined the resulting ones with the categorical varibles obtaining a dataset on which we fitted another linear model.

Following what we have done previously, we attempted also to perform the same computations applying the log transformation to the feature *carat*, since we know from theory that a transformation that increase the linear relationship between data and the predicted variable may help PCA.

As a response variable for all the linear models that we will present soon, we decided to keep log(*price*) as before, since even in this case the residuals were distributed more evenly.

5.1 PCA without log(carat)

From the `screeplot` (Fig. 20) we can see how the information encoded in our 7 numerical predictors can be mainly summarized by the first 4 principal components.

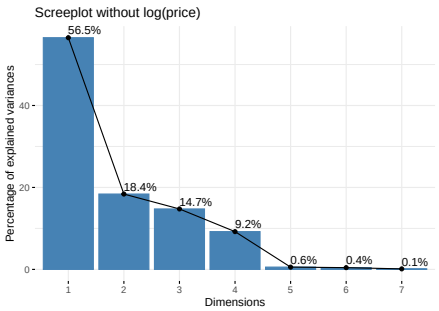


Figure 20

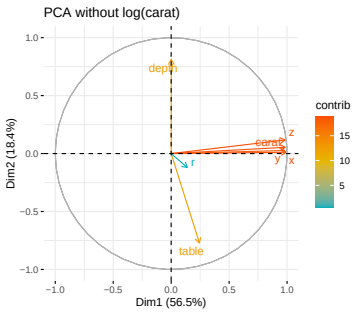


Figure 21

Once these 4 components were selected, we created a new linear model obtaining the following coefficients (Table 9):

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	7.7876	0.0015	5227.91	< 2e-16
Comp.1	0.4839	0.0007	647.05	< 2e-16
Comp.2	0.0408	0.0013	31.21	< 2e-16
Comp.3	0.0223	0.0015	14.82	< 2e-16
Comp.4	0.0765	0.0019	41.30	< 2e-16

Table 9: Summary of linear model with PCA

This linear model yields:

	MAE	RMSE
PCA	1245	4197

Apparently we had a worsening of the performance if compared to the previous models, a fact to be probably imputed to the absence of categorical predictors. Indeed, the addition of the categorical variables yielded the coefficients stored in Table 10.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.4822	0.0114	566.88	< 2e-16
Comp.1	0.5467	0.0006	896.73	< 2e-16
Comp.2	0.0410	0.0010	40.83	< 2e-16
Comp.3	-0.0147	0.0011	-12.79	< 2e-16
Comp.4	0.0392	0.0015	25.51	< 2e-16
cutGood	0.1138	0.0071	15.92	< 2e-16
cutVery Good	0.1377	0.0069	20.01	< 2e-16
cutPremium	0.0935	0.0068	13.79	< 2e-16
cutIdeal	0.1624	0.0071	22.95	< 2e-16
colorI	0.1538	0.0056	27.30	< 2e-16
colorH	0.2919	0.0053	54.90	< 2e-16
colorG	0.4148	0.0052	80.11	< 2e-16
colorF	0.4881	0.0053	92.49	< 2e-16
colorE	0.5084	0.0053	95.85	< 2e-16
colorD	0.5658	0.0056	101.66	< 2e-16
claritySI2	0.5068	0.0094	53.98	< 2e-16
claritySI1	0.6955	0.0093	74.43	< 2e-16
clarityVS2	0.8223	0.0094	87.48	< 2e-16
clarityVS1	0.8901	0.0096	93.19	< 2e-16
clarityVVS2	0.9909	0.0098	100.63	< 2e-16
clarityVVS1	1.0352	0.0101	102.40	< 2e-16
clarityIF	1.1277	0.0109	103.25	< 2e-16

Table 10: Summary of linear model with PCA with categorical variables

This time the performance really improved:

	MAE	RMSE
PCA + cat. vars.	875	3006

This confirms that the information contained in the categorical variables is precious, according to the relevance that the previous linear models gave to them and contrary to what we guessed at the end of Section 1.1.

5.2 PCA with log(carat)

The addition of the logarithmic transformation to *carat* did not bring a big improvement, but still it is worth mentioning it. Again the first four principal components explain more then the 99% of data variability, so we kept only these ones (Fig 22).

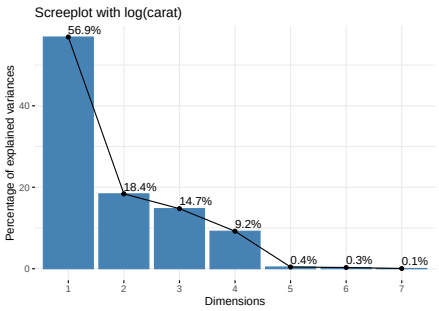


Figure 22

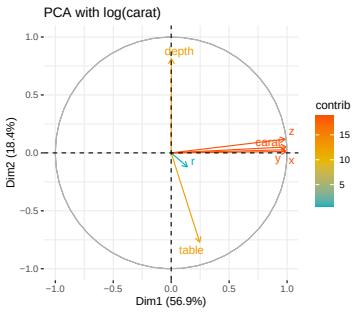


Figure 23

A new linear model has been created with the following results (Table 11):

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	7.7877	0.0013	5777.66	< 2e-16
Comp.1	0.4868	0.0007	721.46	< 2e-16
Comp.2	0.0428	0.0012	36.20	< 2e-16
Comp.3	0.0231	0.0014	17.01	< 2e-16
Comp.4	0.0809	0.0017	48.31	< 2e-16

Table 11: Summary of linear model with PCA with log(*carat*)

MAE and RMSE are the following:

	MAE	RMSE
PCA with log(*carat*)	1252	3891

We can notice a small improvement with respect to the the same model without the log transformation. Once again we added all the categorical variables (Table 12):

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.5150	0.0089	729.95	< 2e-16
Comp.1	0.5500	0.0005	1158.63	< 2e-16
Comp.2	0.0435	0.0008	55.54	< 2e-16
Comp.3	-0.0146	0.0009	-16.29	< 2e-16
Comp.4	0.0413	0.0012	34.34	< 2e-16
cutGood	0.1224	0.0056	21.91	< 2e-16
cutVery Good	0.1543	0.0054	28.70	< 2e-16
cutPremium	0.1170	0.0053	22.08	< 2e-16
cutIdeal	0.1795	0.0055	32.48	< 2e-16
colorI	0.1460	0.0044	33.17	< 2e-16
colorH	0.2731	0.0042	65.77	< 2e-16
colorG	0.3834	0.0040	94.89	< 2e-16
colorF	0.4529	0.0041	110.01	< 2e-16
colorE	0.4829	0.0041	116.71	< 2e-16
colorD	0.5401	0.0043	124.40	< 2e-16
claritySI2	0.4742	0.0073	64.66	< 2e-16
claritySI1	0.6548	0.0073	89.74	< 2e-16
clarityVS2	0.7943	0.0073	108.22	< 2e-16
clarityVS1	0.8627	0.0075	115.68	< 2e-16
clarityVVS2	0.9840	0.0077	127.95	< 2e-16
clarityVVS1	1.0422	0.0079	131.98	< 2e-16
clarityIF	1.1385	0.0085	133.43	< 2e-16

Table 12: Summary of linear model with PCA with log(*carat*) and categorical variables

This addition led to:

	MAE	RMSE
PCA with log(*carat*) + cat. vars.	837	3004

In this way we obtained the smallest MAE among all PCA models, highlighting, once again, the importance of transforming the *carat* variable and considering categorical ones.

6 Conclusion

Summarizing our work, in this project we tried to build a robust linear model to correctly predict the price of diamonds given their main features. We followed three different approaches:

1. Applying Feature Selection in several ways and then using the results to create a standard linear model;
2. Using the so called Shrinkage Methods to automatically shrink the less important feature coefficients;
3. Summarizing all our numerical features in a smaller number of components through PCA.

In Table 13 we report all the MAE and RSME obtained:

	MAE	RMSE
Linear model	418	874
Ridge regression	472	1109
Lasso regression	448	953
PCA	1245	4197
PCA + cat. vars.	875	3006
PCA with log(*carat*)	989	2636
PCA with log(*carat*) + cat. vars.	564	1541

Table 13: MAE and RMSE summary

As we can see, despite its simplicity, the Linear model is still highly competitive with more structured methods like Ridge Regressor and Lasso Regressor, since it is the best performing one. Anyway, their performance are very close one to the other and still pretty good. In fact, if we considered our best performing model compared to the naive linear model in 2.1 fitted on the same training set, we ccould see that the MAE would have dropped significantly from 742 to 418, a clear sign that our analysis went in the right direction.

Definitely, our best models performed better than PCA: indeed, PCA is first of all a technique for dimensionality reduction and so, selecting principal components, we basically discard part of the information in our data, reducing with high probability the quality of our prediction. In this sense, it is probably not the best method applicable for a prediction task, especially when we have a limited number (10) of numerical features as for our dataset: it may instead be fundamental with a high number of predictors, in order to select a limited

number of components that captures the ongoing dynamics and reduce the computational effort associated with big data.

One of the reasons why our simple linear model is as effective as considering Ridge and Lasso regression could be the fact that, because of the low number of features, it was possible to perform the exhaustive feature selection that returned the very simplest model among all the possible. If we had a number of predictors such that this exhaustive search was not feasible, we may have not found a global optimum and so, in that case, ridge and lasso may have had to be preferred.

To proceed toward possibly better predictors, a possibility may be to consider other types of non-linear models or to deepen the analysis of high leverage points we just sketched in Section 3.5.1. Applying more sophisticated Machine Learning tools or even Deep Learning models can of course be another viable option.