Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский технологический университет

«МИСиС»

Институт ИТАСУ

Кафедра Инженерной кибернетики

Лабораторная работа №2

по курсу «Нейронные сети и машинное обучение»

Выполнил:

Студент гр. МПИ-20-4-2

Малынковский О.В.

Проверил:

Курочкин И.И.

Москва 2020г.

Оглавление

Реализация

На языке Python с использованием библиотеки Sklearn реализуем обучение метод главных компонент. Библиотека feature_selector использовалась для быстрого получения списка высоко коррелирующих признаков. Библиотека fa_kit использовалась для проведения тестов Кайзера-Мейера-Олкина и Бартлетта.

Примеры работы

Датасет 1 - https://www.kaggle.com/nithin8702/foresttypes

Forest Types

Этот датасет содержит данные, полученные в результате дистанционного зондирования, в ходе которого были нанесены на карту различные типы лесов на основе их спектральных характеристик в диапазоне длин волн от видимого до ближнего инфракрасного диапазона с использованием спутниковых изображений ASTER. Выходные данные (карта типа леса) можно использовать для идентификации и / или количественной оценки экосистемных услуг (например, накопления углерода, защиты от эрозии), обеспечиваемых лесом.

Итого таблица содержит 198 строк и 28 столбцов (27 признаков, не имеющих описания и столбец с метками классов (5 классов))

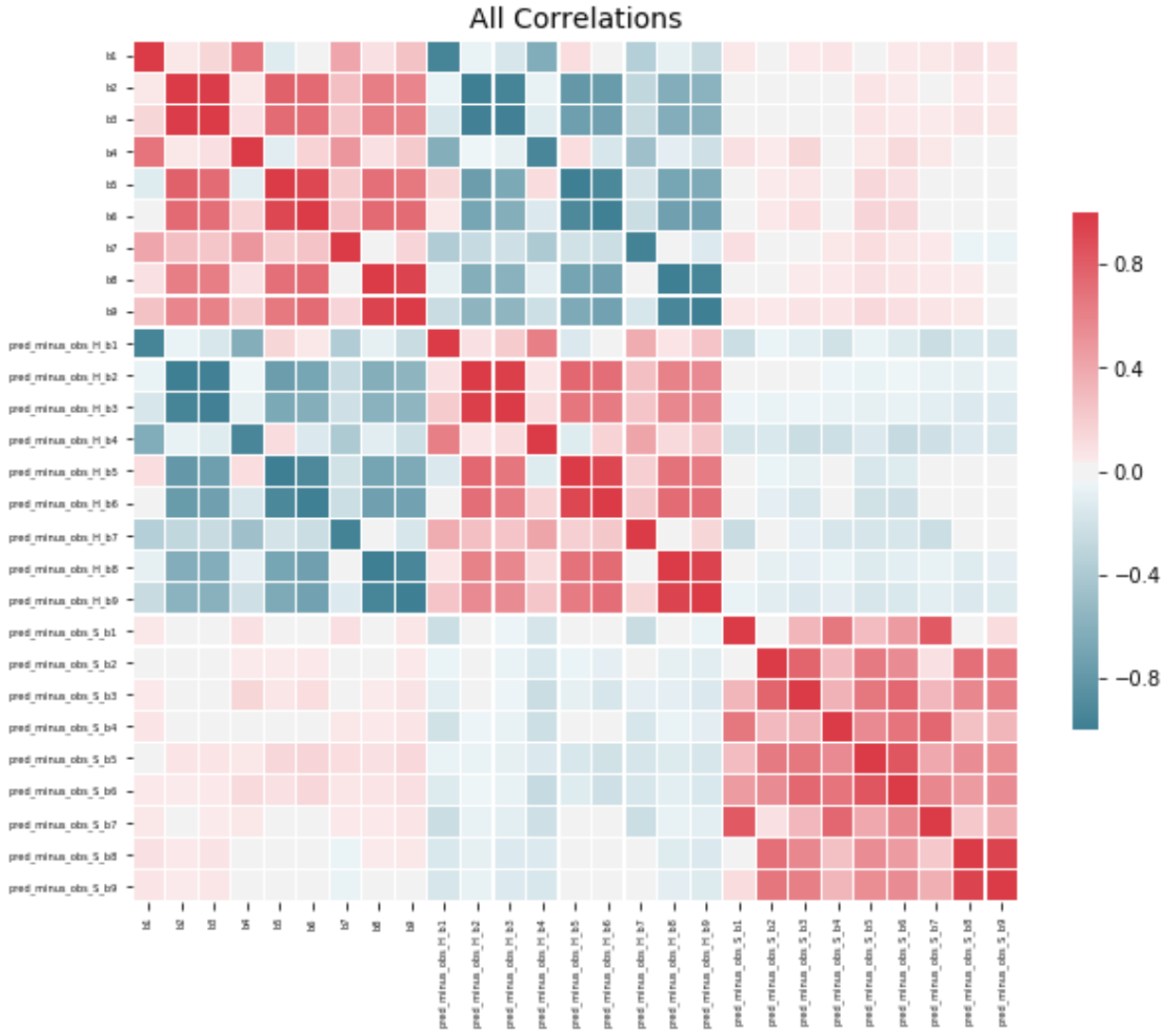
Проведем предобработку данных:

```
df_train.head()
```

class	b1	b2	b3	b4	b5	b6	b7	b8	b9	pred_minus_obs_H_b1	pred_minus_obs_H_b2	pred_mi
d	39	36	57	91	59	101	93	27	60	75.70	14.86	
h	84	30	57	112	51	98	92	26	62	30.58	20.42	
s	53	25	49	99	51	93	84	26	58	63.20	26.70	
s	59	26	49	103	47	92	82	25	56	55.54	24.50	
d	57	49	66	103	64	106	114	28	59	59.44	2.62	

Пропущенных значений не обнаружено. Построим матрицу корреляций:

	b1	b2	b3	b4	b5	b6	b7	b8	b9	pred_minus_obs_H_b1
b1	1.000000	0.062016	0.152092	0.684730	-0.119768	0.005094	0.411557	0.093572	0.261696	-0.956142
b2	0.062016	1.000000	0.981194	0.058709	0.786532	0.741036	0.276964	0.627359	0.588224	-0.068550
b3	0.152092	0.981194	1.000000	0.106463	0.729293	0.708039	0.242349	0.626275	0.603003	-0.158565
b4	0.684730	0.058709	0.106463	1.000000	-0.109407	0.167273	0.505824	0.101474	0.217717	-0.615245
b5	-0.119768	0.786532	0.729293	-0.109407	1.000000	0.926318	0.204106	0.704936	0.661180	0.153329
b6	0.005094	0.741036	0.708039	0.167273	0.926318	1.000000	0.252331	0.734751	0.727810	0.062037
b7	0.411557	0.276964	0.242349	0.505824	0.204106	0.252331	1.000000	0.019066	0.162467	-0.367251
b8	0.093572	0.627359	0.626275	0.101474	0.704936	0.734751	0.019066	1.000000	0.950395	-0.084052
b9	0.261696	0.588224	0.603003	0.217717	0.661180	0.727810	0.162467	0.950395	1.000000	-0.245010
pred_minus_obs_H_b1	-0.956142	-0.068550	-0.158565	-0.615245	0.153329	0.062037	-0.367251	-0.084052	-0.245010	1.000000
pred_minus_obs_H_b2	-0.067626	-0.992465	-0.975372	-0.044667	-0.750601	-0.691022	-0.257164	-0.610827	-0.567551	0.091417



Есть достаточное количество признаков с высокой корреляцией:

	Признак 1	Признак 2	Корреляция
0	b3	b2	0.981194
1	b6	b5	0.926318

2	b9	b8	0.950395
3	pred_minus_obs_H_b1	b1	-0.956142
4	pred_minus_obs_H_b2	b2	-0.992465
5	pred_minus_obs_H_b2	b3	-0.975372
6	pred_minus_obs_H_b3	b2	-0.959232
7	pred_minus_obs_H_b3	b3	-0.980498
8	pred_minus_obs_H_b3	pred_minus_obs_H_b2	0.974795
9	pred_minus_obs_H_b4	b4	-0.943567
10	pred_minus_obs_H_b5	b5	-0.998321
11	pred_minus_obs_H_b5	b6	-0.919764
12	pred_minus_obs_H_b6	b5	-0.923820
13	pred_minus_obs_H_b6	b6	-0.986478
14	pred_minus_obs_H_b6	pred_minus_obs_H_b5	0.926090
15	pred_minus_obs_H_b7	b7	-0.965267
16	pred_minus_obs_H_b8	b8	-0.991388
17	pred_minus_obs_H_b8	b9	-0.940071
18	pred_minus_obs_H_b9	b8	-0.945961
19	pred_minus_obs_H_b9	b9	-0.991239
20	pred_minus_obs_H_b9	pred_minus_obs_H_b8	0.951642
21	pred_minus_obs_S_b9	pred_minus_obs_S_b8	0.9463

Удалим их (если есть два коррелирующих признака, то оставим только один)

Оценим адекватность выборки для применения факторного анализа:

```
kmo_all,kmo_model=calculate_kmo(df_tr.iloc[:,1:])
print(kmo_model)
```

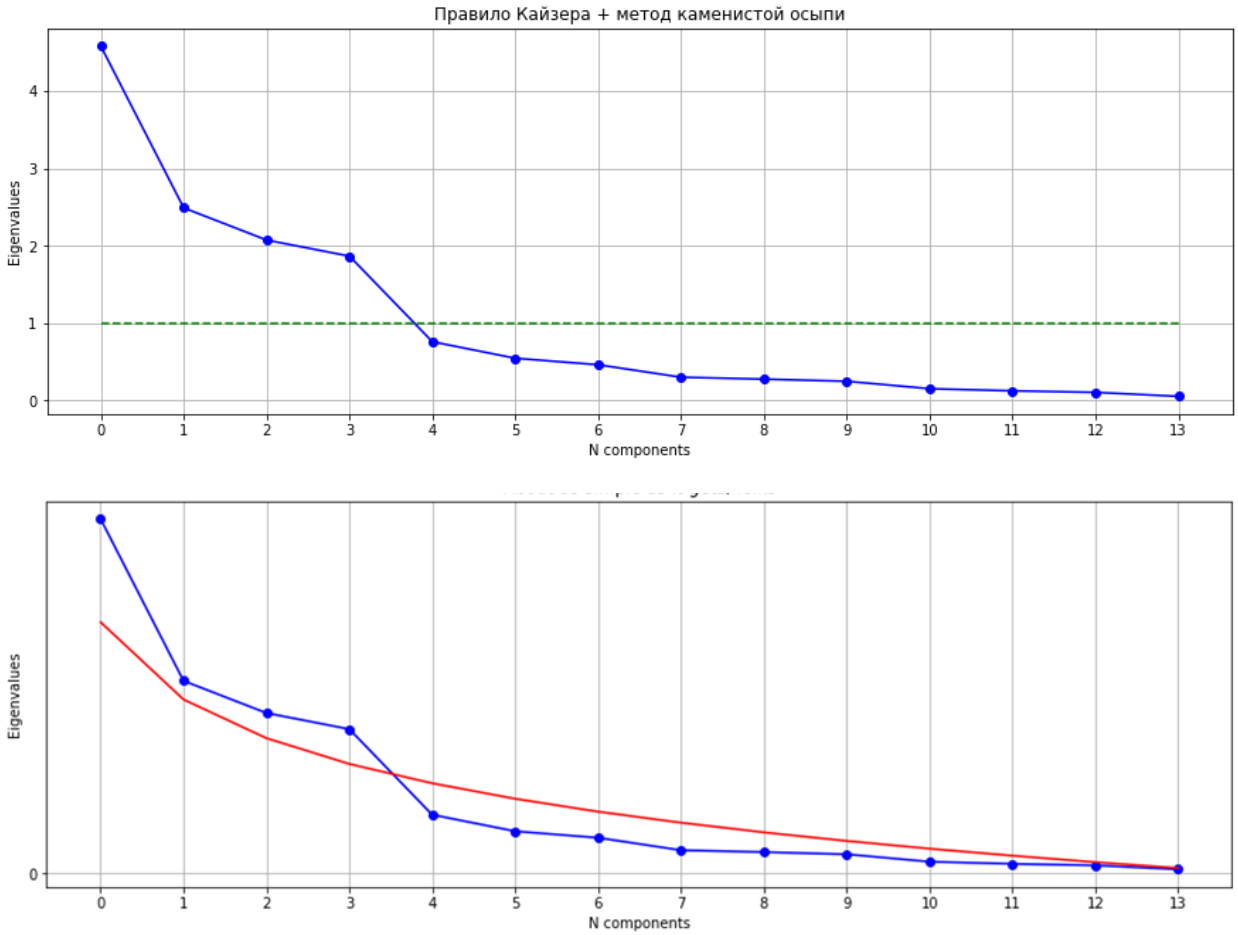
```
0.6619941735706738
```

```
chi_square_value,p_value=calculate_bartlett_sphericity(d
chi_square_value, p_value
```

```
(2091.3495835412596, 0.0)
```

Критерий Кайзера-Мейера-Олкина показывает удовлетворительную адекватность (0,67). Уровень значимости для критерия сферичности Бартлетта много меньше 0.05. Можно сделать вывод, о том что применения факторного анализа имеет смысл.

Приступим к понижению размерности данных методом главных компонент. Для начала определим число по критериям:



По правилу Кайзера, по методу каменистой осыпи, и по правилу сломанной трости берем 5 компонент (на графике 4, но там нумерация идет с нуля)

Построим матрицу факторных нагрузок:

	0	1	2	3
b1	0.136235	0.222087	-0.707849	0.419699
b2	0.127720	0.879365	0.157756	-0.095503
b4	0.174288	0.242820	-0.728995	0.441657
b5	0.132734	0.858121	0.346620	-0.164666
b7	0.133919	0.424237	-0.569496	0.232186
b8	0.151287	0.796352	0.191620	-0.121723
pred_minus_obs_S_b1	0.580252	-0.075645	-0.375131	-0.612151
pred_minus_obs_S_b2	0.687816	-0.109545	0.354766	0.490475
pred_minus_obs_S_b3	0.805668	-0.088006	0.175150	0.295887
pred_minus_obs_S_b4	0.757978	-0.107032	-0.184264	-0.411670
pred_minus_obs_S_b5	0.856221	-0.018436	0.171986	0.130148
pred_minus_obs_S_b6	0.916659	-0.066030	0.037323	-0.025575
pred_minus_obs_S_b7	0.682182	-0.099587	-0.287104	-0.558963
pred_minus_obs_S_b8	0.624172	-0.097025	0.306338	0.435841

Можно вычислить вклады факторов в общую дисперсию всех признаков (путем вычисления суммы квадратов факторных нагрузок для каждого фактора по всем признакам). Чем выше доля этого вклада в общей дисперсии, тем более значимым является данный фактор.

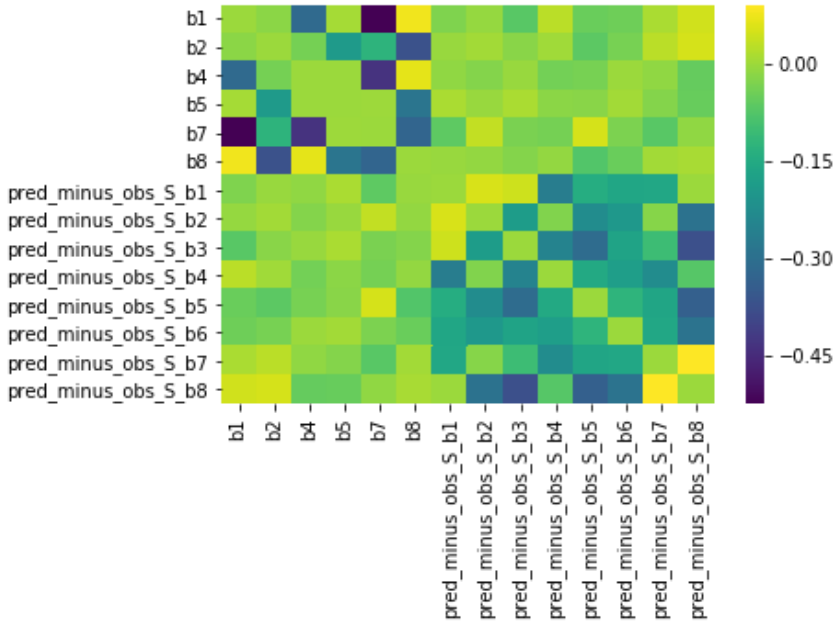
Суммарная нагрузка для признаков:

```
b5                    0.901250
pred_minus_obs_S_b7   0.870159
pred_minus_obs_S_b1   0.857867
pred_minus_obs_S_b2   0.851516
pred_minus_obs_S_b6   0.846670
b2                    0.823602
b4                    0.815832
pred_minus_obs_S_b4   0.789412
pred_minus_obs_S_b5   0.779972
pred_minus_obs_S_b3   0.775073
b1                    0.745080
b8                    0.708599
pred_minus_obs_S_b8   0.682805
b7                    0.576147
```

‘b5’,’pred_minus_obs_S_b7’, ’pred_minus_obs_S_b1’, ’pred_minus_obs_S_b2’ являются наиболее значимыми. (Хотя влияние почти всех остальных тоже велико).

Анализ остатков:

Построим матрицу остатков корреляций, где показана разница между воспроизведенной и реальной корреляциями.



В целом все признаки были воспроизведены удовлетворительно, за исключением ‘b7’, ‘b4’, где остатки выше 0,3. Можно сделать вывод о том, что применение МГК оказалось удачным.

Датасет 2 - https://www.kaggle.com/mattcarter865/mines-vs-rocks

Connectionist Bench (Sonar, Mines vs. Rocks) Data Set

Данные представляют собой таблицу измерений 8 физических параметров морских ракушек. По этим данным нужно обучить сеть различать сигналы сонара, отражаемые от металлического цилиндра, и сигналы, отраженные от скалы цилиндрической формы.

Передаваемый сигнал гидролокатора представляет собой частотно-модулированный щебетание с нарастающей частотой. Набор данных содержит сигналы, полученные под разными углами обзора: 90 градусов для цилиндра и 180 градусов для горных пород.

Каждая строка представляет собой набор из 60 чисел в диапазоне от 0,0 до 1,0. Каждое число представляет энергию в определенной полосе частот, интегрированную за определенный период времени.

Метка, связанная с каждой записью, содержит букву «R», если объект является камнем, и «M», если это металлический цилиндр. Цифры на этикетках расположены в порядке возрастания угла обзора, но они не кодируют угол напрямую.

Итого таблица содержит 208 строк, 60 признаков.

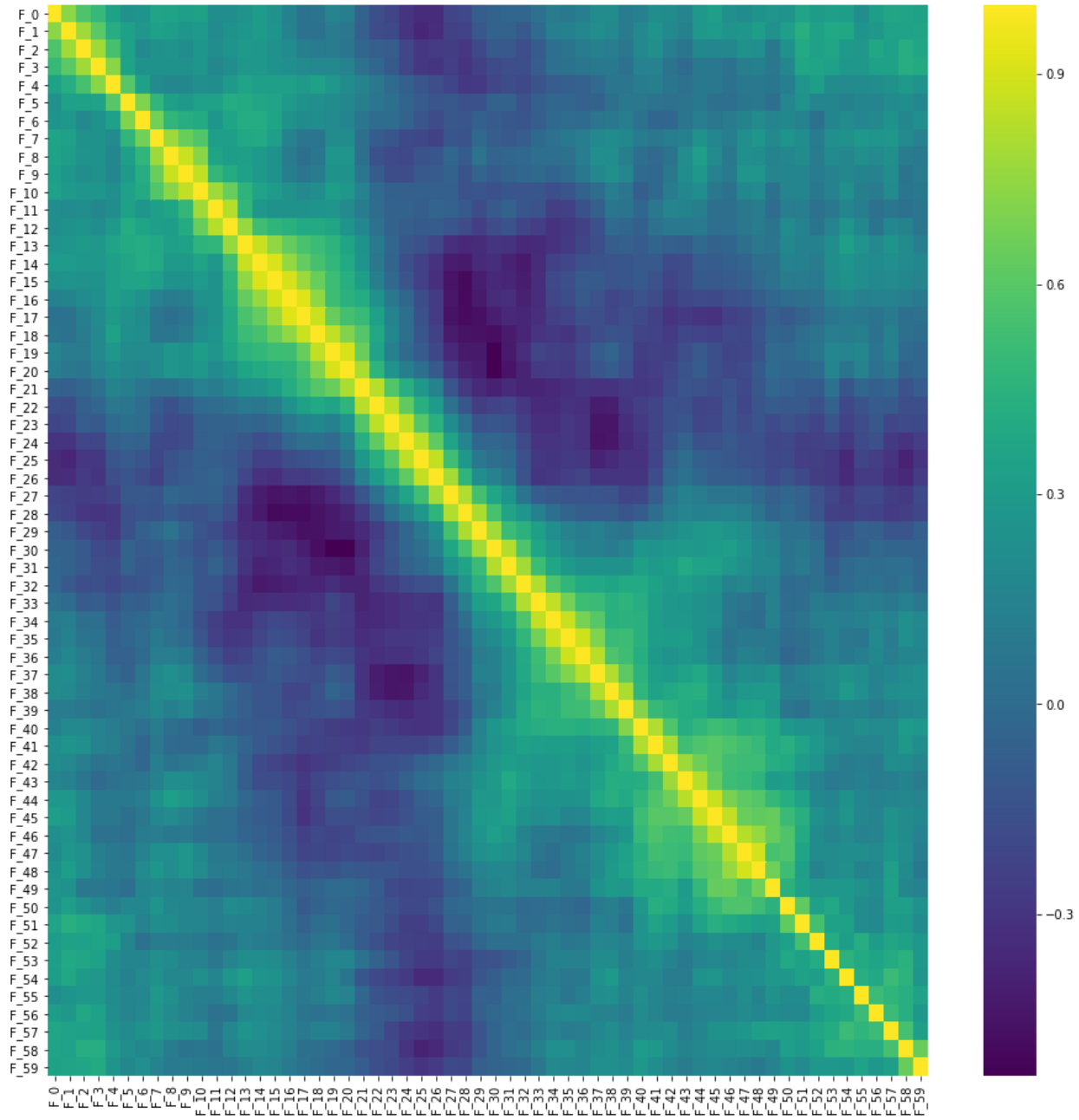
Проведем предобработку данных:

```
df2.head()
```

	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_10	F_11	F_12	F_13
0	0.0200	0.0371	0.0428	0.0207	0.0954	0.0986	0.1539	0.1601	0.3109	0.2111	0.1609	0.1582	0.2238	0.0645
1	0.0453	0.0523	0.0843	0.0689	0.1183	0.2583	0.2156	0.3481	0.3337	0.2872	0.4918	0.6552	0.6919	0.7797
2	0.0262	0.0582	0.1099	0.1083	0.0974	0.2280	0.2431	0.3771	0.5598	0.6194	0.6333	0.7060	0.5544	0.5320
3	0.0100	0.0171	0.0623	0.0205	0.0205	0.0368	0.1098	0.1276	0.0598	0.1264	0.0881	0.1992	0.0184	0.2261
4	0.0762	0.0666	0.0481	0.0394	0.0590	0.0649	0.1209	0.2467	0.3564	0.4459	0.4152	0.3952	0.4256	0.4135

Пропущенных значений не обнаружено. Построим матрицу корреляций:

	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	
F_0	1.000000	0.735896	0.571537	0.491438	0.344797	0.238921	0.260815	0.355523	0.35
F_1	0.735896	1.000000	0.779916	0.606684	0.419669	0.332329	0.279040	0.334615	0.31
F_2	0.571537	0.779916	1.000000	0.781786	0.546141	0.346275	0.190434	0.237884	0.25
F_3	0.491438	0.606684	0.781786	1.000000	0.726943	0.352805	0.246440	0.246742	0.24
F_4	0.344797	0.419669	0.546141	0.726943	1.000000	0.597053	0.335422	0.204006	0.17
F_5	0.238921	0.332329	0.346275	0.352805	0.597053	1.000000	0.702889	0.471683	0.32
F_6	0.260815	0.279040	0.190434	0.246440	0.335422	0.702889	1.000000	0.675774	0.47
F_7	0.355523	0.334615	0.237884	0.246742	0.204006	0.471683	0.675774	1.000000	0.77
F_8	0.353420	0.316733	0.252691	0.247078	0.177906	0.327578	0.470580	0.778577	1.00
F_9	0.318276	0.270782	0.219637	0.237769	0.183219	0.288621	0.425448	0.652525	0.87
F_10	0.344058	0.297065	0.274610	0.271881	0.231684	0.333570	0.396588	0.584583	0.72



Есть достаточное 18 признаков с высокой корреляцией:

	Признак 1	Признак 2	Корреляция
0	F_9	F_8	0.877131
1	F_10	F_9	0.853140
2	F_14	F_13	0.869733
3	F_15	F_14	0.912625
4	F_16	F_15	0.899234
5	F_17	F_16	0.925836
6	F_18	F_17	0.875028

7	F_19	F_18	0.854572
8	F_20	F_19	0.905062
9	F_23	F_22	0.850294
10	F_24	F_23	0.853140
11	F_26	F_25	0.857242
12	F_34	F_33	0.853365
13	F_35	F_34	0.873337
14	F_36	F_35	0.886030
15	F_45	F_44	0.869090
16	F_46	F_45	0.861515
17	F_48	F_47	0.895313

Удалим их (если есть два коррелирующих признака, то оставим только один)

Оценим адекватность выборки для применения факторного анализа:

```
from factor_analyzer.factor_analyzer import calculate_kmo
kmo_all,kmo_model=calculate_kmo(df2.iloc[:,:-1])
print(kmo_model)
```

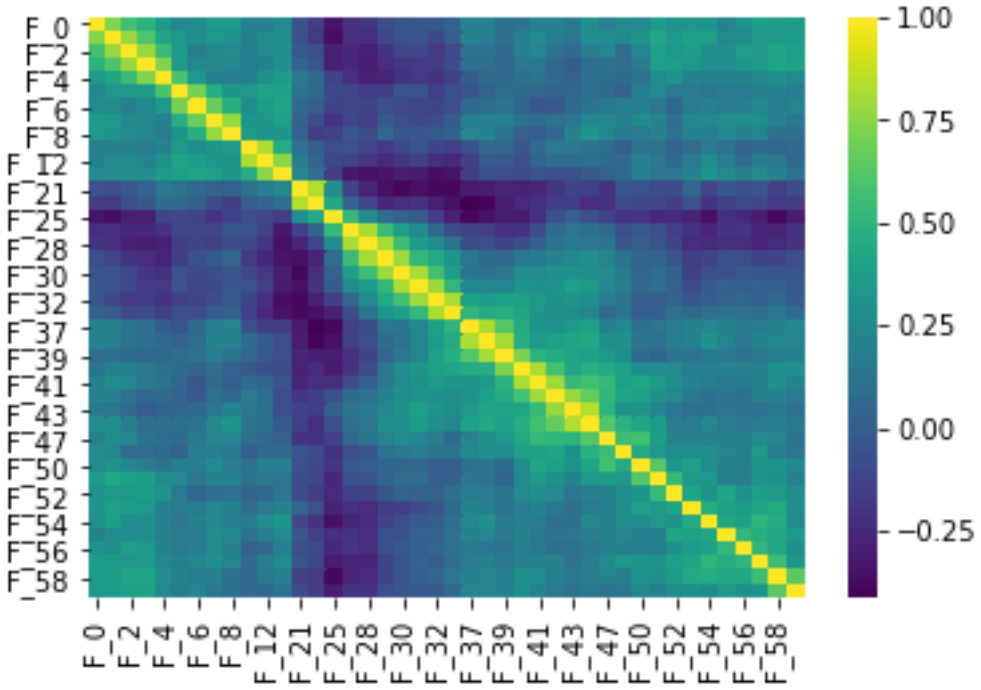
```
0.7523322448128427
/usr/local/lib/python3.6/dist-packages/factor_analyzer/uti
  warnings.warn('The inverse of the variance-covariance ma
```

```
from factor_analyzer.factor_analyzer import calculate_bart
chi_square_value,p_value=calculate_bartlett_sphericity(df2
chi_square_value, p_value
```

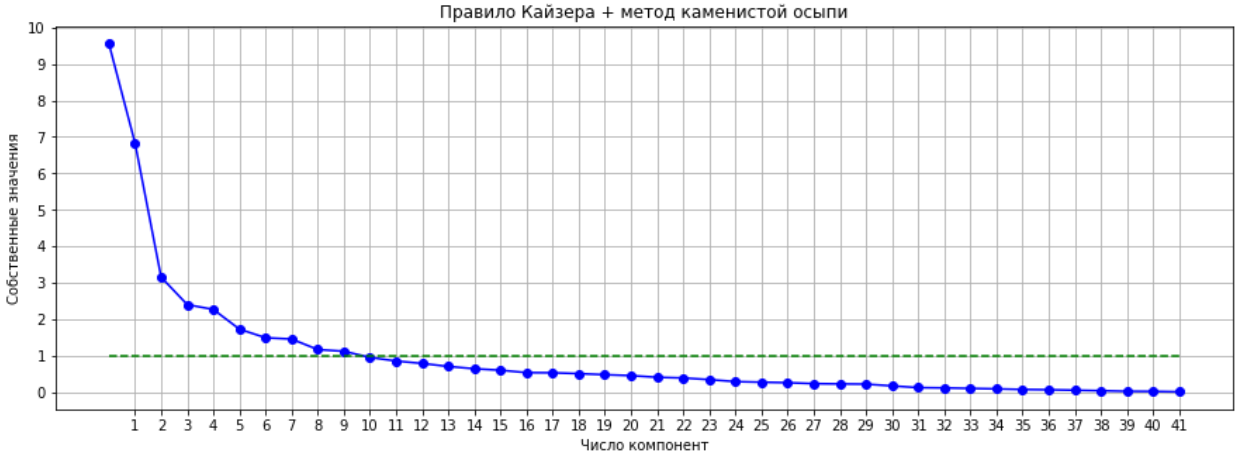
```
(15801.472783241321, 0.0)
```

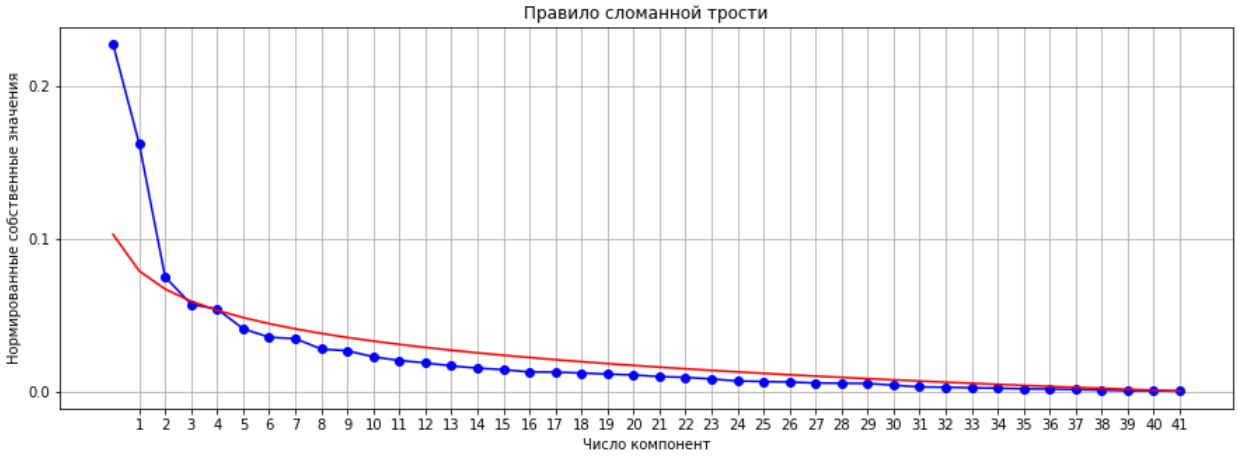
Критерий Кайзера-Мейера-Олкина показывает высокую адекватность (0,75). Уровень значимости для критерия сферичности Бартлетта много меньше 0.05. Можно сделать вывод, о том, что применения факторного анализа имеет смысл.

Матрицы корреляций после удаления высоко коррелирующих признаков:



Приступим к понижению размерности данных методом главных компонент. Для начала определим число по критериям:





По правилу Кайзера и по методу каменистой осыпи берем 9 компонент, по методы каменной осыпи – 3, по правилу сломанной трости – 5.

Построим матрицу факторных нагрузок для 9 компонент:

F_0	0.613323	0.220906	-0.030290	0.077258	0.210616	0.196610	-0.159136	0.145747	-0.068823
F_1	0.666683	0.271896	-0.032062	0.153684	0.217240	0.300505	-0.198090	0.125110	-0.017831
F_2	0.610253	0.370902	0.025804	0.228426	0.196301	0.423341	-0.061475	0.041839	-0.134821
F_3	0.556829	0.432533	0.074538	0.204149	0.215185	0.434324	0.067800	0.072701	-0.087497
F_4	0.440880	0.452584	-0.080991	0.069485	0.056324	0.479325	0.230362	0.009358	0.014199
F_5	0.392217	0.385036	-0.339053	-0.248673	0.161265	0.240803	0.328026	-0.013361	0.269774
F_6	0.389205	0.337137	-0.344606	-0.405148	0.187227	-0.065443	0.221434	0.208878	0.279870
F_7	0.496865	0.251597	-0.338678	-0.442375	0.166588	-0.202251	0.038953	0.326158	0.135932
F_8	0.438571	0.179353	-0.368165	-0.430388	0.211958	-0.156329	-0.050640	0.327926	-0.040529
F_11	0.277534	0.307239	-0.437202	-0.380757	-0.082983	0.063105	-0.140049	-0.470175	-0.203810
F_12	0.313895	0.461991	-0.438651	-0.311299	-0.087783	-0.041370	-0.124399	-0.538789	-0.121952
F_13	0.316422	0.622590	-0.245960	-0.220417	-0.053886	-0.143929	-0.053300	-0.329369	0.010984
F_21	-0.221019	0.516021	-0.192053	0.083524	-0.433675	-0.037700	0.428959	0.157803	-0.018816
F_22	-0.341953	0.377681	-0.226225	0.208887	-0.397451	0.029995	0.533403	0.108983	0.017692
F_25	-0.522627	-0.029641	-0.433867	0.163581	-0.177659	0.021076	0.255648	0.151964	-0.288599
F_27	-0.324925	-0.386806	-0.560316	0.200323	0.260281	-0.016184	-0.006703	0.144921	-0.314128
F_28	-0.268662	-0.568324	-0.489636	0.162445	0.370836	-0.011323	-0.057271	0.035302	-0.127555
F_29	-0.031559	-0.692872	-0.353579	0.091115	0.428134	-0.049752	0.035943	0.057919	-0.003714
F_30	0.071264	-0.711323	-0.273445	0.091845	0.357981	0.055712	0.187572	-0.203241	0.114241
F_31	0.154888	-0.716178	-0.135705	-0.013409	0.217572	0.057502	0.303340	-0.281895	0.079372
F_32	0.180422	-0.737248	0.139771	-0.173403	0.158575	0.033874	0.300512	-0.219267	0.121285
F_33	0.267344	-0.619731	0.324591	-0.234119	0.152329	0.027107	0.254676	-0.094782	0.109756
F_37	0.500662	-0.406300	0.338841	-0.336531	-0.013453	0.015659	-0.077087	0.288562	-0.147988
F_38	0.550797	-0.384692	0.263698	-0.412610	-0.132563	0.016738	-0.077988	0.182815	-0.205112
F_39	0.506529	-0.415290	0.250112	-0.422622	-0.298249	0.135557	0.059638	0.085869	-0.023624
F_40	0.591567	-0.447453	0.166692	-0.019232	-0.310717	0.172873	0.113384	-0.067762	0.164616
F_41	0.602754	-0.429063	-0.046478	0.164082	-0.375880	0.198908	-0.028101	-0.157172	0.090821
F_42	0.539725	-0.468316	-0.239533	0.113700	-0.430167	0.123948	-0.011993	-0.080949	-0.121934
F_43	0.475961	-0.478193	-0.338925	-0.036086	-0.343890	0.067042	0.038638	0.044689	-0.286163
F_44	0.592600	-0.427957	-0.356995	0.018194	-0.266253	0.017473	-0.065659	0.152444	-0.163977
F_47	0.560433	-0.305921	-0.321118	0.187396	-0.202405	-0.130677	-0.152164	0.111097	0.100621
F_49	0.516642	-0.201815	-0.136577	0.170656	-0.251328	-0.293693	-0.036700	0.145047	0.236023
F_50	0.553468	-0.030920	-0.285657	0.348469	-0.154560	-0.237524	-0.198838	-0.000171	0.288255
F_51	0.572673	0.038791	-0.191841	0.394705	0.041385	-0.000430	-0.167591	-0.006296	0.321097

F_52	0.533908	0.063956	0.097755	0.370452	0.077647	-0.056966	0.027347	-0.062319	0.058287
F_53	0.547702	0.201104	0.126387	0.214478	0.007386	-0.274085	0.043290	0.048737	-0.045524
F_54	0.573061	0.166591	0.150302	0.040589	0.092751	-0.366793	-0.064999	-0.168790	-0.105629
F_55	0.531396	0.139352	0.135296	0.067769	0.109765	-0.380165	0.245534	-0.068115	-0.274398
F_56	0.506839	0.145847	0.167431	0.144135	0.191349	-0.276391	0.285015	0.019977	-0.194120
F_57	0.601650	0.178256	0.130137	0.157319	0.095293	-0.352021	0.189951	-0.039328	-0.041395
F_58	0.625496	0.140408	0.256005	0.197860	0.133539	-0.106404	0.171747	-0.145496	-0.152370
F_59	0.515191	0.120106	0.223300	0.133921	0.130081	0.070456	0.122569	0.001480	-0.21012

Можно вычислить вклады факторов в общую дисперсию всех признаков (путем вычисления суммы квадратов факторных нагрузок для каждого фактора по всем признакам). Чем выше доля этого вклада в общей дисперсии, тем более значимым является данный фактор.

Суммарная нагрузка для признаков:

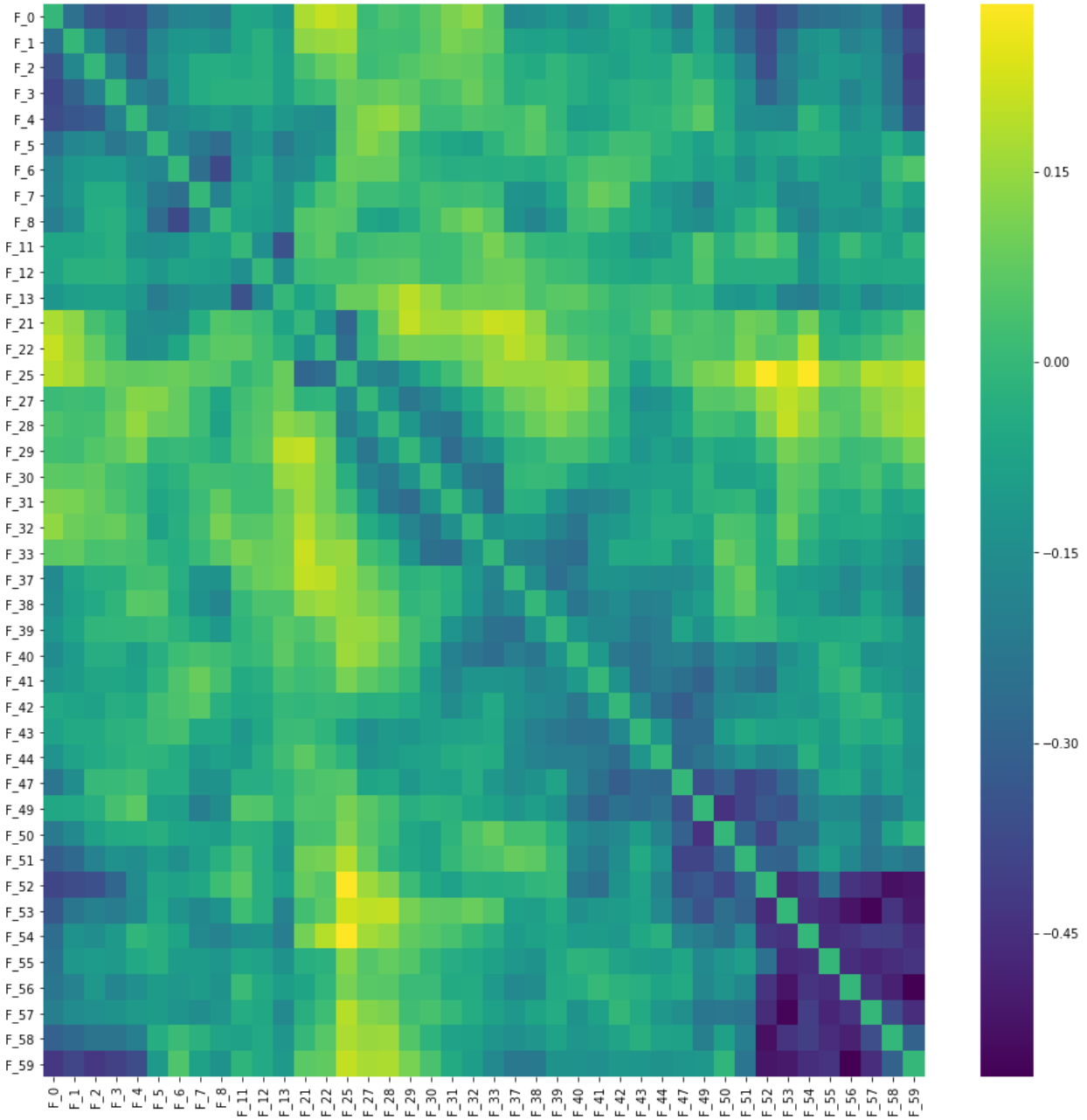
```
F_12    0.931346
F_28    0.819748
F_7     0.815606
F_30    0.815062
F_22    0.809962
F_32    0.805080
F_29    0.804822
F_2     0.804280
F_42    0.802898
F_11    0.800629
F_27    0.797010
F_3     0.796857
F_41    0.791081
F_38    0.790576
F_39    0.789020
F_44    0.787733
F_31    0.783920
F_43    0.779514
F_6     0.758367
F_21    0.757744
F_37    0.755346
F_40    0.749299
F_5     0.743432
F_1     0.735747
F_13    0.731886
F_8     0.726398
F_33    0.725539
F_50    0.713247
F_4     0.696876
F_25    0.692762
F_51    0.654996
F_47    0.649572
F_55    0.621493
F_58    0.618684
F_57    0.607784
F_49    0.582939
```

```
F_54    0.567402
F_0     0.566169
F_56    0.559287
F_53    0.483894
F_52    0.453241
F_59    0.428706
```

‘F_12’,’F_28’,’ F_27’ и до ‘F_16’ являются наиболее значимыми. (Хотя влияние почти всех остальных тоже велико). Самое слабое влияние у ‘F_53’,’ F_52’,’ F_59’

Анализ остатков:

Построим матрицу остатков корреляций, где показана разница между воспроизведенной и реальной корреляциями.

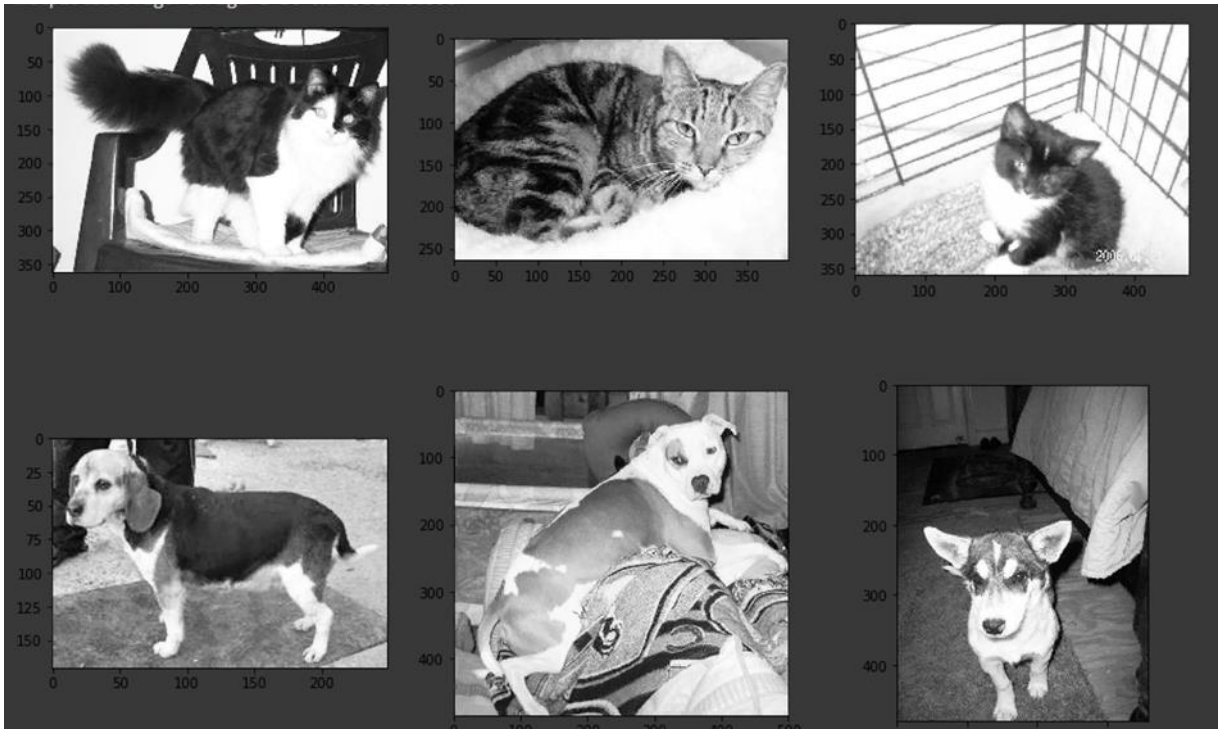


В целом более 15 признаков воспроизведены плохо, остатки выше 0,30. Можно сделать вывод о том, что применение МГК оказалось неудачным.

Датасет 3 – https://www.kaggle.com/chetankv/dogs-cats-images

Dogs & Cats Images

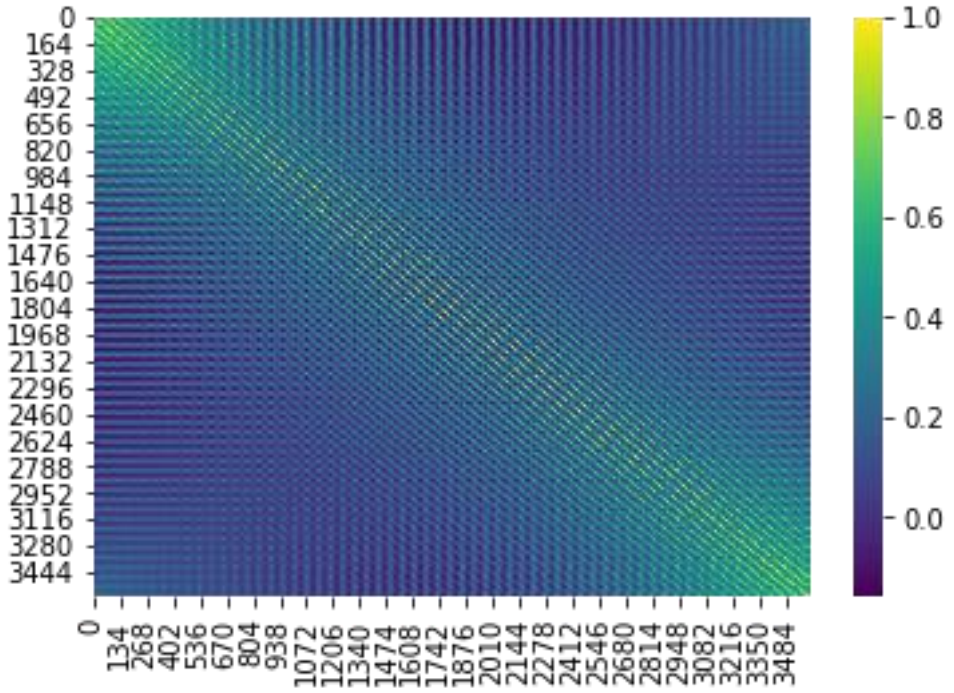
Датасет содержит 4000 тысячи изображений собак и 4000 тысячи изображений кошек. Размер одной картинки 60 x 60. Для простоты возьмем изображения переведены в градацию серого, иначе бы пришлось использовать несколько измененный МГК.



Проведем предобработку данных, все пиксели из матрицы переводим в строку. Тогда каждая строка представляет собой картинку

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	101	104	107	109	106	106	106	106	103	107	102	106	106	105	108	106	107	106	98	90	103	110
1	95	102	109	115	127	148	179	189	188	188	159	159	184	196	189	177	181	142	138	122	170	165
2	199	176	136	199	199	198	197	197	198	198	186	121	200	201	201	200	200	200	198	184	119	203
3	1	5	6	7	9	4	9	11	16	12	5	5	1	2	2	2	5	10	6	7	12	71
4	181	185	190	193	183	102	183	191	190	192	189	195	190	46	193	192	196	198	201	201	209	207

Построим матрицу корреляций:



Если и есть корреляция, то только положительная.

Есть достаточное 18 признаков с высокой корреляцией:

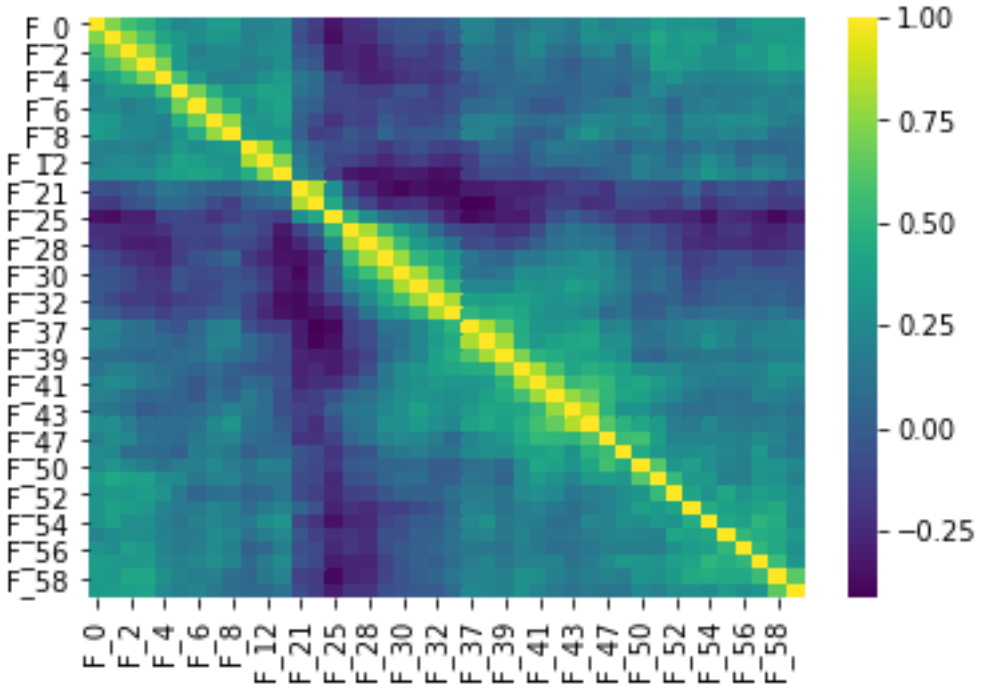
Удалим их (если есть два коррелирующих признака, то оставим только один)

Оценим адекватность выборки для применения факторного анализа:

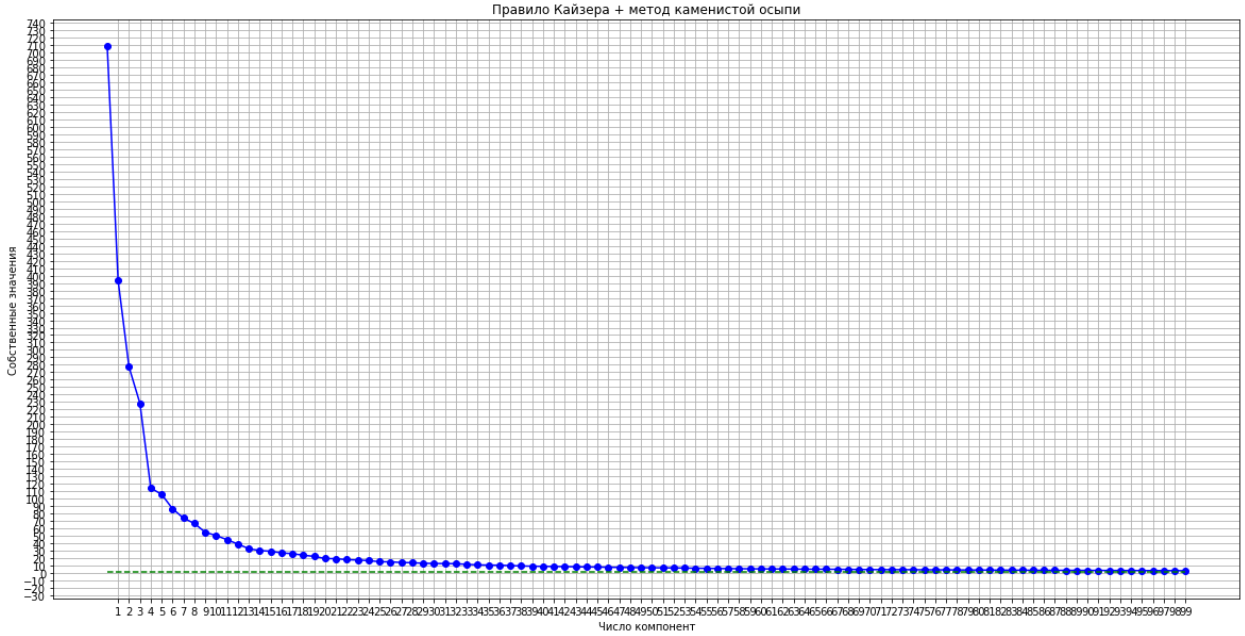
```
7] kmo_all,kmo_model=calculate_kmo(df3)
   print(kmo_model)

   /usr/local/lib/python3.6/dist-packages/n
     r = _umath_linalg.det(a, signature=sig
   0.9940661510600819

3] chi_square_value,p_value=calculate_bartl
   chi_square_value, p_value

   (7500.9510562840005, 0.0)
```

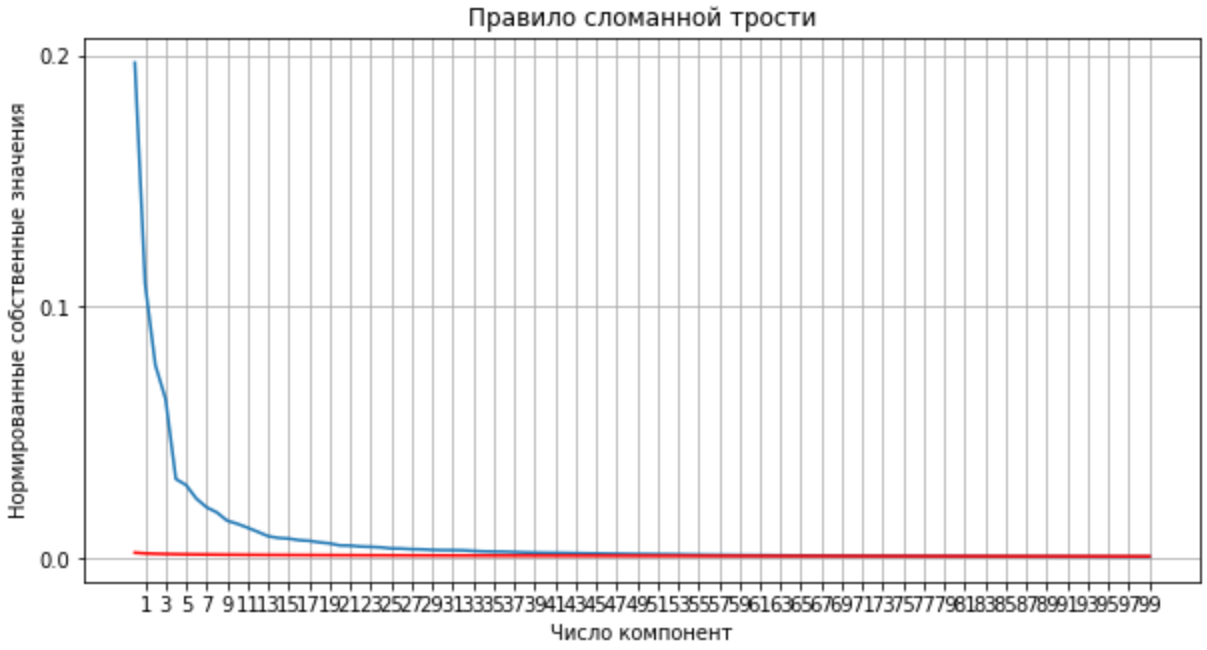
Критерий Кайзера-Мейера-Олкина показывает высокую адекватность (почти 1). Уровень значимости для критерия сферичности Бартлетта много меньше 0.05. Можно сделать вывод, о том, что применения факторного анализа имеет смысл.

Матрицы корреляций после удаления высоко коррелирующих признаков:



Приступим к понижению размерности данных методом главных компонент. Для начала определим число по критериям:





Признаков на столько много, что графически можно было определить число компонент только по методу каменной осыпи – 4.

Два других критерия посчитаны через вручную:

```
] t = pca.explained_variance_[pca.explained_variance_ > 1]
  len(t)

  219

] values = pca.explained_variance_
  e = calc_broken_stick(len(values))
  t = values/len(values)

  a = np.where(e - t > 0)
  a[0][0]

  78
```

По правилу Кайзера берем 219 компонент, по методы каменной осыпи – 3, по правилу сломанной трости – 78.

Построим матрицу факторных нагрузок для 219 компонент:

	0	1	2	3	4	5	6	7	8	9	10	
0	0.535763	0.412027	-0.076514	-0.157078	0.100747	-0.179401	-0.196318	-0.150442	0.179195	0.003149	0.035184	-0
1	0.546372	0.413020	-0.087588	-0.158730	0.104146	-0.186746	-0.195969	-0.152691	0.172327	0.002072	0.032281	-0
2	0.550537	0.418752	-0.092166	-0.163457	0.109962	-0.190887	-0.194674	-0.146420	0.164583	-0.004960	0.036214	-0
3	0.551003	0.423021	-0.099560	-0.162556	0.115570	-0.188342	-0.201377	-0.146582	0.163961	-0.000312	0.044732	-0
4	0.552945	0.422889	-0.105759	-0.158119	0.127468	-0.195438	-0.202897	-0.145862	0.156889	-0.006426	0.047077	-0
5	0.556386	0.424972	-0.115573	-0.157264	0.133058	-0.194605	-0.200249	-0.139223	0.149831	-0.010237	0.055979	-0
6	0.562328	0.424094	-0.115667	-0.159795	0.134363	-0.196753	-0.197947	-0.137527	0.139149	-0.014584	0.057154	-0
7	0.559465	0.421922	-0.123220	-0.165592	0.137356	-0.204610	-0.199207	-0.133402	0.137223	-0.017330	0.067678	-0
8	0.563383	0.423381	-0.121473	-0.166837	0.152410	-0.203194	-0.200449	-0.121560	0.128508	-0.015744	0.064773	-0
9	0.566100	0.427829	-0.129677	-0.150752	0.159995	-0.201204	-0.196269	-0.112303	0.120446	-0.020508	0.069333	-0
10	0.566178	0.421986	-0.137180	-0.145494	0.165001	-0.193998	-0.186139	-0.107675	0.110583	-0.022369	0.080640	-0

Всю матрицу в отчете приводить не будем в силу её слишком большого размера.

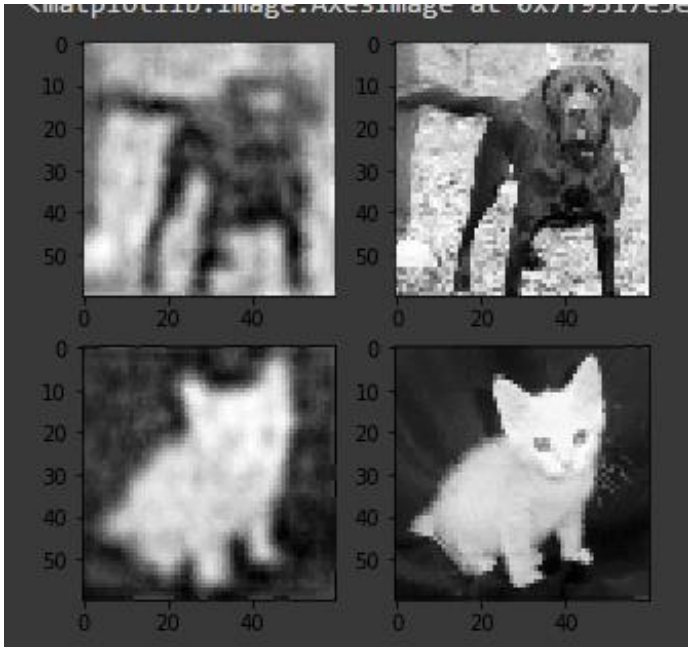
Можно вычислить вклады факторов в общую дисперсию всех признаков (путем вычисления суммы квадратов факторных нагрузок для каждого фактора по всем признакам). Чем выше доля этого вклада в общей дисперсии, тем более значимым является данный фактор.

Суммарная нагрузка для признаков:

```
3536    0.914223
121     0.913988
62      0.913618
181     0.913618
3537    0.913266
61      0.913177
179     0.913085
3421    0.912866
241     0.912860
3482    0.912540
301     0.912536
          ...
1042    0.856460
1237    0.856441
1352    0.856259
1160    0.855736
1049    0.855592
1219    0.855530
863     0.855144
1108    0.854529
1059    0.854103
1176    0.853294
1101    0.852837
1048    0.852280
```

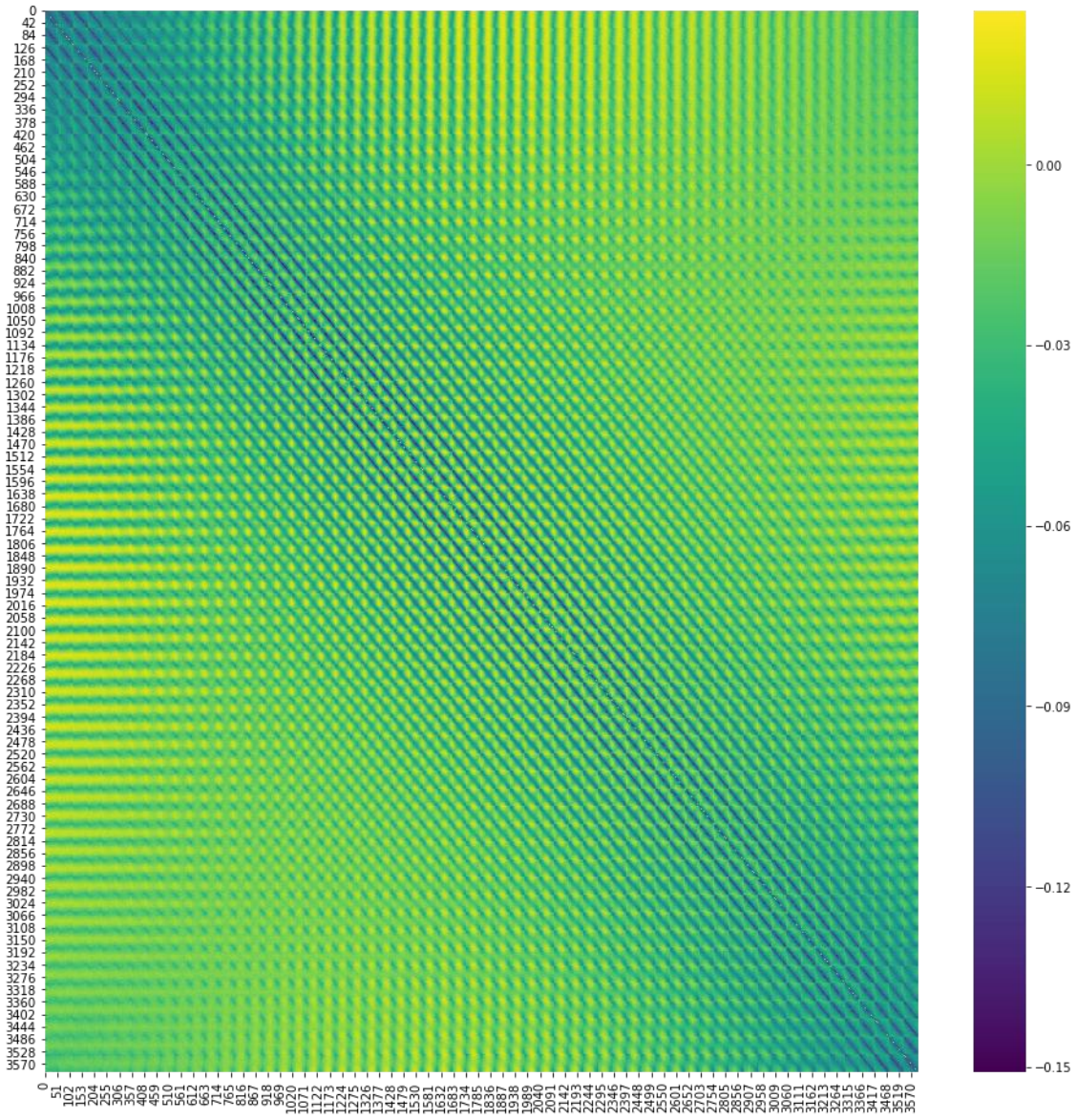
Почти каждый пиксель вносит существенный вклад.

Покажем пример сжатого (восстановленного) изображения (справа оригинал):



Анализ остатков:

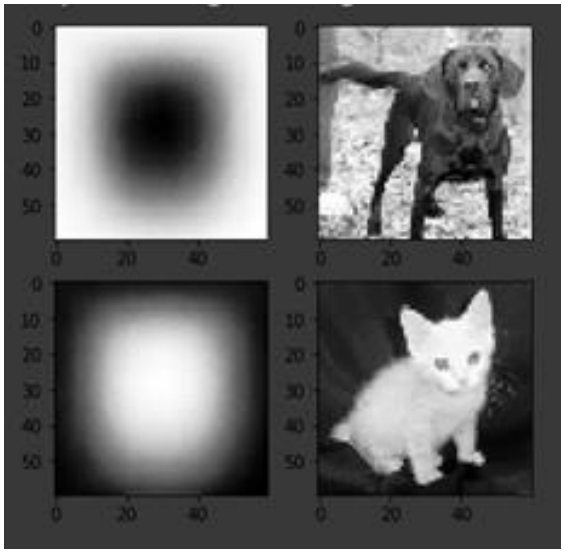
Построим матрицу остатков корреляций, где показана разница между воспроизведенной и реальной корреляциями.



В целом все признаки были воспроизведены удовлетворительно. Можно сделать вывод о том, что применение МГК оказалось очень удачным.

Покажем результат сжатия картинок при 78 компонентах (в целом очертания еще видны, но отличить кошку от собаки уже очень проблематично):

Меньшее число компонент не имеет особого смысла тестировать. Потери информации будут существенные, что после этого будет сложно обучить какой-нибудь классификатор. Покажем результат сжатия картинок 3 компонентах, но не будем считать разницу воспроизведенной и исходной корреляции:



Столь маленькое число компонент ожидаемо породило огромную потерю информации, что, смотря на изображение ясно только что светлее – объект или фон, но не более.

Вывод

В ходе выполнения лабораторной работы были рассмотрен метод главных компонент для понижения размерности данных выборки, критерий выбора числа этих компонент.

В результате использования его для трех датасетов было установлено, что МГК действительно может быть использован для понижения размерности за счет использования линейных комбинаций исходных признаков.

Для первого датасета применение МГК было удачным, большинство признаков могли быть успешно воспроизведены обратным преобразованием. Все три критерия указали на одинаковое число главных компонент.

Во втором датасете понижение размерности не было успешным около 15 из 60 признаков стало плохо воспроизводимыми. При анализе остатков обнаружено много больших чисел в матрице остатков корреляции. Критерии выбора числа главных компонент давали разные результаты. Больше всего компонент было сохранено по правилу Кайзера, меньше всего – по методу каменной осыпи.

Для третьего датасета с картинками МГК прекрасно показал себя для сжатия изображений. По правилу Кайзера предлагалось использование 219 главных компонент, по методы каменной осыпи – 3, по правилу сломанной трости – 78. В первом случае получены отличные результаты: разница между воспроизведенной и реальной корреляциями почти нигде не превышает 0.15. Для 78 компонент результаты похуже, но на сжатых картинках хотя бы еще видно очертания предметов. Для случая с всего 3 компонентами (при более 3000 исходных признаков) видно только то, что светлее – объект или фон, но не более.

Если сравнивать критерии выбора числа компонент, то хуже всего себя показывает метод каменной осыпи, он сильно занижает их количество. Также он в целом будет бесполезен тем больше, чем более плавно будет идти линия на графике с собственными числами.

Остальные критерии показали себя адекватно, и хотя правило сломанной трости немного завышает число компонент, это не так критично, ведь потеря объясненной дисперсии критичнее чем сохранение лишней информации.

Также стоит отметить, что несмотря на наличии матрицы факторной нагрузки и простоты линейной модели для получения компонент, интерпретация получаемых компонент затруднительная. С ростом числа признаков в исходной выборке проблема только усиливается.

Инструкция по запуску

Для запуска потребуется использовать Anaconda (Jupiter Notebook), но предпочтительнее запускать из Google Colab (https://colab.research.google.com/drive/160bpRY_fN34WgBnUvlOY1G8CaCVEdB4B?usp=sharing), так как там уже гарантированно установлены используемые библиотеки. Также все датасеты, использованные в отчете, должны лежать в одной папке с кодом, иначе нужно будет изменить в коде путь к ним. Дополнительно нужно добавить в эту же папку Kaggle.json, третий датасет с изображениями скачивает из внешнего хранилища с помощью этого файла.

Затем нужно просто последовательно выполнить код в ячейках. Никакие параметры менять не нужно, иначе не получиться воспроизвести достигнутые результаты классификации в точности как полученные в отчете к работе.