Modern Methods of Data Analysis

Home project

Anosov Roman, Badalyan Shavarsh, Garmider Petr, Ruziev Jamshid

December 6, 2020

Contents

1 Dataset description 2

2 K-means 2

3 Bootstrap 3

4 Contingency table 5

5 PCA/SVD 7

6 Correlation analysis 9

https://github.com/PetrGarm/mm2020 — full code to replicate presented results.

1 Dataset description

The diamond market, just like the market for other rare and precious commodities, is subject to some degree of subjectivity in pricing ([1]). The aim of our research is to investigate the relationship between the physical properties of diamonds and their prices, to understand the extent to which diamond prices are determined by physical properties.

Original dataset is a dataset containing the prices and other attributes of almost 54,000 diamonds. Each diamond, i.e. row in data frame, is described with 10 variables. The variables are as follows (range min–max is indicated for real value variables, set of possible values (and order, if set) is indicated for categorical variables):

- *price* — price in US dollars ($326–$18,823)
- *carat* — weight of the diamond (0.2–5.01)
- *cut* — quality of the cut (Fair, Good, Very Good, Premium, Ideal)
- *color* — diamond colour, from D (best) to J (worst)
- *clarity* — a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
- x — length in mm (0–10.74)
- y — width in mm (0–58.9)
- z — depth in mm (0–31.8)
- *depth* — total depth percentage = $\frac{z}{mean(x,y)} = \frac{2z}{x+y}$ (43–79)
- *table* — width of top of diamond relative to the widest point (43–95)

Original dataset can be found at https://ggplot2.tidyverse.org/reference/diamonds.html

In the context of the current research only 500 random samples from the original dataset were selected. The set of features was limited to $\{price, carat, cut, x, y, z, depth\}$. The main problem, dictated by the data and solved by us, is to study the influence of factors on the diamond price, so *price* was chosen as a target variable in the study.

2 K-means

For K-means clustering procedure we decided to use 3 features: carat (weight of the diamond), depth (total depth percentage) and price (in US dollars). We believe that these features are meaningful enough, so KMeans can derive different clusters using chosen features. We used exact KMeans from sklearn.cluster with proper parameters commented in code section in github.

	carat	depth	price
#1 (Extra Premium)	122.37	-0.25	262.91
#2 (Quite cheap)	4.45	3.13	-14.31
#3 (Premium)	46.16	-0.08	62.38
#4 (The cheapest)	-41.87	0.17	-65.82
#5 (Middle price)	-17.78	-3.42	-36.54

Table 1: Relative difference of feature using K=5

On Table 1, one can see kmeans result for K=5. As one can see, cluster 1 may be named "Extra Premium" diamonds cluster with high relative price and carat features. Cluster 4 is "The Cheapiest" cluster with the lowest carat feature. The other left form "Middle" clusters where each may also be interpreted as "premium",

[1] https://scielo.conicyt.cl/pdf/jtaer/v13n2/0718-1876-jtaer-13-02-00103.pdf

"middle price" and "quite cheap" diamonds. Carat feature supports this argument, as there is high correlation between carat and price. Unfortunately, depth feature turned out to bee not important at all, as absolute relative difference of that feature does not exceed 5 %.

	carat	depth	price
Cluster #1	131.72	-0.23	285.87
Cluster #2	81.24	0.59	135.60
Cluster #3	-48.38	1.50	-72.29
Cluster #4	37.39	-3.85	48.53
Cluster #5	33.63	-0.14	38.52
Cluster #6	-42.09	-3.13	-67.09
Cluster #7	-1.82	6.12	-34.12
Cluster #8	17.61	2.37	6.40
Cluster #9	-38.61	-0.40	-62.70

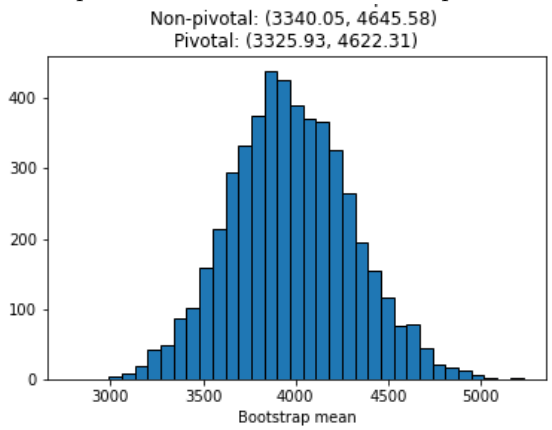
Table 2: Relative difference of feature using K=9

On Table 2, one can see kmeans result for K=9. We believe that this specification made results worse, as there are number of clusters that look quite similar: 4 and 5; 3, 6 and 9. So there is evidence that there are rather less than 9 clusters and also, interpretation of those clusters is quite similiar to the results of 5-means.

3 Bootstrap

For this section, only simple np.random.choice method was used that return sample with return of indicated size. The rest is bootstrap theory.

Figure 1: Bootstrap 95 % confidence intervals for price feature mean value



In order to compute CI a bootstrap procedure was used, with 5000 iterations and 150 sample size on each iteration. As was expected, distribution of bootstrapped mean (see Figure 1) is quite similar to Gaussian distribution, so it is not surprising that pivotal and non-pivotal confidence intervals are quite similar. Build intervals cover true mean of a diamond with probability of 95 % and do not cover it with probability 5 %.

As we see on Figure 2 pivotal and non-pivotal intervals are quite similar again, and price of a diamonds is for sure statistically different between these two clusters. That result makes sense, since the main difference between them is price. Therefore, clustering algorithm returned interpretable clusters.

As expected from Table 1 cluster 1 diamonds price is larger 262 % in average.

`Grand mean of diamonds price = 3977.42`

`Expected difference for the first cluster and grand mean =` $2.62 \times 3977.52 = 10421.1$

As wee see from Figure 3 expected difference lies in bootstrapped confidence intervals. Once again, worth to mention that estimated confidence intervals using pivotal and non-pivotal method give nearly the same results.

Figure 2: Bootstrap 95 % confidence intervals for price difference between first and second clusters diamonds

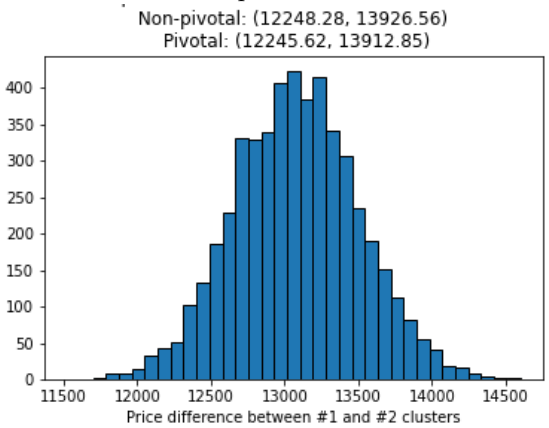
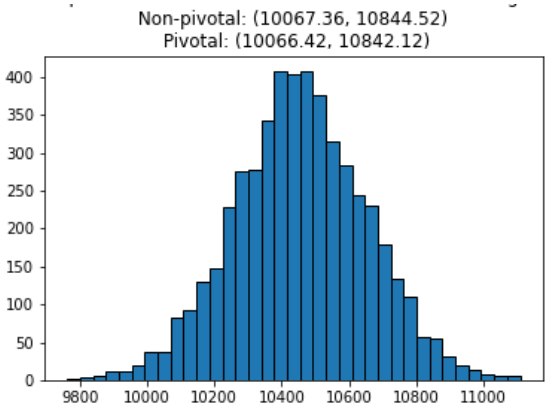


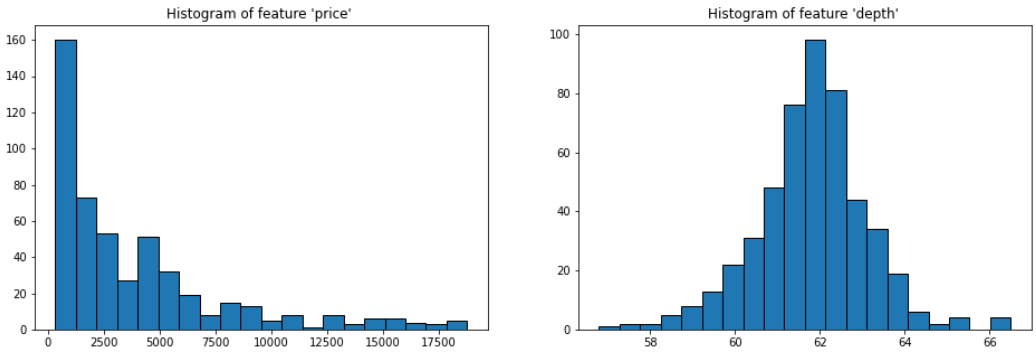
Figure 3: Bootstrap 95 % confidence intervals for price feature difference between the first cluster and grand-mean



4 Contingency table

At Figure 4 one can see histograms for price and depth features. [0, 2500), [2500, 7500), [7500, 12500) [12500, 20000] and [0, 61), [61, 63), [64, 66.5] boundaries is used to categorize 'price' and 'depth' features. After categorizing selected features, we get new features 'price categorize' (3 categories) and 'depth categorize' (3 categories). Feature 'cut' include 3 categories.

Figure 4:



cut price_categorize	Fair	Good	Ideal	Premium	Very Good
(0, 2500]	0.454545	0.411765	0.614130	0.430769	0.475806
(2500, 10000]	0.545455	0.509804	0.277174	0.453846	0.451613
(10000, 20000]	0.000000	0.078431	0.108696	0.115385	0.072581

Table 3: Conditional frequency table for price category and cut feature

depth category price category	(0.0, 61.0]	(61.0, 63.0]	(63.0, 66.5]
(0, 2500]	0.474576	0.554098	0.376623
(2500, 10000]	0.398305	0.357377	0.545455
(10000, 20000]	0.127119	0.088525	0.077922

Table 4: Conditional frequency table for price category and depth feature

As Figure 4 shows, price feature may have lognormal distribution, that is really common case for such called "money" features. Depth feature at look has signs of gaussian distribution with mean at 62.

One can notice from tables 3 and 4 that most of the diamonds presented in the dataset are of ideal quality, even those of a low price category. Moreover, distribution for cut feature for low and high price categories look the same. However, there is an interesting fact, that most of "low price" diamonds are of ideal cut. This may be due to the fact that we are actually working with small sample of data (500 observations from original 10 000). Price Fair cut diomonds commonly is from 2500 to 10000.

What for the depth category feature, its distribution seems do not change that much over different price categories.

cut price category	Fair	Good	Ideal	Premium	Very Good
(0, 2500]	0.36	0.24	0.84	0.29	0.43
(2500, 10000]	0.64	0.53	-0.17	0.36	0.35
(10000, 20000]	-1.00	-0.76	-0.67	-0.65	-0.78

Table 5: Quetelet index table for price category and cut feature

depth category price category	(0.0, 61.0]	(61.0, 63.0]	(63.0, 66.5]
(0, 2500]	0.42	0.66	0.13
(2500, 10000]	0.19	0.07	0.64
(10000, 20000]	-0.62	-0.73	-0.77

Table 6: Quetelet index table for price category and depth feature

Another way to look at category dependency is Quetelet index that shows relative difference for event A probability under event B. Overall this information supports words said earlier, may show however, that depth category distribution differs from different price ranges in terms of quetelet index.

cut price category	Fair	Good	Ideal	Premium	Very Good
(0, 2500]	0.00360	0.01008	0.18984	0.03248	0.05074
(2500, 10000]	0.00768	0.02756	-0.01734	0.04248	0.03920
(10000, 20000]	-0.00000	-0.00608	-0.02680	-0.01950	-0.01404

Table 7: Average Quetelet index table for price category and cut feature

depth category price category	(0.0, 61.0]	(61.0, 63.0]	(63.0, 66.5]
(0, 2500]	0.04704	0.22308	0.00754
(2500, 10000]	0.01786	0.01526	0.05376
(10000, 20000]	-0.01860	-0.03942	-0.00924

Table 8: Average Quetelet index table for price category and depth feature

cut price category	Fair	Good	Ideal	Premium	Very Good
(0, 2500]	0.000970	0.001882	0.087047	0.007405	0.015102
(2500, 10000]	0.002970	0.009529	0.003482	0.011328	0.010409
(10000, 20000]	0.007333	0.019882	0.055710	0.037051	0.050586

Table 9: χ^2 table for price category and cut feature

depth category price category	(0.0, 61.0]	(61.0, 63.0]	(63.0, 66.5]
(0, 2500]	0.014124	0.089189	0.000866
(2500, 10000]	0.002989	0.001058	0.020788
(10000, 20000]	0.030107	0.109674	0.030139

Table 10: χ^2 table for price category and depth feature

Sum of values of Table 9 equals to 0.3207 0.2989 and sum of values of Table 10 equals to 0.2989. We will use this information further. Worth to mention, that over these two tables depth range from 61 to 63 differs from expected value larger than other events. But, one should test statistical significance of it before making any statements.

Let us find a number of observations would suffice to see the features as associated at 95% confidence level and 99% confidence level.

```
 * degrees of freedom of depth -> price table = 4*2 = 8
* degrees of freedom of cut -> price table = 2*2 = 4
t value depth -> price for 95% probability :  15.51
t value depth -> price for 99% probability :  20.09
t value cut -> price for 95% probability :  9.49
```

`t value cut -> price for 99% probability :  13.28`

We provide with steps are made to make a conclusion.

1. $n \times t > \chi^2$
2. $n > \frac{\chi^2}{t}$
3. $n_{depth->price,0.95} > \chi^2_{0.95}/t_{0.95} = 15.51/0.2989 > 52$ That means that at any $N > 52$ the hypothesis of statistical independence should be rejected at 95% confidence level.
4. $n_{depth->price,0.99} > \chi^2_{0.99}/t_{0.99} = 15.51/0.2989 > 68$ That means that at any $N > 68$ the hypothesis of statistical independence should be rejected at 99% confidence level.
5. $n_{cut->price,0.95} > \chi^2_{0.95}/t_{0.95} = 9.49/0.3207 > 30$ That means that at any $N > 30$ the hypothesis of statistical independence should be rejected at 95% confidence level.
6. $n_{cut->price,0.95} > \chi^2_{0.95}/t_{0.95} = 13.28/0.3207 > 42$ That means that at any $N > 42$ the hypothesis of statistical independence should be rejected at 99% confidence level.

Comment for each code part can be found in github repository indicated earlier.

5 PCA/SVD

For this task we selected 4 features: 'carat', 'x', 'y', and 'z' because this features has high correlations with each other (see further).

In order to use PCA visualization we need to standardise our data. **Numpy** package as used in seminars allow us to perform required standartizations.

	carat	x	y	z
Original data	41113.66	47.59	2.66	0.57
Z-normalization	1975.13	15.5909	4.77	0.49
Rank-normalization	268.13	1.01	0.2	0.02
Range-normalization	74.72	0.46	0.20	0.02

Table 11: Natural contributions of features

	carat	x	y	z
Original data	99.8765	0.11	0.006	0.001
Z-normalization	98.9546	0.7811	0.2394	0.0249
Rank-normalization	99.54	0.37	0.07	0.00
Range-normalization	99.08	0.61	0.26	0.02

Table 12: Percent conrtibutions of features

Main points of principal components' analysis one can see on Tables 11, 12. Quite interesting situation, it is emerged that carat feature solely can explain more than 99 % of variance of these features. Under careful exploration, we found that in fact carat is a diamonds' weight unit and of course it would depend on its dimensions. Good for us, PCA such a brilliant approach that analyzing the data does not require domain knowledge to produce intuitive results. In this task, SVD helped us to understand true meaning of our feature, as we are know diamonds experts. Let us see obtained conventional PCA visuaizations for different versions of normalization[2]:

[2]We used sklearn.cluster.PCA(n components=2) that produces two principal components we need

Figure 5: Scatter plot of PC for original data

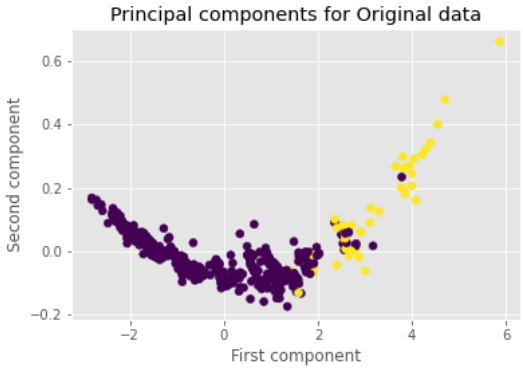


Figure 6: Scatter plot of PC for z-normalized data

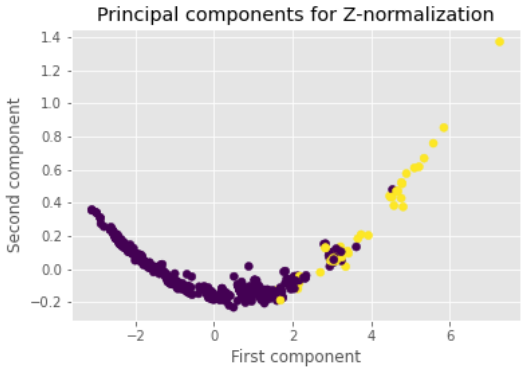


Figure 7: Scatter plot of PC for rank-normalized data

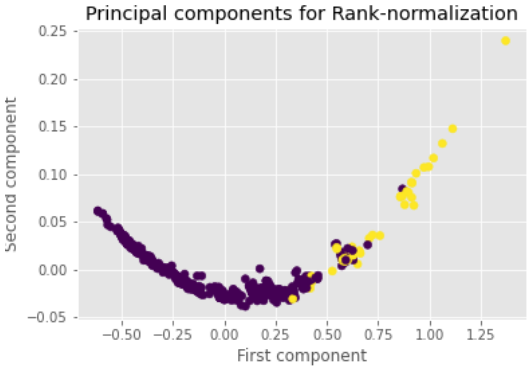
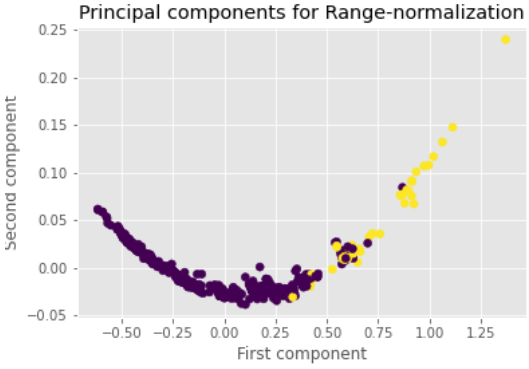


Figure 8: Scatter plot of PC for range-normalized data



We colored yellow those diamonds which price is higher than 10 000 $ and violet the others . As wee see, first component almost perfectly devide out data according to its price properties, therefore we can interpret the first principal component as quality of a diamond hidden feature. There is not significant difference between any normalization procedure, as scatter plots look quite similar, however we prever z-scoring procedure because that normalization is the most popular according to our little experience. With high probability we can say that second component also carries some hidden interpretable information, but unfortunately we failed to uncover it using our small dataset.

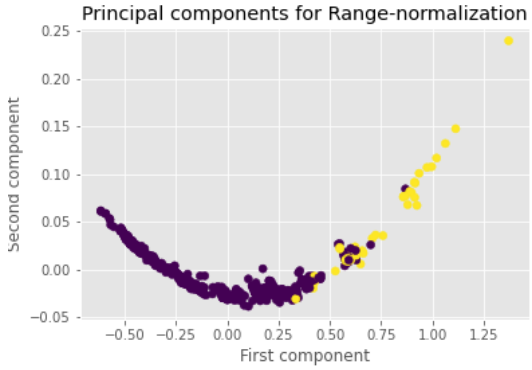
Figure 9: Ranking analysis results

```
c :  0.3549 0.5287 0.5303 0.5596
loading :  0.1798 0.2679 0.2687 0.2835
top_5 :  100.0 87.2928 84.8058 82.678 81.5255
arg_top_5 :  127 326 453 156 246
```

- there is the perfect diamond (value of ranking factor is 100). It is 127 diamond. This diamond is the best.

On Figure 9 one can see indices for top-5 diamonds accoring to their rank and also some auxiliary information for rank estimation.

Figure 10: Scatter plot of PC for range-normalized data



6 Correlation analysis

Compute and show correlation matrix.

	carat	depth	price	x	y	z
carat	1.000000	0.000490	0.913612	0.978323	0.978141	0.975653
depth	0.000490	1.000000	-0.019945	-0.034895	-0.036634	0.082426
price	0.913612	-0.019945	1.000000	0.877762	0.880222	0.874172
x	0.978323	-0.034895	0.877762	1.000000	0.999005	0.992642
y	0.978141	-0.036634	0.880222	0.999005	1.000000	0.992485
z	0.975653	0.082426	0.874172	0.992642	0.992485	1.000000

Table 13: Correlation Matrix

Visualise pairwise scatterplots (with regression line) in figure 11.

Let's choose carat - price pair of features with correlation = 0.913612. Let's visualize scatterplot for carat - price one more time (figure 12).

Now we build linear regression model on feature 'carat' to predict target variable 'price' (figure 13).

Slope of the model is **8052.28**. First of all, slope has positive value, so correlation between carat and price has positive direction. The value of the slope indicates that by increasing value of carat by 1 unit, price of the diamond increases by **8052.28** on average.

```
Correlation:  0.9136116464594373
Determinacy coefficient:  0.8346862405463232
```

Figure 11: Pairwise scatterplots with regression line

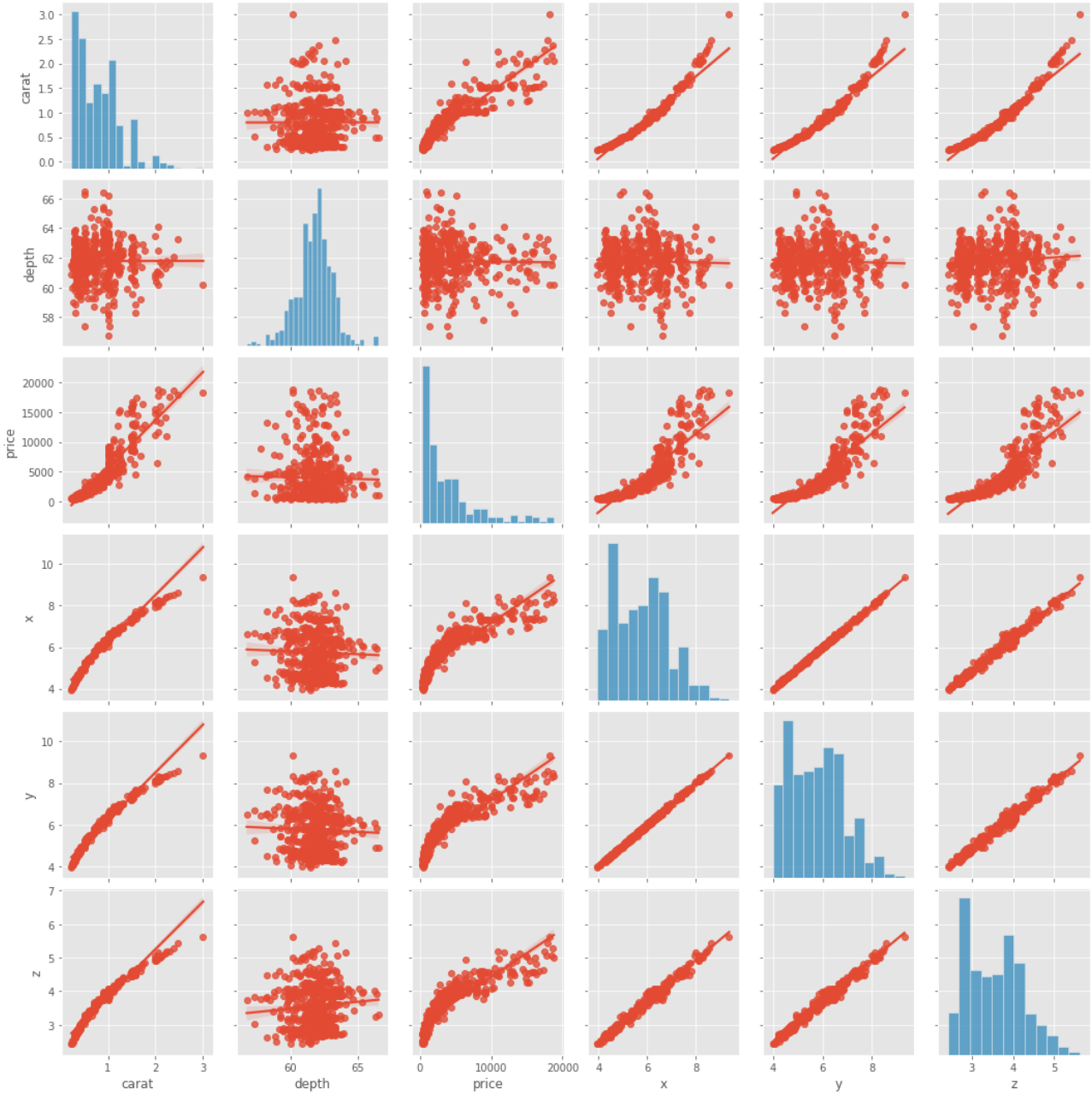


Figure 12: Scatterplot for carat - price

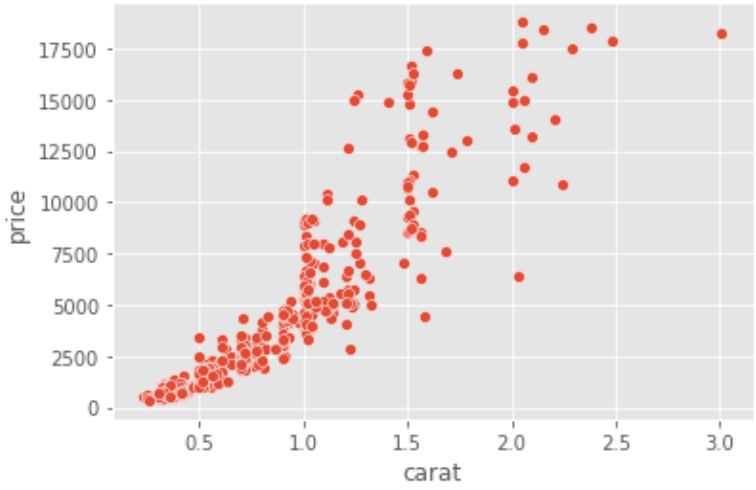
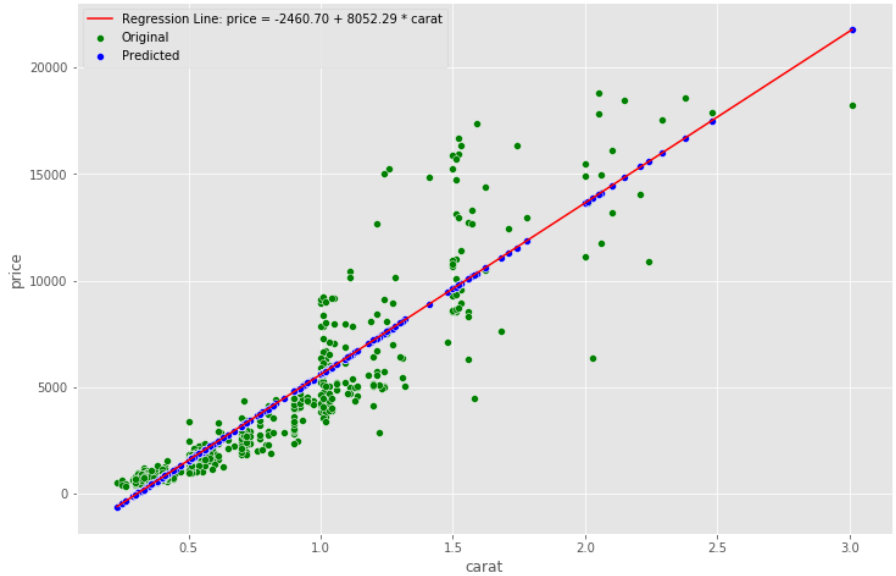


Figure 13: Linear regression carat on price



Determinacy coefficient is **0.834**. It means that built linear regression model explains about **83%** of target feature variance. This score is quite high for our dataset, actually one can build forecasting model for price solely on carat feature. This corresponds to common knowledge about pricing of diamonds.

Now let's build linear regression model on feature set {"carat", "x", "depth"} for target "price".

Determinacy coefficient of extended model on carat, x, depth features is **0.842**. Not a great increase from previous try. Earlier, scatterplots showed that depth has almost zero correlation with target variable, now we see that this feature didn't contribute that much. Despite feature x is highly correlated with target variable, it also has high correlation value with almost existing feature carat, so it also didn't increase model quality much.

Let's compare mean relative absolute errors of two models.

```
Mean Relative Absolute Error
{carat} -> {price}:  0.40102417980974486
{carat, x, depth} -> {price}:  0.2648060500901686
```

Interesting result. Extended model shows more or less significant improvement compared to single-feature model. In general both models (one-feature and multiple-feature) have adequate forecasting performance according to mean relative absolute error critetia. It is important to mention that models did not train to minimize this criteria. Extended model has 14% improvement despite the fact that it didn't show significant increase in R^2 criteria.

This section used simple sklearn.linear models.LinearRegression class with default parametrs that returns slope and intercept of an estimated model.