Modern Methods of Data Analysis Home Work

Pugachev Alexander, Solovyov Alexey

02.12.2019

Contents

1 Dataset information

For the project on Modern Methods of Data Analysis we decided to choose Mobile Price Classification dataset[1]. This dataset represents different characteristics of various mobile phones and their price category (4 categories). It is used for classification task: to predict price category of mobile phone based on its characteristics. The initial dataset consists of 2000 entities with 21 features. We chose 400 entities (100 for each category) and the most important features in our opinion, these features are:

- battery_power (quantitative): Total energy a battery can store in one time measured in mAh
- clock_speed (quantitative): Speed at which microprocessor executes instructions
- four_g (binary): Has 4G or not
- int_memory (quantitative): Internal memory in gigabytes
- n_cores (quantitative): Number of cores of processor
- pc (quantitative): Primary camera mega pixels
- ram (quantitative): Random access memory in Megabytes
- talk_time (quantitative): Longest time that a single battery charge will last when you are
- touch_screen (binary): Has touch screen or not
- wifi (binary): Has Wi-Fi or not
- price_range (categorical): Price categories with values of 0 (low cost), 1 (medium cost), 2 (high cost), 3 (very high cost)

So, the dataset we will work with consists of 400 entities with 11 features. We present first five elements of the dataset in Table 1.

battery_power	clock_speed	four_g	int_memory	n_cores	pc	ram	talk_time	touch_screen	wifi	price_range
1142	1.4	0	6	8	8	663	5	1	1	0
728	1.7	1	5	2	20	834	5	1	0	0
1868	0.5	1	40	8	17	298	17	1	0	0
890	2.2	0	44	8	13	751	3	0	0	0
1433	1.6	0	4	8	11	258	4	1	1	0

Table 1: First five rows of the dataset.

2 K–Means

2.1 Feature selection and applying K–Means

For this task we took 4 quantitative features from our dataset:

battery_power, pc, int_memory and ram.

We did not choose binary or categorical features because the Euclidean distance which is used in K–Means is not really meaningful on the space of categorical features. Also, in our opinion among numerical features these features are the most important for predicting mobile phone price. We standardized each feature by using range normalization and applied K–Means with $K = 5$ and $K = 9$. In each case we made 20 random initializations and

[1]Mobile Price Classification https://www.kaggle.com/iabhishekofficial/mobile-price-classification

took best partition based on inertia value — sum of squared distances of samples to their closest cluster center. After getting best partition we computed relative differences for each cluster and feature, the results are presented in Table 2 and Table 3, values whose absolute value is greater than 30 are highlighted.

Cluster	battery_power	pc	int_memory	ram	Number of elements in cluster
1	19.201	-4.226	**52.238**	**48.499**	82
2	-25.820	**-41.180**	**-46.782**	8.655	92
3	8.399	**-47.176**	**50.558**	**-45.033**	76
4	**37.271**	**43.081**	**-50.063**	-14.601	71
5	**-31.438**	**59.008**	-3.385	-3.975	79

Table 2: Relative differences with $K = 5$. Minimum inertia equals to 69.5278.

Cluster	battery_power	pc	int_memory	ram	Number of elements in cluster
1	**36.966**	**56.288**	**-50.804**	**-48.224**	38
2	28.871	**44.726**	**55.438**	**40.896**	49
3	**-36.113**	-6.477	**-48.772**	**41.298**	47
4	**32.491**	4.518	**-51.249**	**44.018**	40
5	-27.473	14.828	**62.378**	**-37.392**	45
6	-28.790	**67.809**	**-45.040**	-14.597	39
7	-12.416	**-53.841**	**47.193**	**47.986**	51
8	-27.769	**-51.398**	**-43.397**	**-47.598**	41
9	**34.181**	**-53.309**	**37.566**	**-42.340**	50

Table 3: Relative differences with $K = 9$. Minimum inertia equals to 46.5020.

We also performed the same experiment with number of clusters equal to number of categories ($K = 4$). In this case we want K–Means to cluster items according to their categories. The results are presented in Table 4, values whose absolute value is greater than 30 are highlighted.

Cluster	battery_power	pc	int_memory	ram	Number of elements in cluster
1	**-33.652**	15.016	-28.915	-25.209	111
2	18.999	**-48.584**	**42.282**	**-33.982**	91
3	**34.432**	**52.033**	-22.739	5.996	101
4	-15.166	-25.783	17.099	**54.485**	97

Table 4: Relative differences with $K = 4$. Minimum inertia equals to 80.2316.

2.2 Interpretation of partitions

Let's analyze the constructed clusters and partitions and start with the case when $K = 5$. According to Table 2 the first cluster consists of the elements which have big value of internal memory and RAM. The second cluster contains mobile phones with weak primary camera and small amount of internal memory. Mobile phones which have weak camera, small amount of random access memory and big amount of internal memory belong to the third cluster. Cluster №4 contains phones with battery power better than the average and better primary camera. The fifth cluster consists of smartphones with slightly worse battery and significantly better camera than the average.

Now let's interpret partition in case $K = 9$. The first cluster consists of elements with good battery power and primary camera and both small internal and random access memories. The second cluster contains smartphones with good camera, internal and random access memories. The third cluster has phones with low batteries and small amount of internal storage. However the elements from this cluster have high value of RAM. Smartphones with battery power slightly better than the average, small amount of storage and large amount of RAM belong to the fourth cluster. Cluster №5 consists of the phones which have the highest value of internal memory. However their RAM value is less than the average. Comparing to the other clusters, cluster №6 consists of the phones with best camera. In spite of this the amount of internal memory of these phones is much less than the average. The seventh cluster contains smartphones with bad cameras but good values of internal memory and RAM. The eighth cluster is formed by phones with bad cameras, low amount of both internal and random access memories. Finally, the ninth cluster consists of phones which have battery and internal memory slightly better than the average but with bad cameras and RAM values.

Let's describe the clusters in case $K = 4$. The first cluster consists of phones with small battery power. The second one contains phones with bad camera, high value of internal memory and low RAM value. The third cluster consists of the phones with both good battery power and cameras. The cluster №4 contains smartphones with high amount of random access memory.

2.3 Conclusion

After getting different partitions we can provide the following conclusion. According to the results we can see that the battery_power feature has practically the same value in each cluster we get. The maximum difference between the mean value of this feature in a cluster and the grand mean is less than 38%. Also, this feature in most cases differs from the grand mean by less than 30%. We consider that this feature can be excluded from the clustering process.

Talking about the best partition, we consider that we obtain the best one when the number of clusters is equal to 4. In this case for each feature (except int_memory) we have two different clusters that respectively contain elements with low and high values of this feature. The difference between elements from different clusters is the most noticeable in case when $K = 4$ because there are no two different clusters which both have high (or low) value of a certain feature.

3 Bootstrap

In this homework, we chose the partition that consists of five clusters. We got this data in the previous task. We made random multiple entity samples of same size 5000 (with replacement). The number of entities is 150. In this task we used random sampling method (bootstrap) that why we fixed random seed that equal 10. It was done to make the results repeatable.

There are different methods for calculating confidence intervals and comparing statistics of features. In our case, we used bootstrapping. Each application of bootstrap was done in both, pivotal and non-pivotal, versions. In the first instance, you simply takes the empirical quantiles from the bootstrap distribution of the parameter. On another version, you need to use percentiles of the bootstrap distribution.

3.1 Confidence interval for feature grand mean

First of all, we took the feature from our dataset:

battery_power

Further, we built the histogram of battery_power variable. This histogram is depicted on Figure 1.

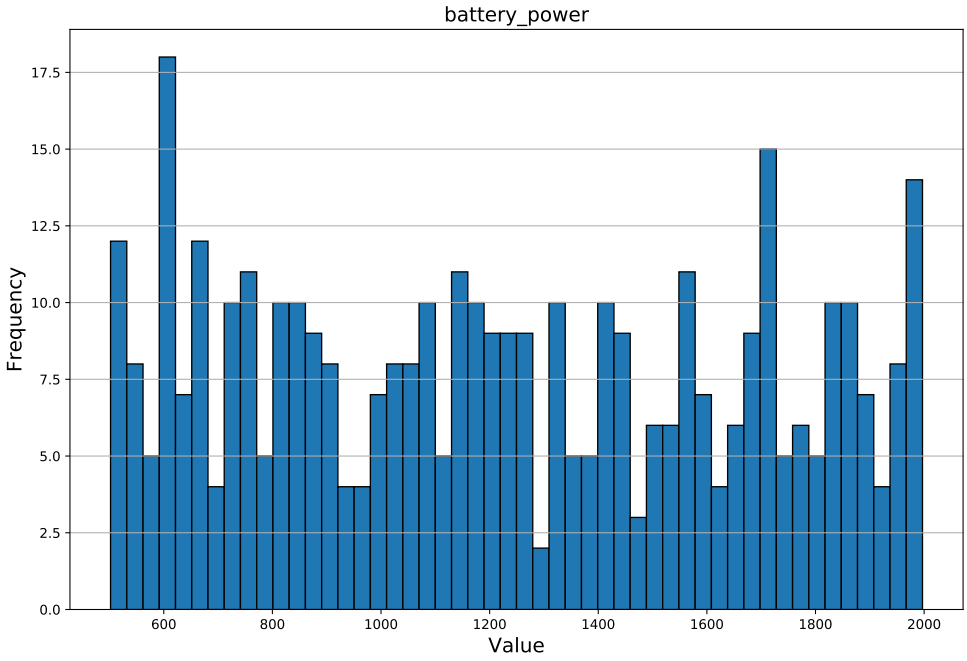

Figure 1: Histogram of bootstrap feature.

- bins : 100
- expected value of feature : 1226.425
- standard deviation of feature : 448.959

The distribution is rather far from Gaussian. In order to calculate the 95% confidence interval for feature grand mean we created bootstrap sample of means. Further, we built histogram of this variable. The Figure 2 shows its histogram, for each of the means we took 100 bins.

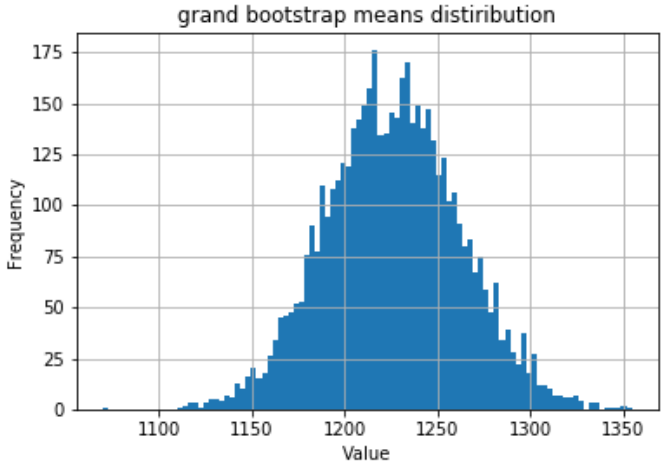

Figure 2: Histogram of bootstrap means.

As you can see, the distribution is quite close to the Gaussian. Its expected value equals 1226.427 and standard deviation is 36.261.

We found 95% confidence interval for the grand mean:

- Pivotal method : $[1155.356, 1297.498]$
- Non-pivotal method :$[1157.418, 1298.274]$

These intervals are very similar. In non-pivotal version we took 2.5% and 97.5% percentiles as the boundaries. The actual value of the mathematical expectation of the battery power lies within the intervals obtained with the bootstrap. Therefore, the confidence intervals are correct.

3.2 Comparing the within–cluster means

We compared second and fourth clusters in our dataset. We built the histogram of means bootstrap distribution for each cluster. The Figure 3 represents its histogram, for each of the features we took 100 bins.

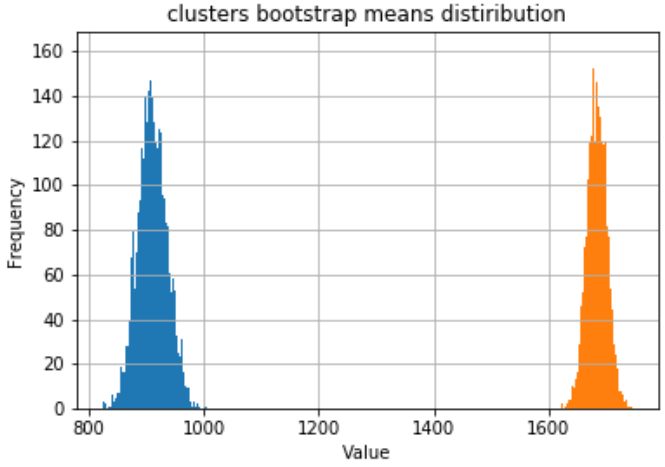

Figure 3: Histogram of clusters bootstrap means.

Both distributions are quite close to the Gaussian. Their expected values equal 910.482; 1683.529 and standard deviations are 25.818; 17.651 respectively. The histogram shows that means in clusters are very different. In general, means of first cluster are much smaller than means the another cluster. It means that the clustering algorithm separated well the objects in the dataset (in previous step homework). Further, we calculated differences between distributions and drew its histogram. The Figure 4 represents its histogram.

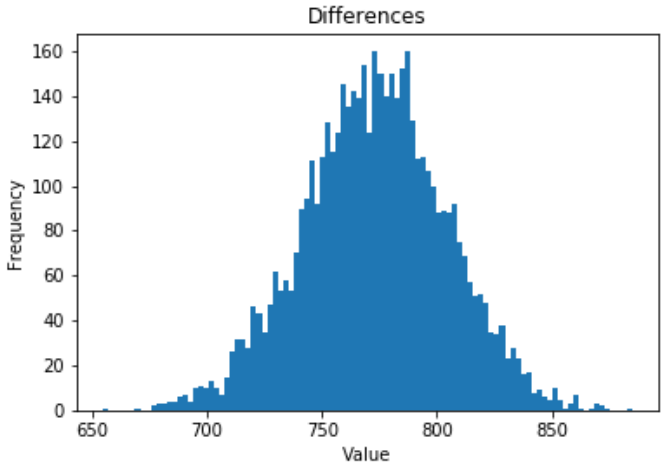

Figure 4: Histogram of clusters differences.

We wanted to compare mean in different taxes that why we found 95% confidence interval A for differences D. 95% confidence interval A for D :

- Pivotal method : $[711.592, 834.504]$
- Non-pivotal method :$[711.317, 833.949]$

In both cases, confidence intervals do not contain zero. It means that expectations in clusters are different. In other words, one mean is greater than the other.

3.3 Comparing the grand mean with the within–cluster mean

We compared the grand mean with the within mean of second cluster by using bootstrap. We built the histogram of means bootstrap distribution. The Figure 5 represents its histogram, for each of the features we took 100 bins.

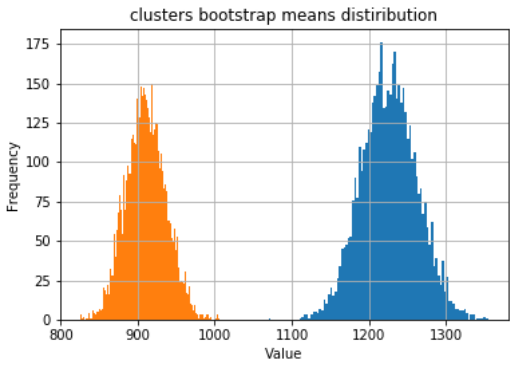

Figure 5: Histogram of differences.

Both distributions are quite close to the Gaussian. Their expected values and standard deviation were calculated in previous sub tasks. They are different. The histogram shows that means distribution of cluster is not like grand

means distribution. We calculated differences between distributions and drew its histogram 6. Also, we found 95% confidence interval for differences in both version for comparing the grand mean with the within cluster mean.

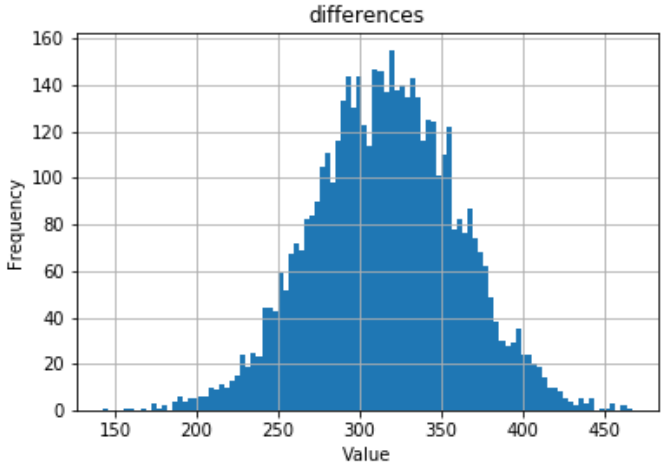

Figure 6: Histogram of differences.

95% confidence interval for D :

- Pivotal method : $[228.388, 403.502]$
- Non-pivotal method :$[228.465, 403.222]$

Confidence intervals are very similar. In both variants, they do not contain zero. Hence, means differ.

4 Contingency Table

For the fourth homework we took the next three features from our dataset:

battery_power, ram, price_range.

The first two features are quantitative and we need to categorize them. The last one feature, price_range, is categorical and it is target feature of our dataset.

4.1 Contingency tables

Let's consider battery_power feature. Based on this feature we developed a nominal one in the following way. First of all we built the histogram of this feature. The Figure 7 represents its histogram, for each of the features we took 50 bins.

Then we found minimas of histogram which define the boundaries of categories. Based on the histogram we defined the following boundaries for the battery_power feature (the first and the last values are minimum and maximum values of the feature:

$$battery\_bounds = [502; 950; 1300; 1600; 1997]$$

Similarly we built histogram and found boundaries for ram feature. The histogram is shown on Figure 8. The boundaries for this feature are the following:

$$ram\_bounds = [258, 1000, 2400, 3200, 3978]$$

Based on these boundaries we can build contingency tables for each feature. We define as *categories* the groups of elements that are divided by the counted boundaries and as *classes* the groups of elements with a specific value of price_range feature from the initial dataset. The elements from first class are elements which have value 0

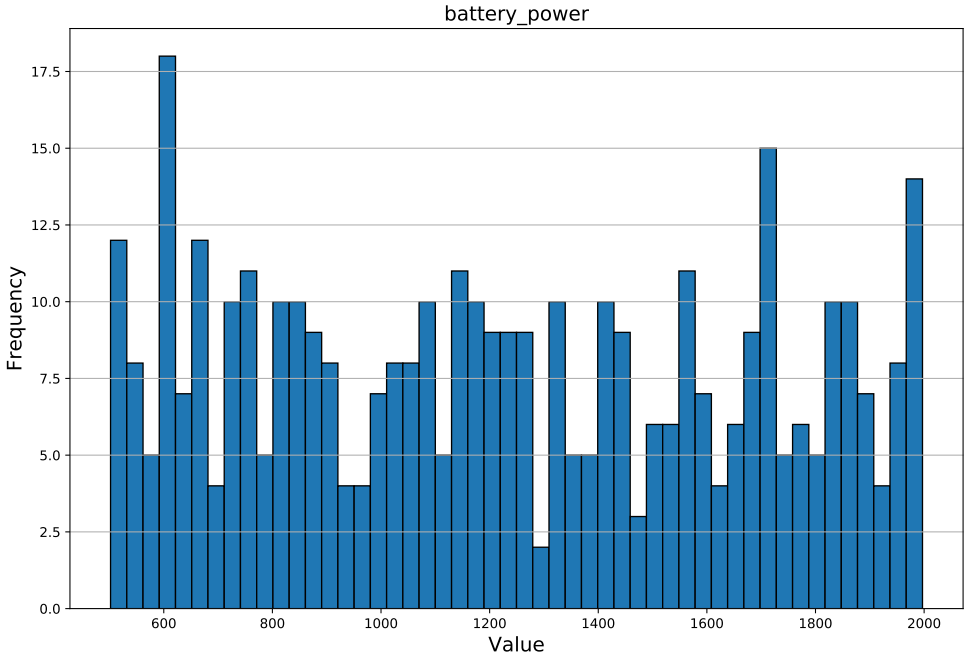

Figure 7: Histogram of battery_power feature.

of price_range feature, these are low cost mobile phones. Thereby the elements from the fourth class are very expensive mobile phones. So, we are dealing with four categories and four classes. The contingency tables of features battery_power and ram are presented in Tables 5 and 6 respectively.

Class	Category 1	Category 2	Category 3	Category 4	Total
Class 1	44	25	10	21	100
Class 2	34	25	21	20	100
Class 3	37	19	16	28	100
Class 4	18	21	24	37	100
Total	133	90	71	106	400

Table 5: Contingency table over battery_power feature.

Class	Category 1	Category 2	Category 3	Category 4	Total
Class 1	69	31	0	0	100
Class 2	8	87	5	0	100
Class 3	0	32	58	10	100
Class 4	0	1	31	68	100
Total	77	151	94	78	400

Table 6: Contingency table over ram feature.

The values in contingency tables mean how many elements that belong to a certain category fall into a particular class. For example, looking at Table 6 there are 69 objects which have ram feature value that belongs to the first

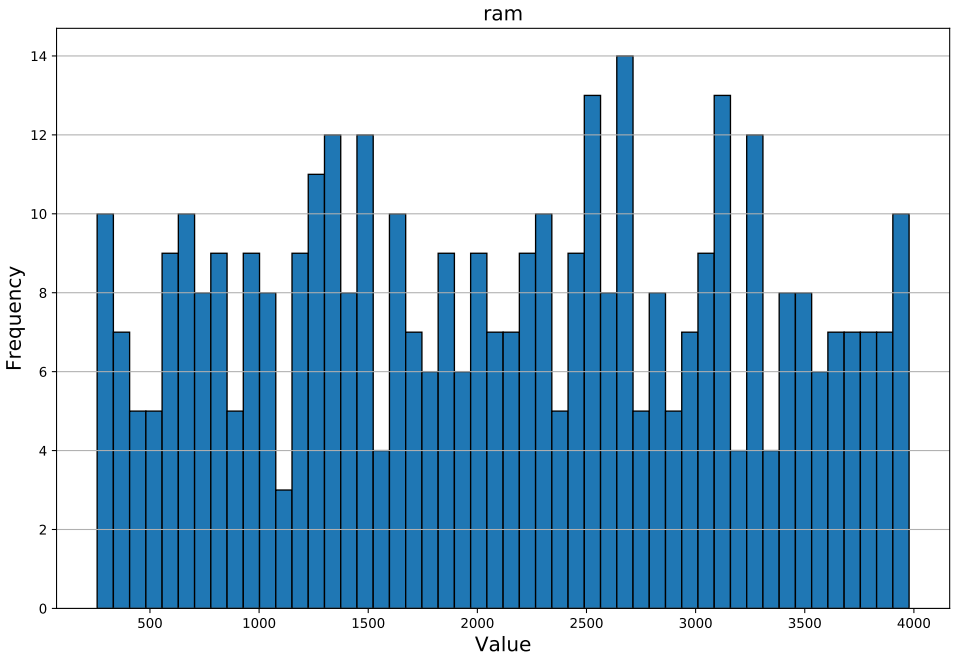

Figure 8: Histogram of ram feature.

category and are in the first class. Based on the contingency table we can build relative frequency table by dividing each element of contingency table by the number of elements in the initial dataset. The relative frequency tables are presented in Tables 7 and 8. They would be useful for us in the next tasks.

Class	Category 1	Category 2	Category 3	Category 4	Total
Class 1	0.110	0.062	0.025	0.052	0.250
Class 2	0.085	0.062	0.052	0.050	0.250
Class 3	0.092	0.048	0.040	0.070	0.250
Class 4	0.045	0.052	0.060	0.092	0.250
Total	0.333	0.225	0.177	0.265	1.000

Table 7: Relative frequency table over battery_power feature.

Class	Category 1	Category 2	Category 3	Category 4	Total
Class 1	0.172	0.077	0.000	0.000	0.250
Class 2	0.020	0.217	0.013	0.000	0.250
Class 3	0.000	0.080	0.145	0.025	0.250
Class 4	0.000	0.003	0.077	0.170	0.250
Total	0.193	0.378	0.235	0.195	1.000

Table 8: Relative frequency table over ram feature.

4.2 Conditional frequency tables

Having several counted contingency tables we can build conditional frequency tables for each feature. In order to build conditional frequency table for each element in contingency table we have to divide it by the total number of elements in corresponding column. Thus, for example, for element 69 in the contingency Table 6, the corresponding one in the conditional frequency table will be $69/77 = 0.896$. The calculated conditional frequency tables for considered features are presented in Tables 9 and 10. The significant values are highlighted in the tables.

Class	Category 1	Category 2	Category 3	Category 4
Class 1	0.331	0.278	0.141	0.198
Class 2	0.256	0.278	0.296	0.189
Class 3	0.278	0.211	0.225	0.264
Class 4	0.135	0.233	0.338	0.349

Table 9: Conditional frequency table over battery_power feature.

Class	Category 1	Category 2	Category 3	Category 4
Class 1	**0.896**	0.205	**0.000**	**0.000**
Class 2	0.104	**0.576**	**0.053**	**0.000**
Class 3	**0.000**	0.212	**0.617**	0.128
Class 4	**0.000**	**0.007**	0.330	**0.872**

Table 10: Conditional frequency table over ram feature.

In table cell there is written probability of an element to be in the i-th class provided that it belongs to the j-th category. For example, if an elements is in the first category of battery_power feature, there probability of this element to be in the first class is 33.1% (according to Table 9).

4.2.1 Interpretation

The conditional frequency tables of feature battery_power can be considered a "dull" one because it do not contain any high or low values. So, in this case the conditional probability can not help us to properly identify to which class does the item belong according to its category. On the other hand, according to the Table 10 we could provide some consequences based on conditional probability. For example, we can see that if the item's ram feature value got into the first category, there exists 89.6% probability that this item belongs to the first class. In other words, if a mobile phone has few random access memory, its price in most cases will be low. On the other hand, if a smartphone has big amount of RAM, we can say that with the probability of 87.2% it will cost much. Moreover, if a mobile phone has small amount of RAM, it can not cost much, on the other hand, if it has big amount of random access memory it can not be cheap.

4.3 Quetelet relative index tables

Using the computed conditional frequency tables we can build Quetelet relative index tables. For a certain feature this matrix is built by element-wise dividing the relative contingency table by the relative frequency table under independence and subtracting 1 from each element of matrix. We computed the Quetelet relative index tables for each of the considered features and the results are presented in Tables 11 and 12. The significant values are highlighted in the tables.

Class	Category 1	Category 2	Category 3	Category 4
Class 1	0.323	0.111	**-0.437**	-0.208
Class 2	0.023	0.111	0.183	-0.245
Class 3	0.113	-0.156	-0.099	0.057
Class 4	**-0.459**	-0.067	0.352	0.396

Table 11: Quetelet relative index table over battery_power feature.

Class	Category 1	Category 2	Category 3	Category 4
Class 1	**2.584**	-0.179	**-1.000**	**-1.000**
Class 2	-0.584	**1.305**	-0.787	**-1.000**
Class 3	**-1.000**	-0.152	**1.468**	-0.487
Class 4	**-1.000**	**-0.974**	0.319	**2.487**

Table 12: Quetelet relative index table over ram feature.

4.3.1 Interpretation

The Quetelet tables allowed us to find new dependencies in data which we could not detect based on conditional frequency tables. Namely, according to Table 11 if we find out that our element belongs to the first category of battery_power feature, the probability of this element to be in the fourth class decreases by approximately 46%. Also, if we take into consideration that our element belongs to the third category of the same feature, the probability of this element to belong to the first class decreases by 43.7%.

Talking about the Quetelet table over ram feature, it fully corresponds to the conditional frequency table over ram feature. For example, if we take into consideration that our element belongs to the first category, the probability of this element to belong to the first class increases by 258.4%. Similarly, the items from category №4 are 248.7% more likely to belong to the class №4. This corresponds with the statement that the more ram mobile phone has, the more expensive it is and vice versa.

4.4 Chi–square index

The chi–square index for a certain feature we computed using the following formula:

$$\chi^2 = \sum_{i,j}((ngr - ngn).*(ngr - ngn))./ngn$$

where:

ngr — relative frequency table;

ngn — relative frequency under independence table;

$.*$ — element–wise multiplication;

$./$ — element–wise division;

$\sum_{i,j}$ — sum of all elements of the matrix.

Thus we obtained the following χ^2 values for each of the features:

$$\chi^2_{battery} = 6.36\%$$
$$\chi^2_{ram} = 132.8\%$$

The χ^2 value tells us what could be the average relative increase in the occurrence of a certain class when the category becomes known. We can notice that for the battery_power feature we have very small χ^2 value. That means that on the average the knowledge of which category does the item belong to does not help us to determine in what class does this object fall into. Analyzing the χ^2 value for ram feature, we can conclude that if we know what category does the element belong to, we can qualitatively identify what is the class of this element.

Let's find out what numbers of observations would be sufficient to see the features as associated at 95% and 99% confidence levels. Firstly, we found out the number of degrees of freedom. In our case there are 4 categories and 4 classes so the number of degrees of freedom is equal to $f = (4-1)*(4-1) = 9$. According to the χ^2 table for the 95% probability, t is equal to 16.92, for the 99% probability, t is equal to 21.67. Now we find smallest natural N such that $N \cdot \chi^2 > t$.

For battery_power feature:

$$N_{95} > \frac{t_{battery}}{\chi^2} = \frac{16.92}{0.0636} = 266.037 \Rightarrow N_{95} = 267$$
$$N_{99} > \frac{t_{battery}}{\chi^2} = \frac{21.67}{0.0636} = 340.723 \Rightarrow N_{99} = 341$$

For ram fearute:

$$N_{95} > \frac{t_{ram}}{\chi^2} = \frac{16.92}{1.327} = 12.75 \Rightarrow N_{95} = 13$$
$$N_{99} > \frac{t_{ram}}{\chi^2} = \frac{21.67}{1.327} = 16.33 \Rightarrow N_{99} = 17$$

According to the results that we obtained, for battery_power feature, if N is more or equal than 267 the hypothesis of statistical independence of elements should be rejected at 95% confidence level. If N is more or equal than 341, then the hypothesis should be rejected at 99% confidence level. If N is less than 267 then the hypothesis should be accepted. Similarly for the ram feature, if N is more or equal than 13, the hypothesis should be rejected with 95% confidence. If it is even more or equal than 17, hypothesis should be rejected with 99% confidence. In cases when N is less than 13, hypothesis should be accepted.

5 PCA/SVD

5.1 Feature selection

For this task we chose 5 quantitative features:

battery_power, clock_speed, talk_time, pc, ram.

For the PCA/SVD task we did not choose any categorical features because mathematical operations which are used in the SVD algorithm do not make any sense when applying them on categorical features. Also in our opinion all of these features describe mobile phones in sufficient details and can be used for predicting the price category of mobile phones.

5.2 Contribution of principal components

After selecting features we performed ranking standardization on our dataset. There are presented first five rows of the Table 13 after ranking normalization.

Then we computed data scatter (ds) using formula:

$$ds = \sum_{i}^{400} \sum_{j}^{5} Y_{ij}$$

where:

	battery_power	clock_speed	talk_time	pc	ram
1	0.428	0.360	0.166	0.400	0.109
2	0.151	0.480	0.166	1.000	0.155
3	0.914	0.000	0.833	0.850	0.011
4	0.260	0.680	0.056	0.650	0.133
5	0.623	0.440	0.111	0.550	0.000

Table 13: First five rows of the dataset after performing ranking standardization.

$Y = X * X$;

X — standardized matrix;

$*$ — element–wise multiplication;

We got that the data scatter of our standardized matrix is equal to 649.702. Then we applied Singular Value Decomposition to this matrix and obtained singular values and principal components (squared singular values). Besides the natural contribution of principal components to the data scatter we also computed their percentage contribution. We sorted values in descending order and the results are presented in the Table 14.

Singular value	Natural contribution	Percentage contribution, %
22.351	499.569	76.892
6.727	45.250	6.965
6.229	38.799	5.971
5.849	34.206	5.265
5.646	31.878	4.907

Table 14: Singular values and the contribution of principal components to the data scatter.

According to the principal components which we obtained, we can say that the first component makes the biggest contribution to the data scatter (over 76%). The other components make significantly less contribution (approximately 5–7 %).

5.3 Hidden ranking factor

Now we need to find hidden factor. Let's take a look at the loadings:

$$c_1 = [-0.453, -0.376, -0.464, -0.477, -0.460]$$

All of the elements are negative. We make them positive because the weights of features must be positive:

$$c_1 = [0.453, 0.376, 0.464, 0.477, 0.460]$$

Next we apply equation:

$$Z = (0.453 * \text{battery\_power} + 0.376 * \text{clock\_speed} + 0.464 * \text{talk\_time} + 0.477 * \text{pc} + 0.460 * \text{ram}) * \alpha$$

We find α considering that Z is equal to 1 when each subject mark is equal to 1:

$$1 = (0.453 * 1 + 0.376 * 1 + 0.464 * 1 + 0.477 * 1 + 0.460 * 1) * \alpha$$

$$\Downarrow$$

$$\alpha = 0.448$$

The final equation is:

$$Z = 0.203 * \text{battery\_power} + 0.168 * \text{clock\_speed} + 0.209 * \text{talk\_time} + 0.214 * \text{pc} + 0.206 * \text{ram}$$

Comparing to the mean case when each of the weights is equal to 0.2 we can notice that the weight of clock_speed has significantly lower value (0.168). Other weights have slightly higher values, especially the weight of pc feature, which is equal to 0.214. From the obtained results we can conclude that pc feature is 27% more important than clock_speed (0.214/0.168 = 1.273). Its importance comparing to the other feature is much less:

$$0.214/0.203 = 1.05 \text{ (pc is 5\% more important than battery\_power)}$$
$$0.214/0.209 = 1.02 \text{ (pc is 2\% more important than talk\_time)}$$
$$0.214/0.206 = 1.04 \text{ (pc is 4\% more important than ram)}$$

We can consider that the importance of features battery_power, talk_time and ram is approximately the same.

5.4 Visualization

Next we need to visualize our data using two first principal components. For this we applied two types of standardization to our data: range normalization and z–scoring. There are presented first five rows of standardized versions of data in Tables 15 and 16.

	battery_power	clock_speed	talk_time	pc	ram
1	-0.057	-0.045	-0.334	-0.112	-0.390
2	-0.333	0.075	-0.334	0.488	-0.344
3	0.429	-0.405	0.333	0.338	-0.488
4	-0.225	0.275	-0.445	0.138	-0.367
5	0.138	0.035	-0.389	0.038	-0.499

Table 15: First five rows of the dataset after performing range normalization.

	battery_power	clock_speed	talk_time	pc	ram
1	-0.188	-0.136	-1.071	-0.376	-1.371
2	-1.109	0.229	-1.071	1.638	-1.210
3	1.427	-1.231	1.068	1.135	-1.716
4	-0.748	0.838	-1.427	0.464	-1.288
5	0.460	0.108	-1.249	0.128	-1.754

Table 16: First five rows of the dataset after performing z–scoring.

We applied SVD algorithm to each of the matrices and took two first singular triplets in each case. The visualizations (scatter plots) of normalized data are presented in Figures 9 and 10. The first singular triplet is located on X–axis, the second one is located on Y–axis. We coloured the elements from the dataset according to their category, the legend of scatter plot is presented in Figures 9 and 10.

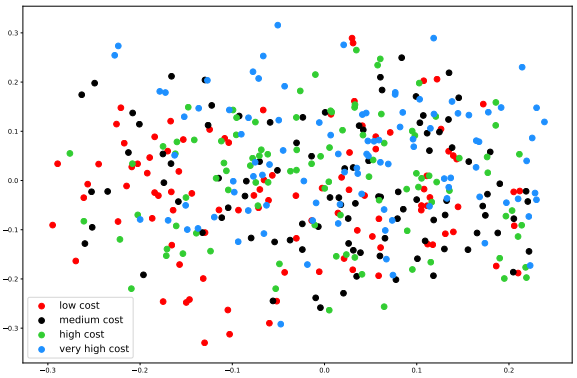

Figure 9: Visualization of the data with range standardization.

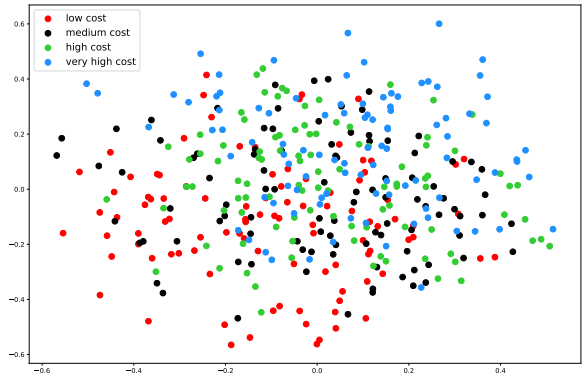

Figure 10: Visualization of the data with z–scoring normalization.

5.5 Conventional PCA

We also found two first principal component vectors and visualized them using conventional PCA approach. For this we centered our initial dataset (by subtracting mean values from each feature), computed matrix $Y = X^T X$ (where X is centered version of initial dataset) and divided each element of Y matrix by the number of elements in initial dataset. After that we found eigenvalues and eigenvectors of matrix Y, sorted them by descending order of eigenvalues and for the first and second eigenvalue and eigenvector we computed principal component scoring vectors using formula:

$$z_i = \frac{Y c_i}{\sqrt{N \lambda_i}}$$

where:

$Y = X^T X$, X — centered matrix;

c_i — i-th eigenvector of Y;

N — number of elements in initial dataset;

λ_i — i-th eigenvalue of Y.

After computing the first two principal component vectors using conventional PCA we made the visualization which is shown of Figure 11. The values of the first vector are set on X-axis, values of the second vector are set on Y-axis. The first two eigenvalues of matrix Y ($Y = X^T X$) are $\lambda_1 = 21150.622$, $\lambda_2 = 8978.545$. These values are equal to the singular values of matrix X, which is centered matrix of the initial dataset.

5.6 Conclusion

According to the visualizations that we obtained we can claim that the best visualization was built after applying conventional PCA approach over the centered dataset. We can see that in this case the elements from different categories are separated from each other much better than in the other cases. Also the elements from the same class are grouped very well, we can not note such good grouping of elements by classes in the other visualizations. In case of conventional PCA there is no such small area of space in which objects of all four categories could be located. For the other two visualizations, such areas are ubiquitous.

The reason for such a big difference can be the division of element values on range or standard deviation. In z–scoring and ranking normalization after subtracting the mean value we must divide our values on standard deviation and difference of maximum and minimum respectively. If we do not apply division but only subtract the means from the data, after computing SVD of the centered matrix we obtain singular values and vectors which are equal to the eigenvalues and eigenvectors of matrix in conventional PCA. That means that the visualization in case of applying SVD to the centered matrix will be the same as in case of conventional PCA.

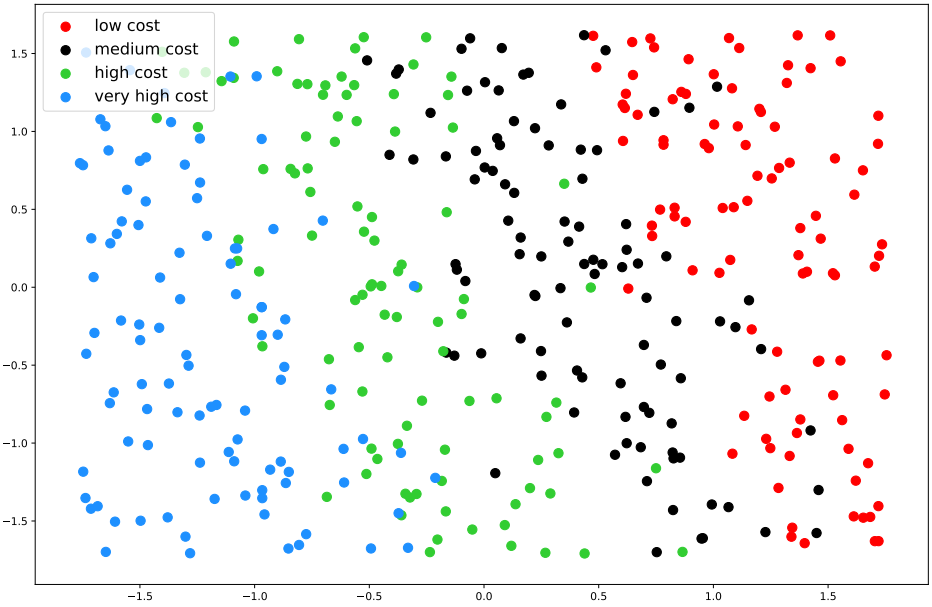

Figure 11: Visualization of the data with principal component vectors found by conventional PCA.

6 Correlation Coefficient

6.1 Feature selection

For this task we should take two features which have "linear-like" scatter plot. Unfortunately, none of the feature pairs do have "linear-like" scatter plot, the screenshot of all scatter plots is available on Github[2]. However, we have chosen ram and battery_power features and the first half of the dataset (200 elements, categories 0 and 1). Based on the ram feature we would like to predict values for battery_power feature. The scatter plot of these features which is shown on Figure 12 can be considered as "linear-like".

This pair of features is suitable for building linear regression because according to the scatter plot we can suggest that there could exists an inverse linear dependency over ram and battery_power feature.

6.2 Linear regression

Next we built linear regression over the ram and battery_power features. A linear regression line has an equation of the form:

$$y = ax + b$$

where:
x — the predictor variable;
y — the target variable.

The slope of the line is a, and b is the intercept (the value of y when $x = 0$). Their formulas are:

$$b = \overline{y} - a \cdot \overline{x}$$

$$a = \frac{\sum_{i=1}^{N} (y_i - \overline{y}) \cdot (x_i - \overline{x})}{\sum_{i=1}^{N} (x_i - \overline{x}) \cdot (x_i - \overline{x})}$$

[2]Features scatter plots https://github.com/apugachev/mobile-price-classification/blob/master/features_scatter_plot.pdf

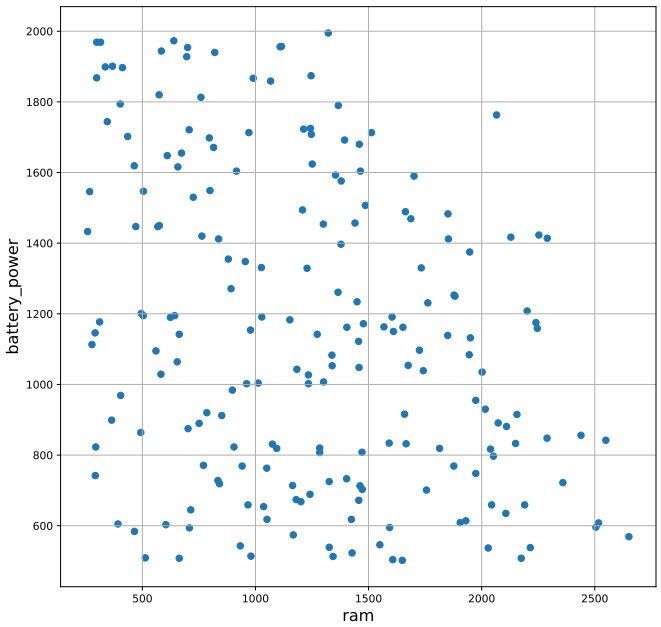

Figure 12: Scatter plot of ram and battery_power features for the first 200 items.

where:
y_i — i-th element of target vector;
x_i — i-th element of predictor vector;
$\overline{y}$ — mean value of target vector;
$\overline{x}$ — mean value of predictor vector.

Using these formulas we calculated a and b, they are equal to $a = -0.234$, $b = 1441.580$. The equation of linear regression is:

$$y = -0.234 * x + 1441.580$$

The visualization of linear regression with scatter plot of ram and battery_power features is presented on Figure 13.

The sign of the slope is negative, that means that we are dealing with inverse dependency between these features. The higher value of battery_power is, the lower value of ram will be. If we increase ram value by 100 Megabytes, the battery_power value decreases by 24 mAh.

6.3 Correlation and determinacy coefficients

Then we need to calculate correlation and determinacy coefficients. The formulas of correlation coefficient and determinacy coefficient are the following:

$$\rho = \frac{\sum_{i=1}^{N}(y_i - \overline{y})(x_i - \overline{x})/N}{\sigma(x) \cdot \sigma(y)}$$

$$\rho^2 = \rho \cdot \rho = 1 - \frac{\sum_{i=1}^{N}(y - \hat{y})^2}{\sum_{i=1}^{N}(y - \overline{y})^2}$$

where:
y_i — i-th element of target vector;

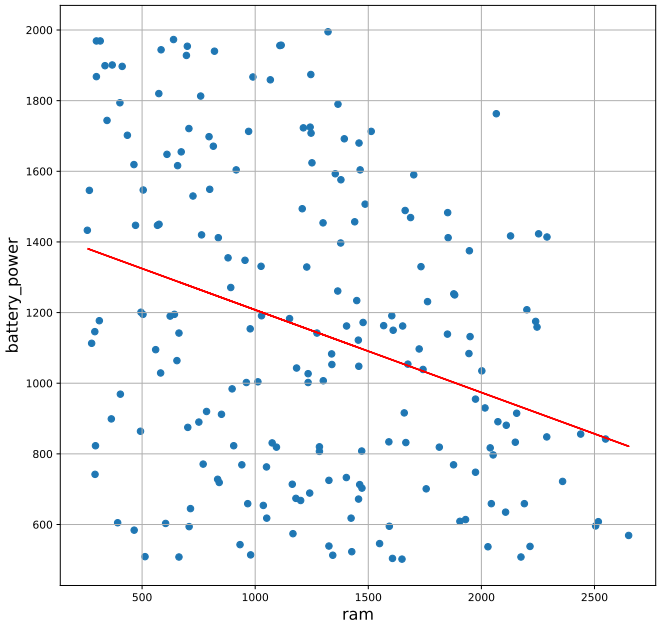

Figure 13: Visualization of linear regression.

x_i — i-th element of predictor vector;
$\overline{y}$ — mean value of target vector;
$\overline{x}$ — mean value of predictor vector;
$\sigma(y)$ — standard deviation of target vector;
$\sigma(x)$ — standard deviation of predictor vector;
$\hat{y}$ — regression residuals.

We calculated these coefficients and in our case they have these values:

$$\rho = -0.506$$
$$\rho^2 = 0.256$$

The correlation coefficient is a measure of degree of a linear relation between two features. The closer it is to 1 or -1 the more linearly dependent are features. The sign of correlation coefficient is negative, the same as the sign of the slope. That means that the features have inverse dependence, they have negative correlation. The correlations is not considered significant if its absolute value does not surpass 0.8. In our case the correlation can not be considered significant.

The coefficient of determination is a measure of how well the regression predictions approximate the real data points. This correlation is known as the "goodness of fit." A value of 1 indicates a perfect fit, and it is thus a very reliable model for future forecasts. On the other hand, a value of 0 would indicate that the model fails to accurately model the data. Based on the value of coefficient of determination that we obtained we can not say that the linear regression is a suitable model to use for the prediction of battery_power feature based on ram. The small value indicates that the dependency of these features is weak.

6.4 Prediction for three predictor's values

We have taken three items of our dataset and based on their ram feature we tried to predict their battery_power feature. The ram feature values, predicted and real battery_power values are presented in the Table 17

	ram value	Predicted battery_power	Target battery_power
1	1649	1055.845	502
2	267	1379.123	1546
3	815	1250.934	1671

Table 17: Prediction by linear regression of three predictor's values.

As we can see from the Table, the linear regression predicts the target value poorly. For example, regarding the first element, the prediction of linear regression exceeds the real target value twice. The reason for this is that these two features have weak correlation as we saw earlier.

6.5 Mean relative absolute error

Finally we need to calculate the relative absolute error of the linear regression. To calculate it we used the following formula:

$$\frac{\sum_{i=1}^{N} \frac{|y_i - ax_i - b|}{|y_i|}}{N}$$

where:
y_i — i-th element of target vector;
x_i — i-th element of predictor vector;
a, b — coefficients of the linear regression.

The relative absolute error is equal to 0.366. It means that on average, linear regression predicts values with a 36.6% error. In our opinion this is a direct consequence of low value of determinacy coefficient. This in turn means that our features correlate weakly, so we can claim that the linear regression can not be considered as a suitable model for make predictions between ram feature and battery_power feature.

7 Technical details

The code for all tasks was written on Python language version 3.6. We used the following Python libraries:

- Numpy 1.14.5
- Scikit-learn 0.19.1
- Pandas 0.23.1
- Matplotlib 2.2.2

Our code is freely available on Github[3].

8 References

[1] Mirkin B. Modern Methods of Data Analysis Lectures, 2019.

[3]Alexander Pugachev Github https://github.com/apugachev/mobile-price-classification