МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Понижение размерности пространства признаков»

Тема: Машинное обучение

Студент гр. 6304 _____ Виноградов К.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Загрузка данных.

Загрузим данные из csv таблицы, произведем их нормировку к интервалу [0, 1] и построим диаграммы рассеяния для пар соседних признаков по нормированным данным, также добавим легенду для определения типа признака по цвету элемента на диаграмме. Результат представлен на рис.1.

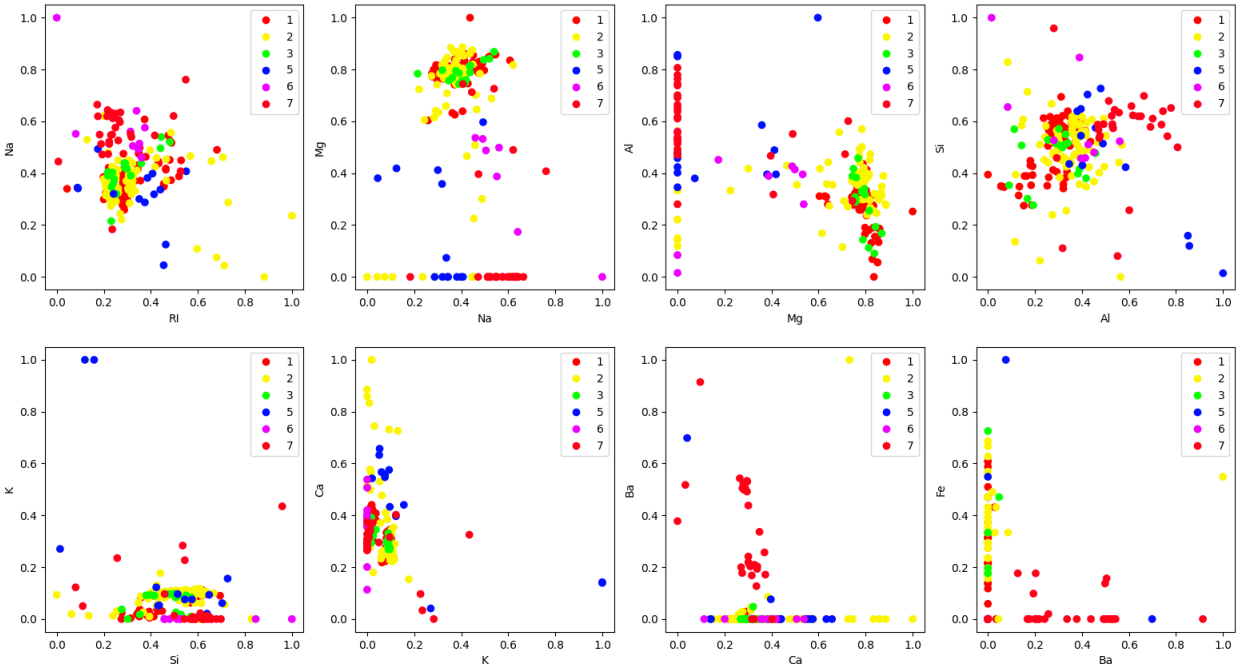

Рисунок 1 – Диаграммы рассеяния исходных данных

Метод главных компонент.

Используя метод главных компонент, произведем понижение размерности пространства признаков до 2. Выведем также информацию о сохраненной дисперсии множества и сингулярные числа, соответствующие оставшимся компонентам. Информация представлена на рис. 2 и 3.

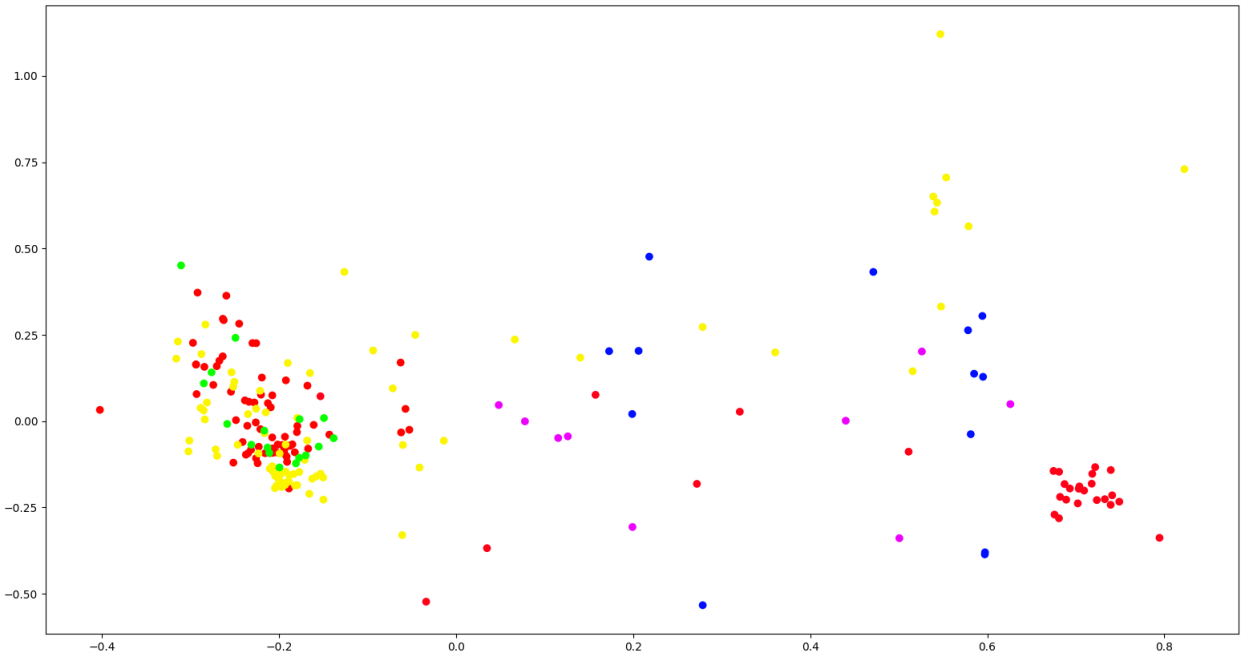

Рисунок 2 – Диаграммы рассеяния данных после понижения размерности пространства

```
Суммарная дисперсия 63.41966621042779 %
Дисперсия соответствующего признака [0.45429569 0.17990097]
Сингулярные числа соответствующего признака [5.1049308  3.21245688]
```

Рисунок 3 – Данные после алгоритма PCA

Исходя из полученных данных можно сделать вывод, что два признака суммарно содержат более 63% информации всего набора, причем больший из них содержит примерно 45% информации, что наглядно представлено на диаграмме – измерения в большей степени распределены по оси абсцисс чем по оси ординат.

Было установлено что для сохранения более 85% информации достаточно оставить 4 измерения. Результат на рис. 4.

```
Суммарная дисперсия 85.8669730510272 %
Дисперсия соответствующего признака [0.45429569 0.17990097 0.12649459 0.09797847]
Сингулярные числа соответствующего признака [5.1049308  3.21245688 2.69374532 2.3707507 ]
```

Рисунок 4 – Признаки 4-мерного пространства

После восстановления данных с помощью inverse_transform() можно заметить что восстановленные данные не идентичны изначальным. Это происходит из-за потери информации при сокращении размерности. Результаты представлены на рис. 5 и 6.

	0	1	2	3
0	0.43284	0.43759	1.00000	0.25234
1	0.28358	0.47519	0.80178	0.33333
2	0.22081	0.42105	0.79065	0.38941
3	0.28578	0.37293	0.82183	0.31153
4	0.27524	0.38195	0.80624	0.29595
5	0.21115	0.30977	0.80401	0.41433
6	0.27568	0.38647	0.80178	0.26480
7	0.28139	0.36391	0.80401	0.23676
8	0.35250	0.49774	0.79733	0.33645
9	0.28095	0.34135	0.80178	0.33333
10	0.20018	0.29925	0.77060	0.39564
11	0.28446	0.31128	0.81514	0.30530
12	0.20808	0.32331	0.76392	0.34579
13	0.27788	0.32030	0.79287	0.30530
14	0.28446	0.28271	0.79955	0.31776
15	0.28358	0.31278	0.78842	0.29283
16	0.29368	0.29323	0.81737	0.27103
17	0.47454	0.54586	0.85746	0.18692

	0	1	2	3
0	0.41356	0.40513	0.99134	0.22821
1	0.26026	0.42006	0.80388	0.33447
2	0.21073	0.41765	0.78426	0.35840
3	0.28310	0.40648	0.82520	0.30764
4	0.25152	0.39928	0.80299	0.30586
5	0.21452	0.30646	0.79864	0.39752
6	0.25964	0.39589	0.80117	0.29462
7	0.26119	0.38732	0.80198	0.27822
8	0.33630	0.42995	0.80583	0.32280
9	0.26374	0.35554	0.79368	0.32641
10	0.20199	0.30882	0.76567	0.39141
11	0.27144	0.39068	0.80503	0.28971
12	0.20864	0.30506	0.76394	0.37689
13	0.25710	0.32359	0.78566	0.32436
14	0.25281	0.38518	0.78379	0.29134
15	0.25696	0.38770	0.77979	0.29059
16	0.28006	0.38231	0.80659	0.27035
17	0.48094	0.41914	0.87400	0.22613

Рисунок 5 – Восстановленные данные после уменьшения размерности до 4 (оригинал слева, восстановленные справа)

	0	1	2	3
0	0.43284	0.43759	1.00000	0.25234
1	0.28358	0.47519	0.80178	0.33333
2	0.22081	0.42105	0.79065	0.38941
3	0.28578	0.37293	0.82183	0.31153
4	0.27524	0.38195	0.80624	0.29595
5	0.21115	0.30977	0.80401	0.41433
6	0.27568	0.38647	0.80178	0.26480
7	0.28139	0.36391	0.80401	0.23676
8	0.35250	0.49774	0.79733	0.33645
9	0.28095	0.34135	0.80178	0.33333
10	0.20018	0.29925	0.77060	0.39564
11	0.28446	0.31128	0.81514	0.30530
12	0.20808	0.32331	0.76392	0.34579
13	0.27788	0.32030	0.79287	0.30530
14	0.28446	0.28271	0.79955	0.31776
15	0.28358	0.31278	0.78842	0.29283
16	0.29368	0.29323	0.81737	0.27103
17	0.47454	0.54586	0.85746	0.18692

	0	1	2	3
0	0.43284	0.43759	1.00000	0.25234
1	0.28358	0.47519	0.80178	0.33333
2	0.22081	0.42105	0.79065	0.38941
3	0.28578	0.37293	0.82183	0.31153
4	0.27524	0.38195	0.80624	0.29595
5	0.21115	0.30977	0.80401	0.41433
6	0.27568	0.38647	0.80178	0.26480
7	0.28139	0.36391	0.80401	0.23676
8	0.35250	0.49774	0.79733	0.33645
9	0.28095	0.34135	0.80178	0.33333
10	0.20018	0.29925	0.77060	0.39564
11	0.28446	0.31128	0.81514	0.30530
12	0.20808	0.32331	0.76392	0.34579
13	0.27788	0.32030	0.79287	0.30530
14	0.28446	0.28271	0.79955	0.31776
15	0.28358	0.31278	0.78842	0.29283
16	0.29368	0.29323	0.81737	0.27103
17	0.47454	0.54586	0.85746	0.18692

Рисунок 6 – Восстановленные данные без изменения размерности (оригинал слева, восстановленные справа)

Алгоритм PCA, реализованный в библиотеке sklearn использует SVD разложение как способ нахождения собственных векторов и собственных чисел через сингулярные вектора и сингулярные числа, при этом доступно 3 вида алгоритмов:

- Full SVD – вычисляет разложение $U\Sigma V^T$ размерности m×n для исходной матрицы размерности m×n
- Truncated SVD – вычисляет разложение U размерности m × k Σ размерности k × k V^T размерности k × n для исходной матрицы размерности m×n
- Randomized SVD – сводится к разложению SVD для приближения $A \approx QQ^TA = Q(S\Sigma V^T) = U\Sigma V^T$ где Q – ортонормированная матрица для случая QR-разложения матрицы AΩ где Ω – Гауссова случайная матрица

Обычно применяется вариант Full SVD, однако Truncated SVD может дать прирост производительности в случае когда m < n или m > n для матрицы размерности m × n. Randomized SVD же дает хороший прирост производительности при большом объеме матрицы, однако результаты могут незначительно отличаться от расчета с помощью Full SVD или Truncated SVD, так как SVD разложение рассчитывается для приближения оригинальной матрицы. Результаты для случайной матрицы размерности 1000 × 500 на рис. 7 и 8.

```
Время работы Full SVD: 2.116121292114258 Размерность матрицы: (1000, 500)
Время работы Truncated SVD: 0.9380536079406738 Размерность матрицы: (1000, 500)
Время работы Randomized SVD: 0.36602067947387695 Размерность матрицы: (1000, 500)
```

Рисунок 7 – Сравнение времени работы алгоритмов

	0	1	2
0	15.56795	15.56795	15.55617
1	15.36075	15.36075	15.35146
2	15.25724	15.25724	15.23452
3	15.13950	15.13950	15.11428
4	15.08571	15.08571	15.07135
5	15.01668	15.01668	14.99816
6	14.96319	14.96319	14.94304
7	14.88722	14.88722	14.86440
8	14.87003	14.87003	14.85311
9	14.81132	14.81132	14.78484
10	14.79048	14.79048	14.76900
11	14.68313	14.68313	14.65557
12	14.65720	14.65720	14.62852
13	14.59976	14.59976	14.55576
14	14.49965	14.49965	14.46337
15	14.48952	14.48952	14.44107
16	14.47138	14.47138	14.42746
17	14.44039	14.44039	14.40194

Рисунок 8 – Сингулярные числа, столбцы слева направо – Full SVD, Truncated SVD, Randomized SVD

Kernel PCA.

Kernel PCA – модификация метода PCA при которой к изначальному пространству признаков применяется ядерная функция, отображающая набор признаков в другое пространство (обычно большей размерности), в котором можно избежать потери определенной части информации при сокращении размерности пространства признаков.

Для расчета ядерной матрицы доступны следующие подходы:

- линейный
- RBF
- полиномиальный
- сигмоидальный
- косинусовый

Произведем для каждого из методов тестирование влияния характеристик функции ядра на итоговый результат, в частности на концентрацию информации в собственных числах. Проиллюстрируем это с

помощью графика кумулятивной функции зависимости количества информации от количества собственных чисел.

При использовании линейного метода Kernel PCA идентичен обычному PCA так как отображение происходит в то же самое пространство. Результаты представлены на рис. 9.

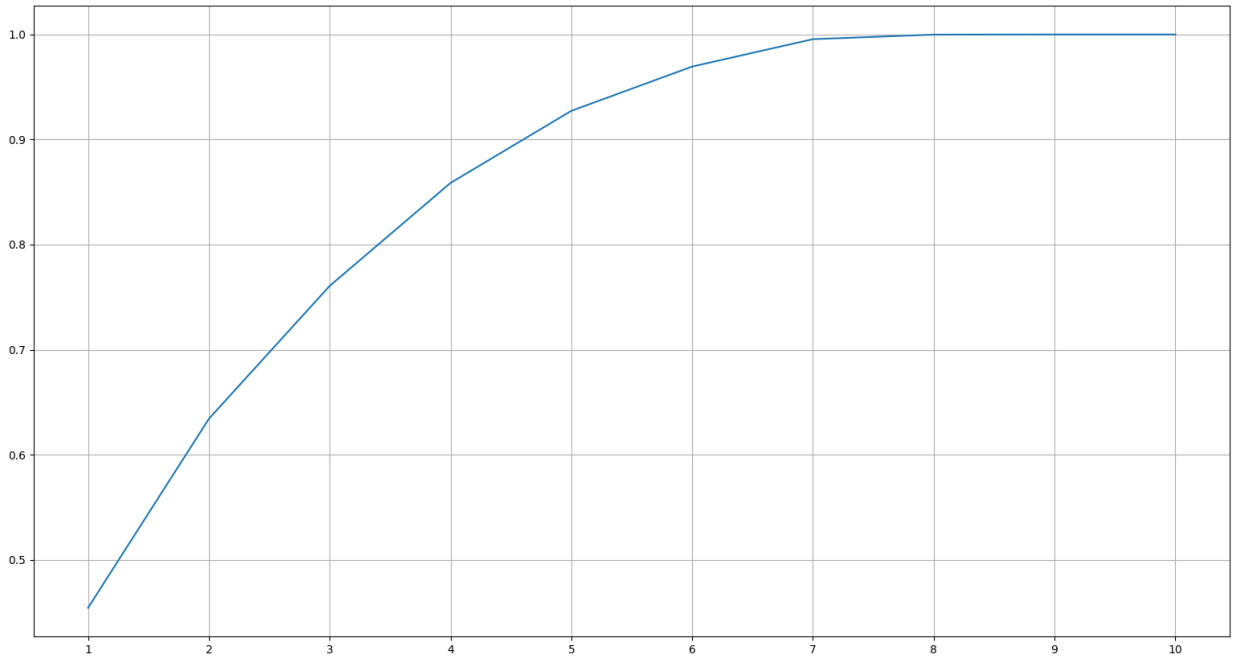

Рисунок 9 – Кумулятивная функция линейного метода

При использовании метода RBF можно корректировать параметр γ. Результаты кумулятивной функции с различными параметрами γ представлены на рис. 10.

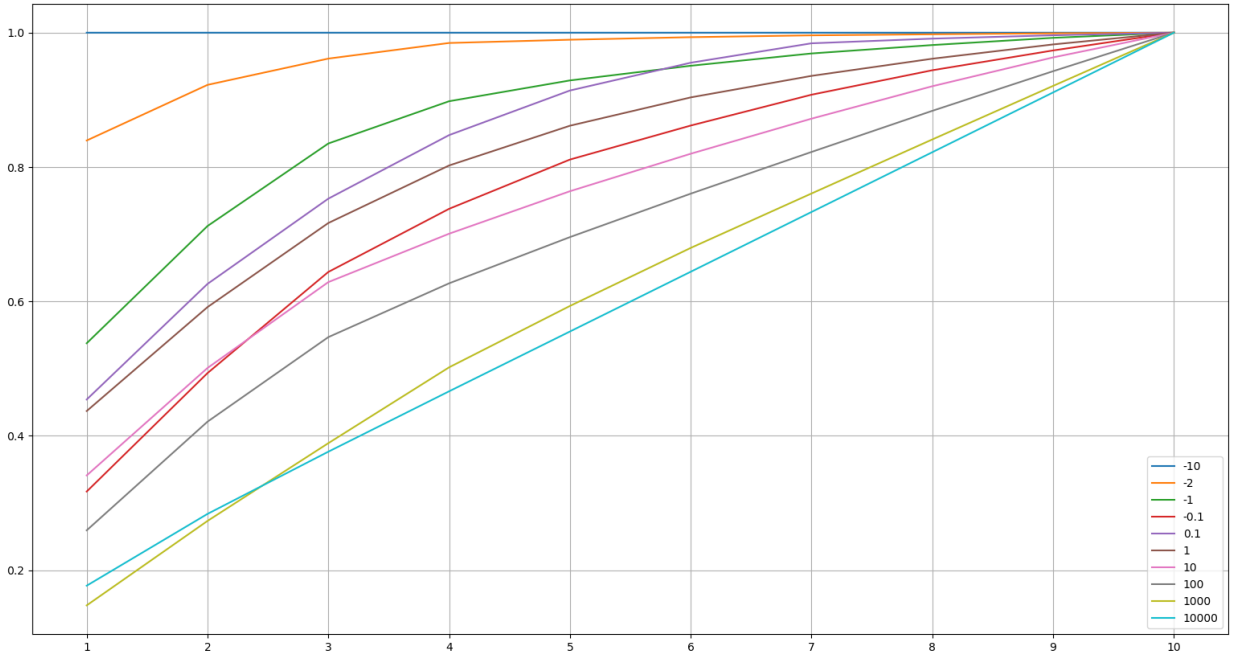

Рисунок 10 – Кумулятивная функция RBF метода при изменении γ

Можно отметить, что при увеличении параметра γ в отрицательную сторону больший вес приобретает первое собственное число, а при увеличении параметра γ в положительную сторону распределение весов между собственными числами выравнивается.

Полиномиальный метод Kernel PCA регулируется сразу тремя параметрами: степенью, свободным членом и также параметром γ. Результаты тестирования метода для различных значений параметров представлены на рис. 11, 12 и 13.

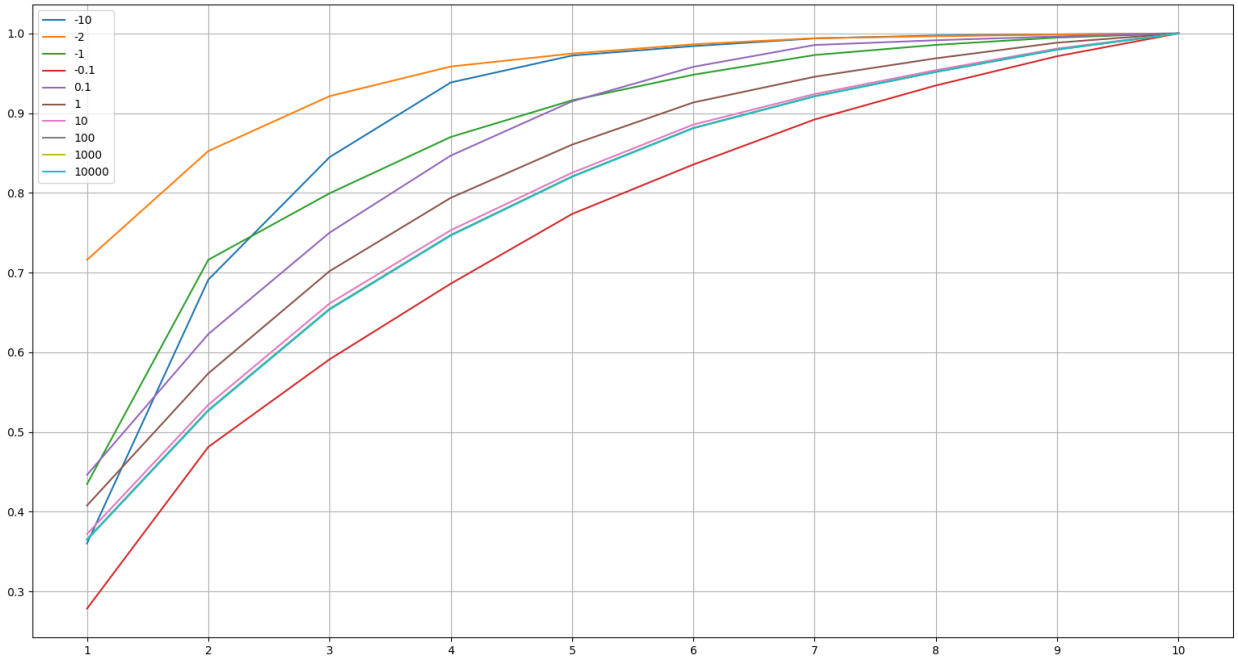

Рисунок 11 – Кумулятивная функция полиномиального метода при изменении γ

В случае полиномиального метода можно отметить, что при приближении параметра γ к нулю с положительной стороны вес первых собственных чисел увеличивается. При использовании отрицательных значений параметра линейную зависимость выявить не удалось.

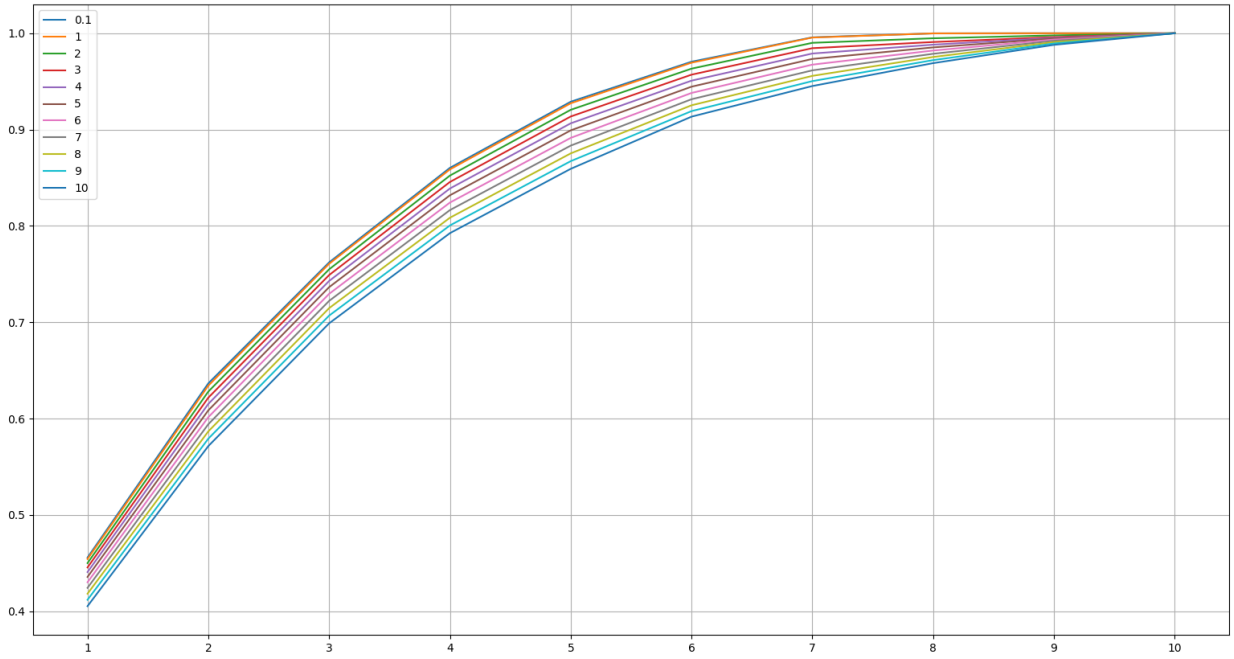

Рисунок 12 – Кумулятивная функция полиномиального метода при изменении степени

Изменение степени в большую сторону более равномерно распределяет веса между собственными числами.

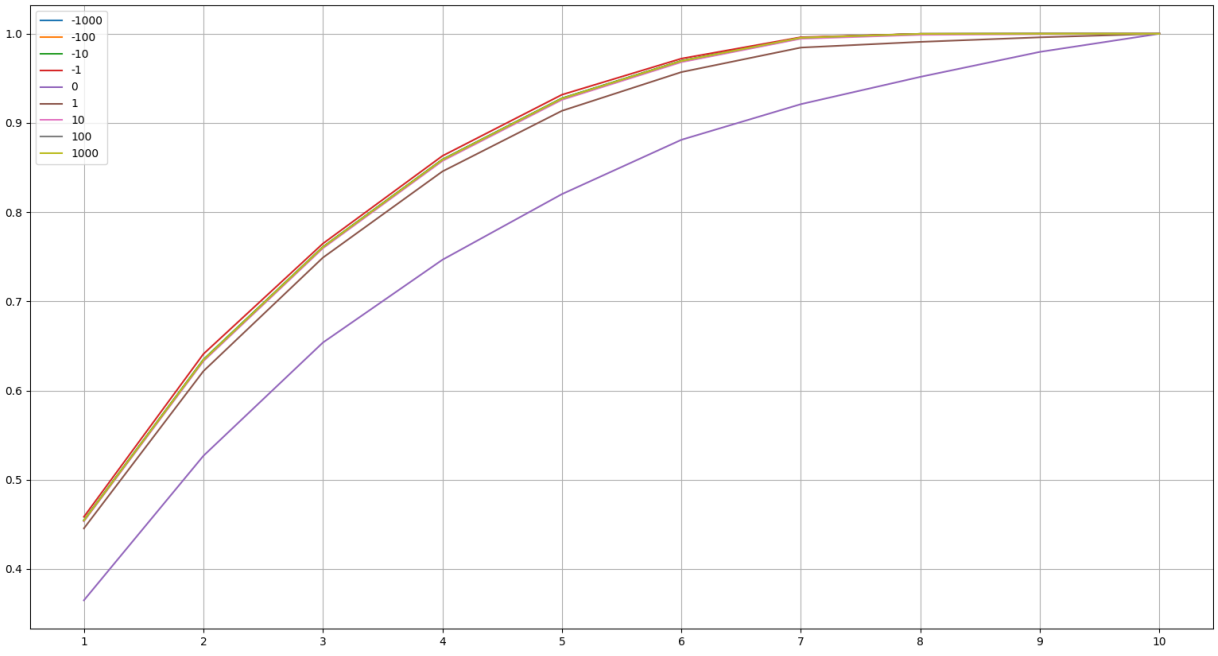

Рисунок 13 – Кумулятивная функция полиномиального метода при изменении свободного члена

Изменение значения свободного члена практически не изменяют результат за исключением принятия его равным 0 – в данном случае веса собственных чисел ощутимо выравниваются.

У сигмоидального метода можно изменять два параметра – параметр γ и свободный член. Тестирование метода с различными значениями параметров представлено на рис. 14 и 15.

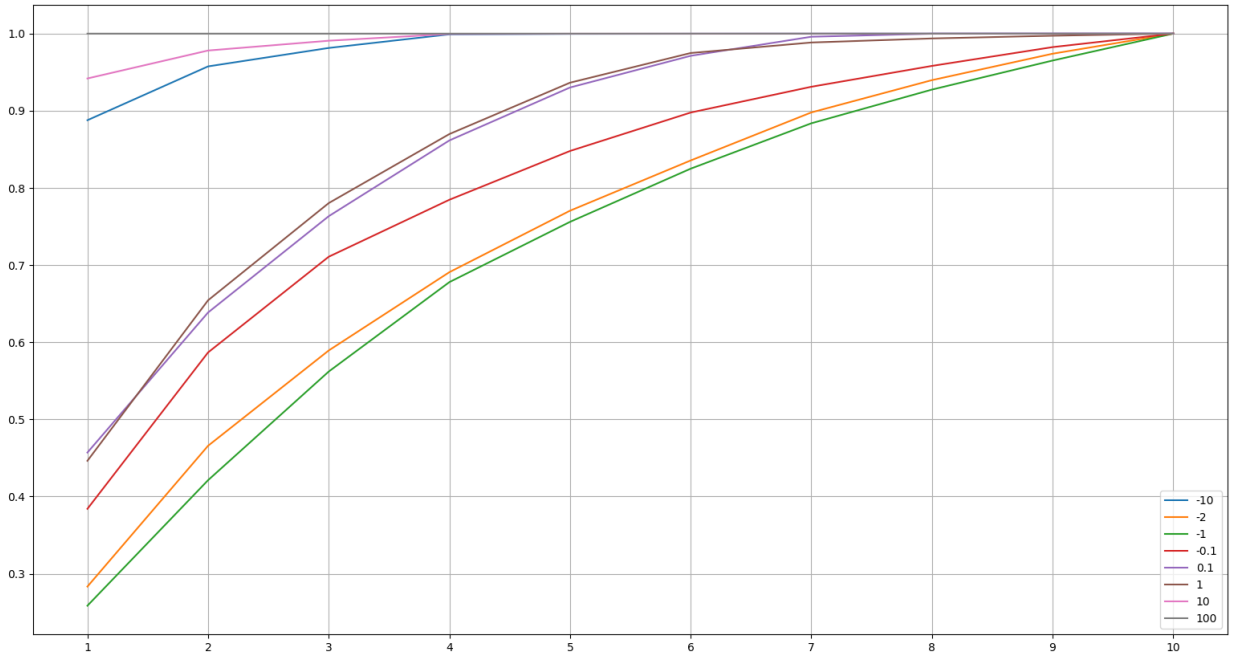

Рисунок 14 – Кумулятивная функция сигмоидального метода при изменении параметра γ

Чем ближе параметр к при увеличении значения параметра в положительную сторону вес первого собственного числа увеличивается. При увеличении в отрицательную сторону сначала веса стремятся к равномерному распределению, а затем после определенного значения вес первого параметра также начинает увеличиваться. Также метод не справляется с расчетом при слишком больших по модулю значениях параметра.

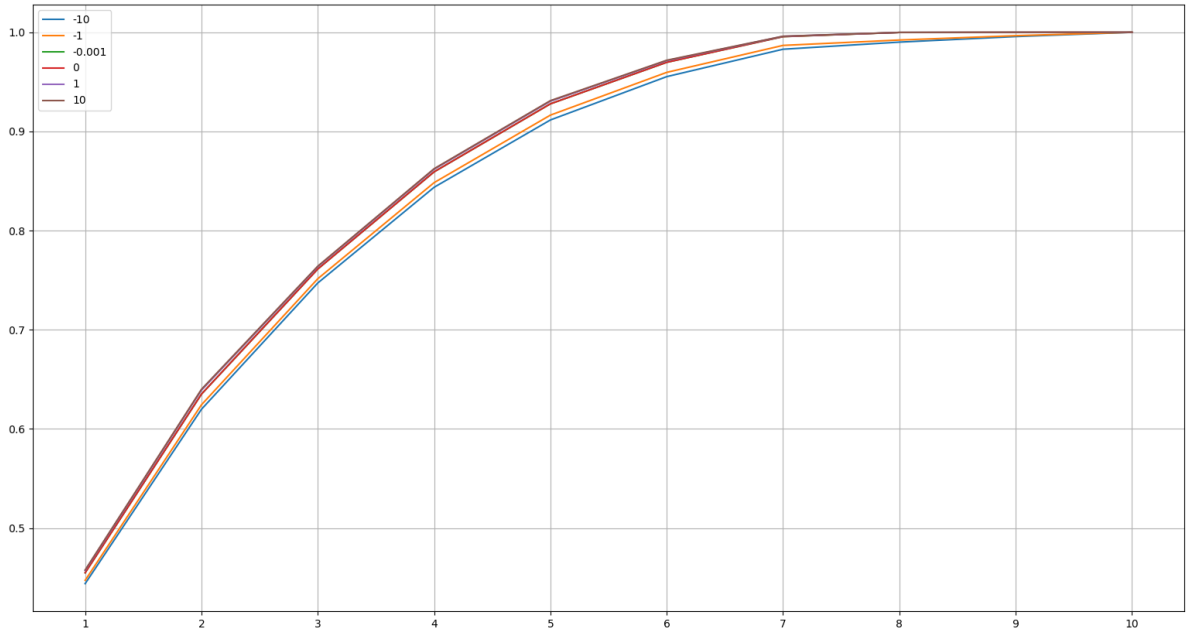

Рисунок 15 – Кумулятивная функция сигмоидального метода при изменении свободного члена

Свободный член слабо влияет на результат, однако более близкие результаты немного добавляют вес первым собственным числам. Метод также не справляется с расчетом при больших по модулю значениях параметра.

У косинусного метода нет изменяемых параметров. Результат применения метода показан на рис. 16.

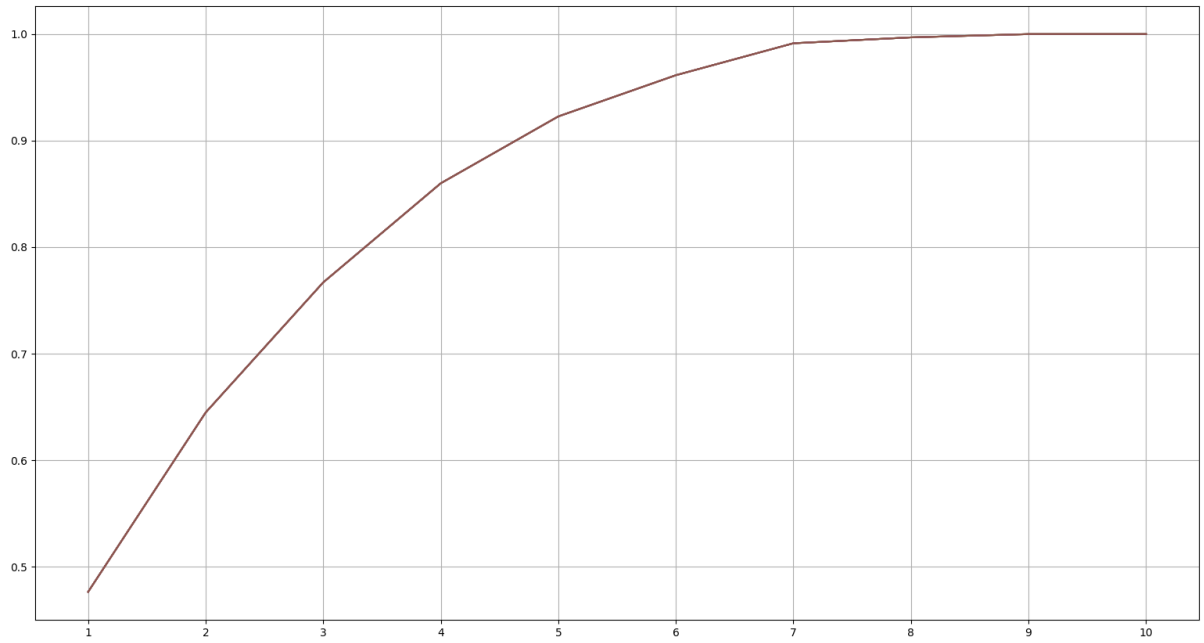

Рисунок 16 – Кумулятивная функция косинусного метода

Сравнение методов показано на рис. 17.

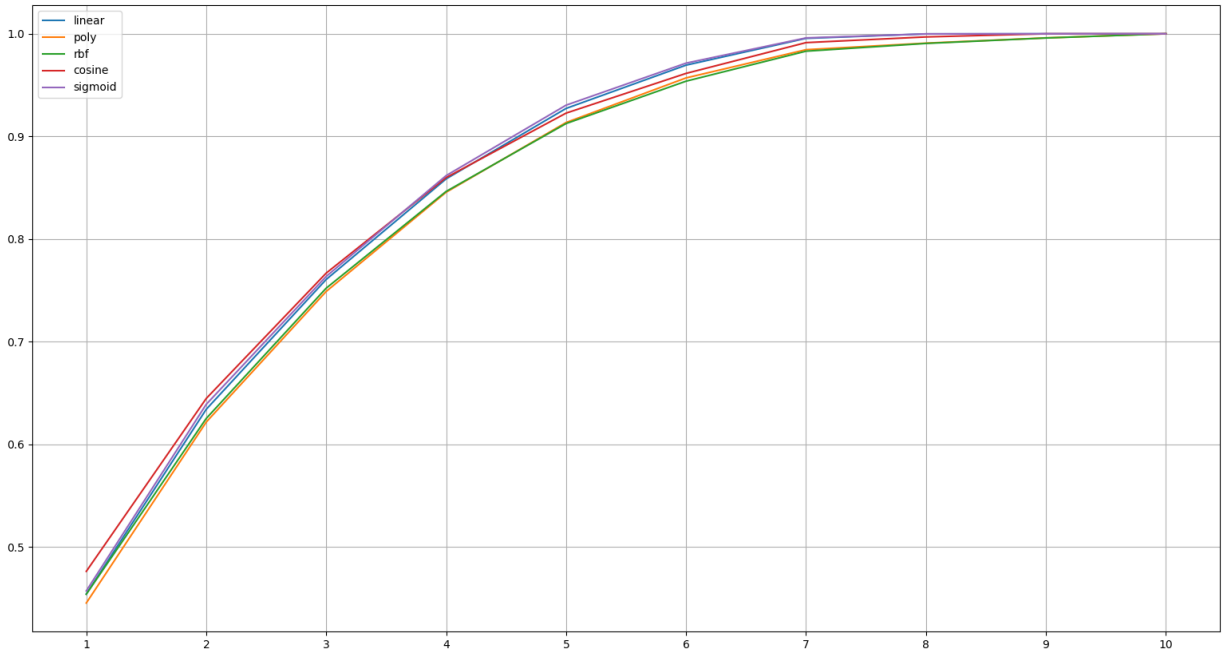

Рисунок 17 – Сравнение методов

На данном наборе данных результат применения методов со стандартными параметрами отличается слабо.

Sparce PCA.

Sparce PCA – модификация метода PCA которая рассчитывает наиболее разреженную матрицу компонент для сохранения согласованности и более удобной интерпретации результатов. Результат уменьшения размерности с помощью этого метода представлен на рис. 18.

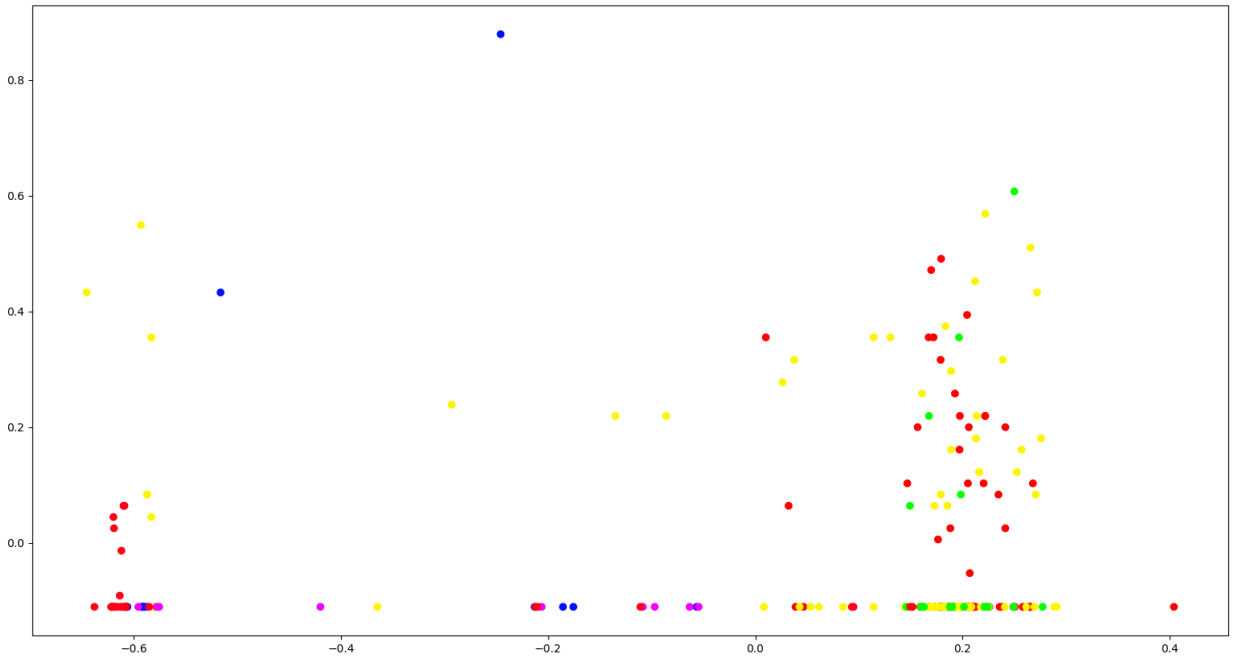

Рисунок 18 – Сокращение размерностей пространства до 2 с помощью Sparce PCA

С помощью параметра α можно менять разреженность компонент. Результаты тестирования зависимости от параметра α представлены на рис. 19.

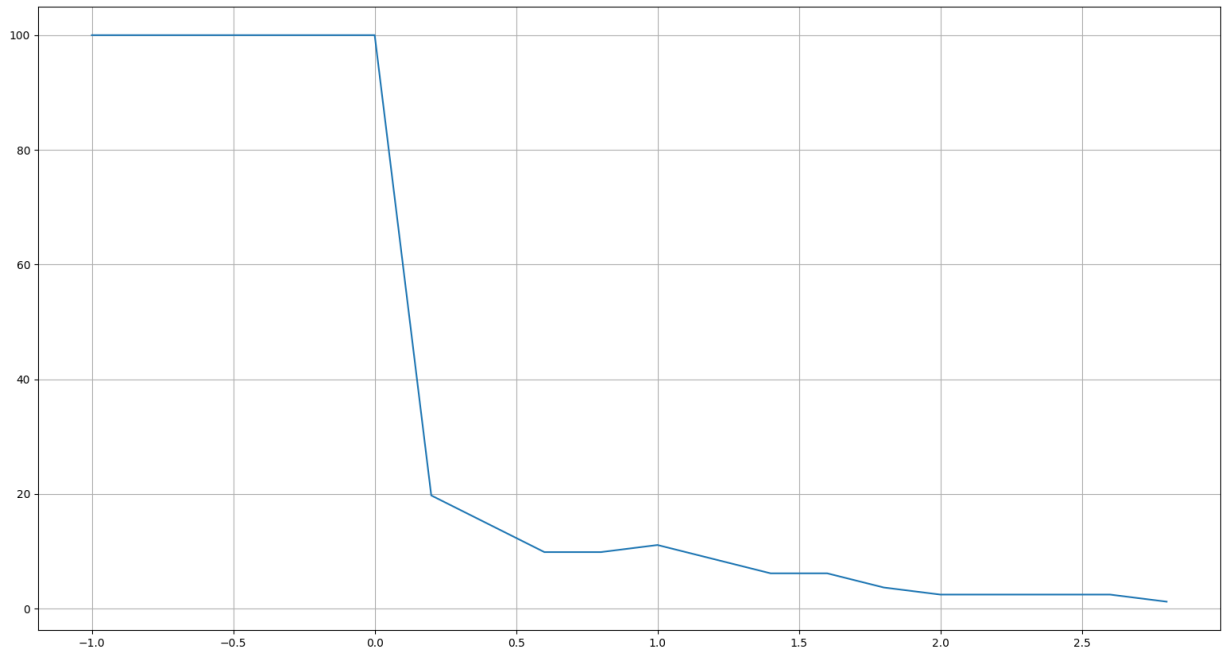

Рисунок 19 – Процент ненулевых элементов матрицы компонент

Можно отметить, что при значении параметра $\alpha \leq 0$ разреживание не производится и Sparce PCA аналогичен обычному PCA.

Factor analysis.

Факторный анализ – процесс выявления взаимосвязей между переменными и поиска скрытых зависимостей. В процессе проведения факторного анализа можно объединять сильно коррелирующие признаки, а следовательно, и сокращать размерность пространства признаков. Результат применения факторного анализа представлен на рис. 20.

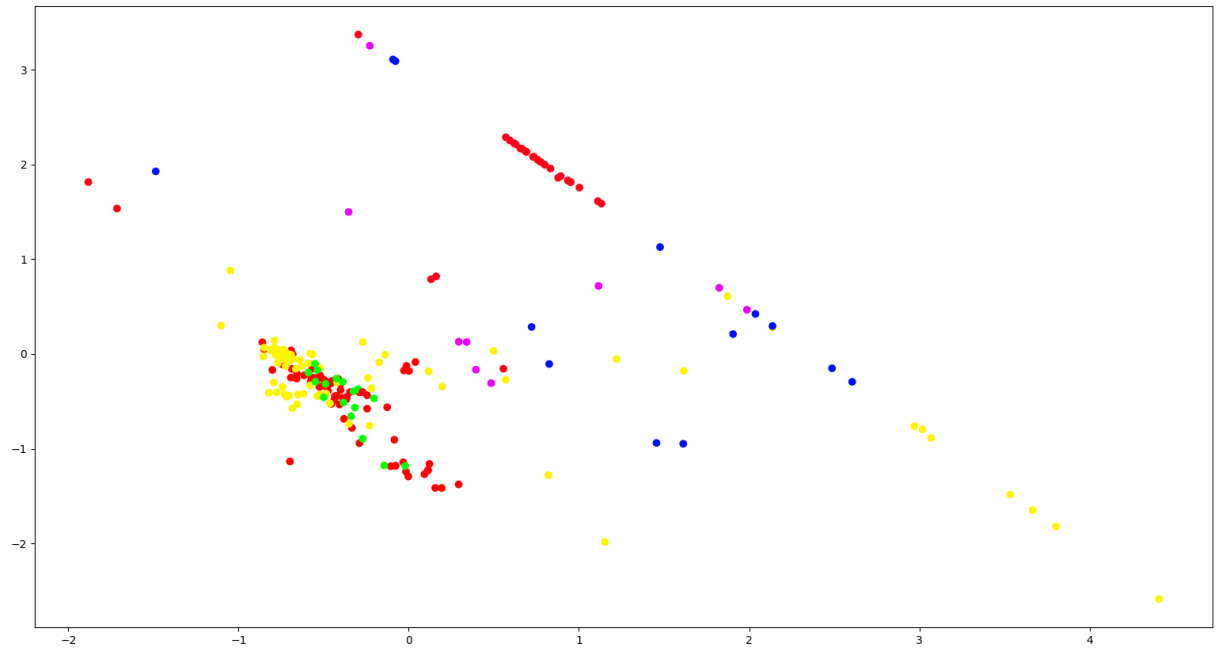

Рисунок 20 – Уменьшение пространства признаков с помощью факторного анализа

Можно отметить, что при значении параметра $\alpha \leq 0$ разреживание не производится и Sparce PCA аналогичен обычному PCA.

Отличия PCA от факторного анализа:

- факторный анализ направлен на поиск скрытых взаимосвязей и зависимостей (признаков), в то время как PCA на поиск линейных комбинаций
- факторный анализ является статистическим и аналитическим методом который в большинстве случаем можно интерпретировать в то время как PCA является математическим инструментом и часто результат его применения интерпретировать невозможно
- факторный анализ рассматривает ковариацию признаков а PCA работает с их дисперсией
- преобразования PCA приводят к нахождению ортогональных компонент, преобразования при факторном анализе это не гарантируют

Выводы

В ходе выполнения данной лабораторной работы были изучены метод PCA и его модификации, в частности Kernel PCA и Sparce PCA. Была проделана работа по изучению влияния параметрических переменных на результат вышеназванных алгоритмов.

Было проведено сравнение метода PCA с подходом факторного анализа и выделены схожести и различия этих методов.