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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Машинное обучение»

Студент гр. 6304 _____ Антонов С.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы:

Ознакомиться с методами понижения размерности из библиотеки Scikit Learn.

Ход работы:

Загрузка данных

1. На данном этапе был скачан и загружен датасет в датафрейм, выделены описательные признаки и признак, отображающий класс, после этого выполнена нормировка исходных данных к интервалу [0, 1].

```
df = pd.read_csv('glass.csv')

var_names = list(df.columns)

labels = df.to_numpy('int')[:,-1]
data = df.to_numpy('float')[:,:-1]

# Data Scaling
data = preprocessing.minmax_scale(data)
```

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
0	1.52101	13.64	4.49	1.10	71.78	0.06	8.75	0.00	0.0	1
1	1.51761	13.89	3.60	1.36	72.73	0.48	7.83	0.00	0.0	1
2	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0.00	0.0	1
3	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0.00	0.0	1
4	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0.00	0.0	1
..	...	...	...	...	...	...	...	...	...	...
209	1.51623	14.14	0.00	2.88	72.61	0.08	9.18	1.06	0.0	7
210	1.51685	14.92	0.00	1.99	73.06	0.00	8.40	1.59	0.0	7
211	1.52065	14.36	0.00	2.02	73.42	0.00	8.44	1.64	0.0	7
212	1.51651	14.38	0.00	1.94	73.61	0.00	8.48	1.57	0.0	7
213	1.51711	14.23	0.00	2.08	73.36	0.00	8.62	1.67	0.0	7

Рисунок 1 Загруженный датасет

2. Были построены диаграммы рассеяния данных.

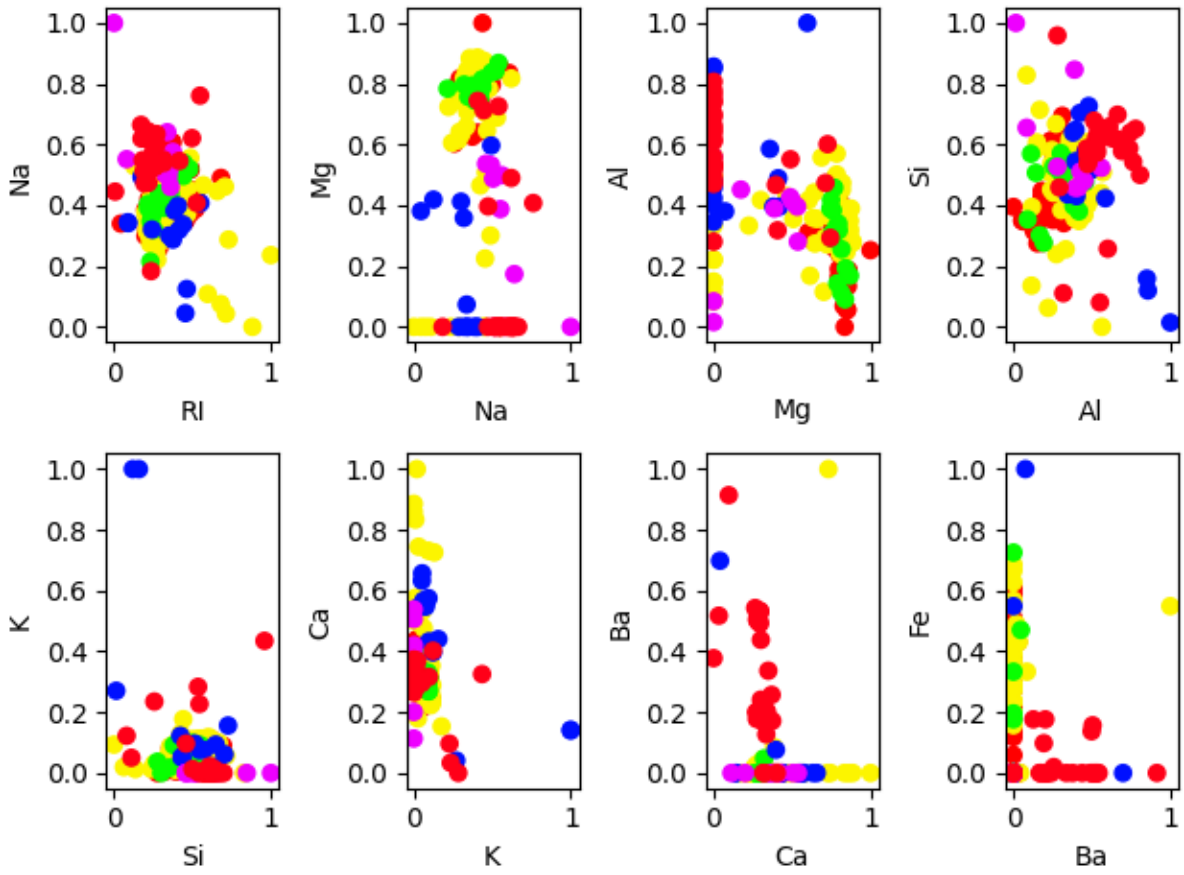

Рисунок 2 Диаграммы рассеяния исходных данных

Метод главных компонент

1. Было выполнено понижение размерности пространства до размерности 2 и построена диаграмма рассеяния (рис. 3).
   - Объясненная дисперсия: (0.4542 0.1799)
   - Собственные числа: (5.105 3.212)

Данные компоненты объясняют всего 63% общей дисперсии, исходя из чего можно сделать вывод, что двух компонент недостаточно, чтобы объяснить достаточно большую долю дисперсии. Кроме того, диаграмма рассеяния показывает, что данными отсутствует явная линейная взаимосвязь.

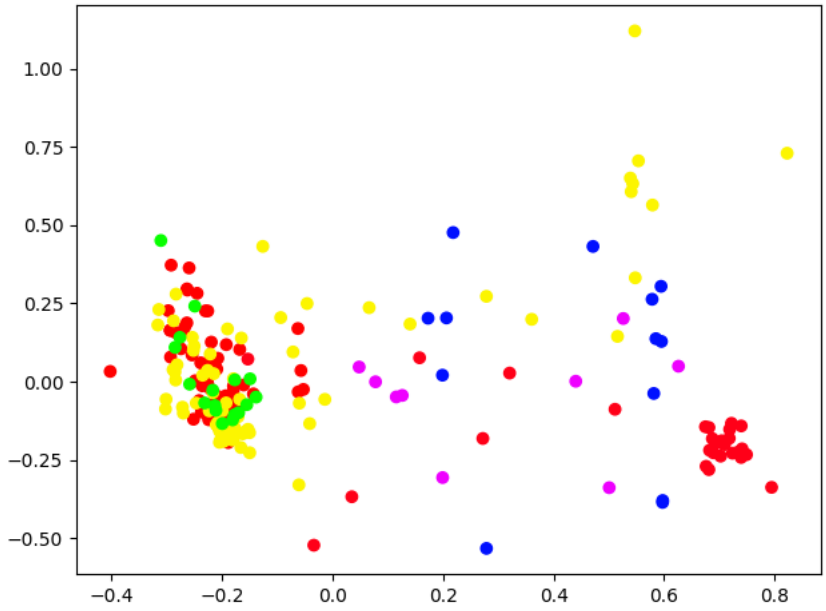

Рисунок 3 Гистограмма стандартизированных признаков

2. Выполнено исследование зависимости бъясненной дисперсии от количества компонент (рис. 4)

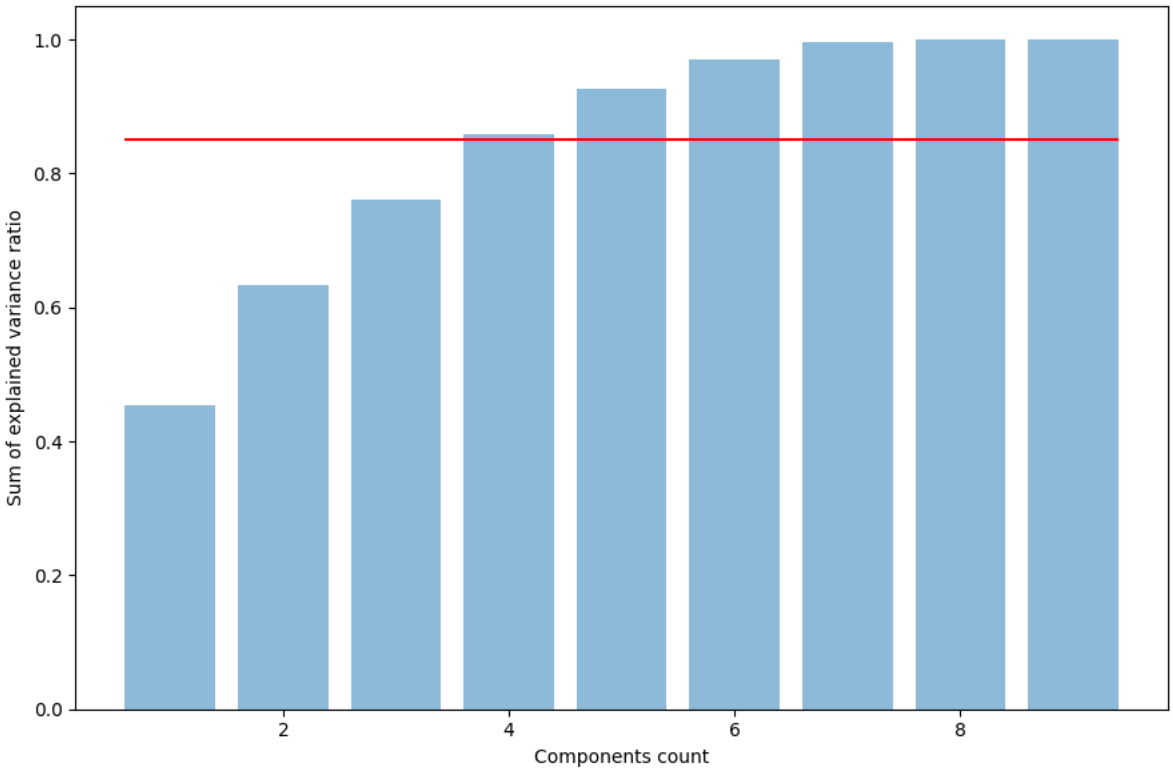

Рисунок 4 Зависимость объясненной дисперсии от количества компонент

Минимальное количество компонент, при котором данные объясняют не менее 85% дисперсии равняется 4.

3. Используя метод inverse_transform исходные данные были восстановлены. Проведено сравнение математического ожидания и дисперсии (табл. 1).

	Среднее значение		Среднеквадратичное отклонение		
	исходное	восстановленное	исходное	восстановленное	Потеря, %
RI	0,317	0,317	0,133	0,130	2,3
Na	0,403	0,403	0,123	0,069	44
Mg	0,598	0,598	0,321	0,321	0,03
Al	0,360	0,360	0,156	0,136	12,8
Si	0,507	0,507	0,138	0,130	6,5
K	0,080	0,080	0,105	0,044	58,1
Ca	0,328	0,328	0,132	0,129	3,03
Ba	0,056	0,056	0,158	0,132	16,5
Fe	0,112	0,112	0,191	0,189	1,0

По приведенным данным видно, что мат. ожидание восстанавливается полностью, однако СКО теряется в диапазоне от 1 до 58%.

4. Выполнено сравнение результатов работы алгоритма анализа главных компонент при различном значении параметра svd_solver.

Параметр svd_solver может принимать следующие значения: full, aprack, randomized.

Результаты работ при различных значениях параметра изображен на рисунке 5:

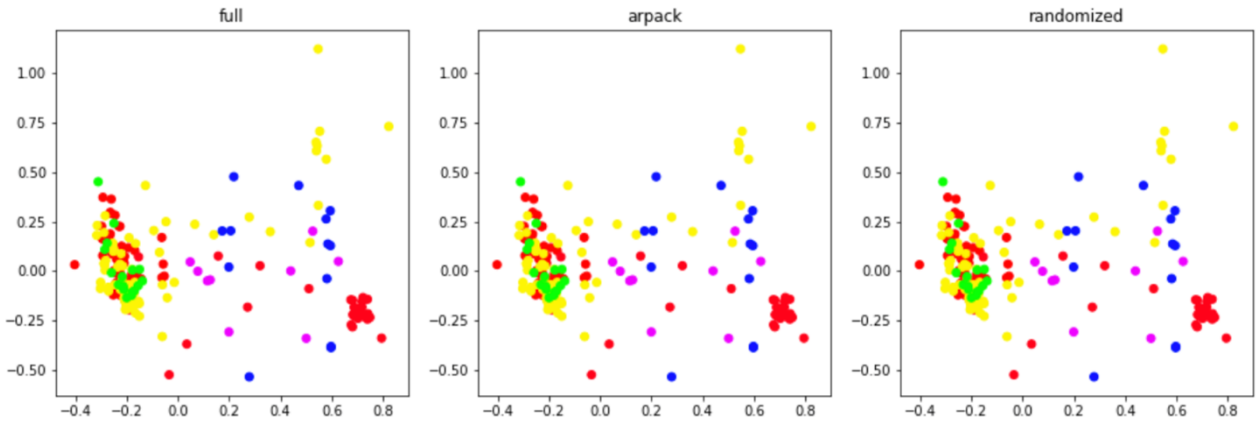

Рисунок 5 Диаграмма рассеяния при различных параметрах svd_solver.

Модификации метода главных компонент.

KernelPCA предлагает модификацию метода главных компонент с использованием различных ядерных функций:

- Линейное ядро: $k(x, y) = x^T y$
- Полиномиальное ядро: $k(x, y) = (\gamma x^T y + c_0)^d$
- Радиально-базисная функция (RBF): $k(x, y) = \exp\left(-\gamma \left|\left|x - y\right|\right|^2\right)$
- Сигмоида: $k(x, y) = tanh(\gamma x^T y + c_0)^d$
- Косинус: $k(x, y) = \frac{x^T y}{||x||\,||y||}$
- Precomplited – ядро заданное пользователем.

KernelPCA ведет себя аналогично PCA при использовании линейного ядра. После настройки данных в поле lambdas_ объекта KernelPCA содержатся собственные числа (26.060, 10.320, 7.2560, 5.620), являющиеся квадратами сингулярный чисел, хранящихся в поле singular_valves: (5.105, 3.212, 2.694, 2.371). Приведем сравнение объясненной дисперсии для различных ядер:

Ядро	Labdas_[0]	Labdas_[1]	Labdas_[2]	Labdas_[3]	Об. Дисп.
linear	26.060	10.320	7.256	5.620	0.859
polynomial	10.918	4.319	3.119	2.368	0.849
RBF	5.351	2.018	1.496	1.111	0.850
Sigmoid	1.006	0.400	0.274	0.216	0.862
cosine	18.314	6.475	4.696	3.578	0.860

SparsePCA

Находит набор разреженных компонент, которые могут оптимально восстановить данные. SparsePCA позволяет использовать один из двух методов: наименьшая угловая регрессия (least angle regression) или координатный спуск (coordinate descent).

Применение различных методов не дает видимых результатов и получаются одинаковые компоненты, однако при варьировании параметра alpha видны различия, также получаются разные компоненты.

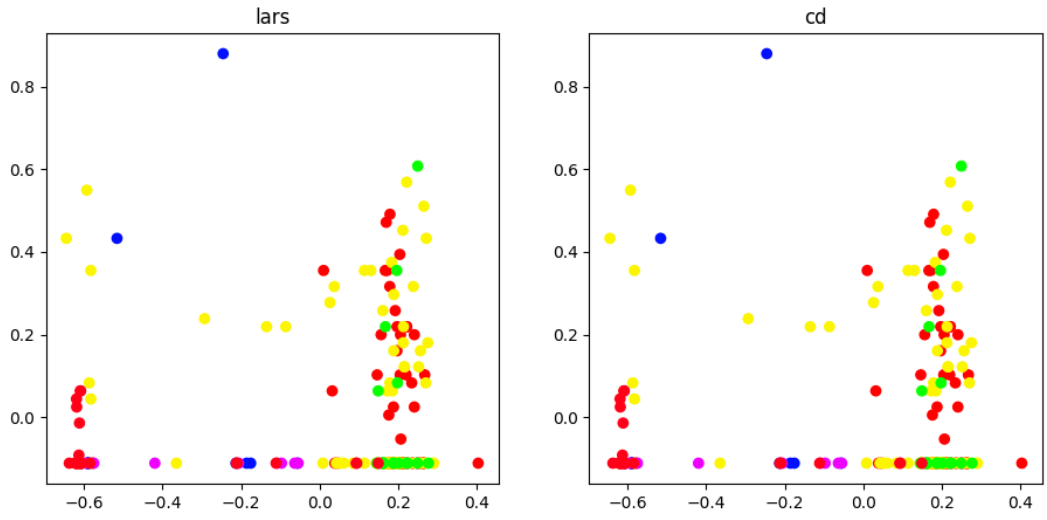

Рисунок 6 диаграмма рассеяния с использованием различных методов

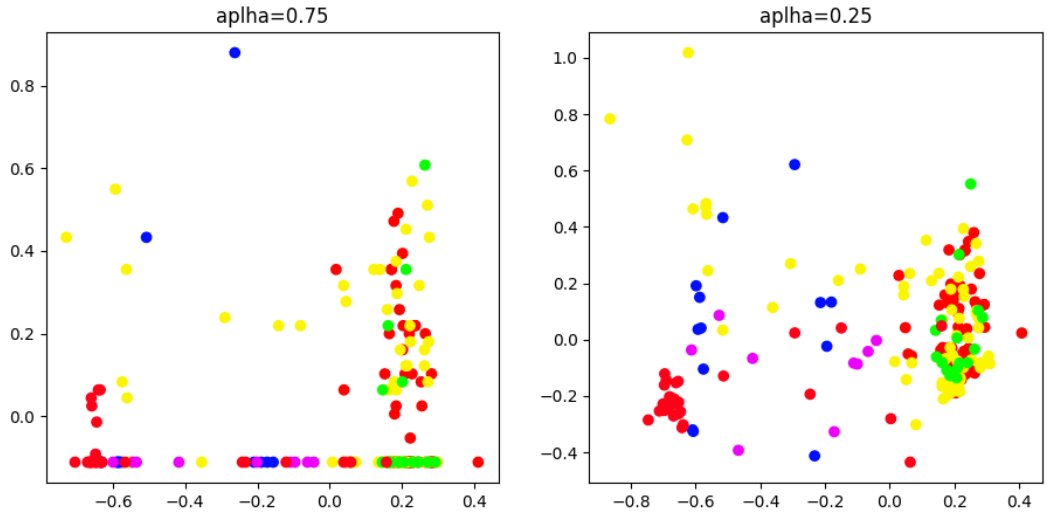

Рисунок 7 диаграмма рассеяния при варьировании параметра alpha

a = 0.75	PC1	0	0	0.985	-0.017	0	0	-0.005	-0.129	0
	PC2	0	0	0	0	0	0	0	0	1
a = 0.25	PC1	-0.008	-0.052	0.943	-0.203	0	0	-0.126	-0.228	0

	PC2	0.477	-0.169	0	-0.288	-0.191	-0.022	0.437	-0.073	.654

Факторный анализ:

1. Выполнено понижение размерности с использованием факторного анализа:

```
transformer = FactorAnalysis(n_components=2)
fa = transformer.fit_transform(data)
```

2. Сравнение результатов PCA и факторного анализа показывает серьзеные различия в полученных данных.

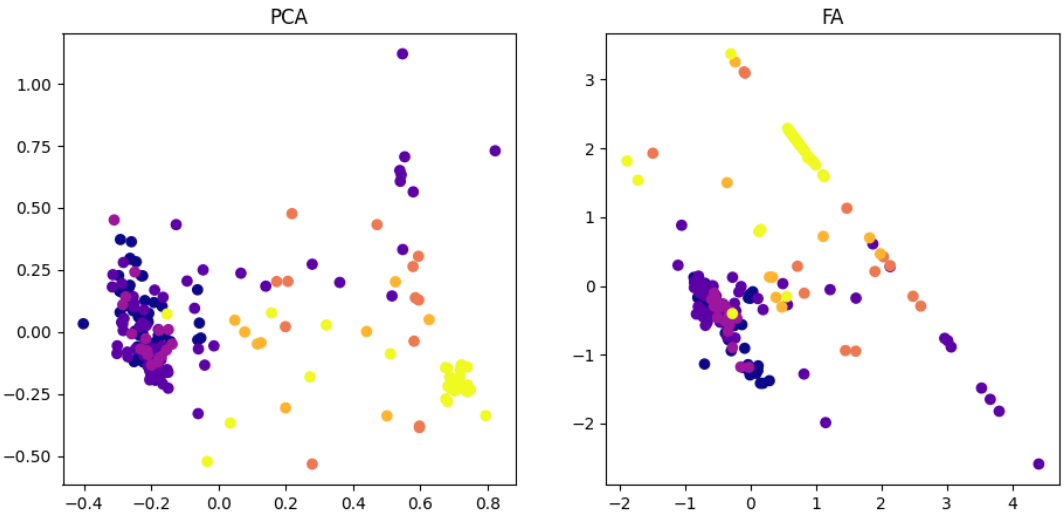

Рисунок 8 Диаграмма рассеяния результатов PCA и FA

3. Различия:
   - Компоненты PCA объясняют максимальную дисперсию, а факторный анализ – ковариацию данных.
   - Компоненты PCA полностью ортогональны друг к другу (независимы), в то время как факторный анализ не накладывает таких ограничений.
   - Компонент PCA представляет собой линейную комбинацию наблюдаемой переменной, в то время как в FA наблюдаемые переменные представляют собой линейные комбинации фактора.

Выводы:

В данной работе были изучены методы понижения размерности: PCA, KernelPCA, SparsePCA и Factor Analysis.

Разное количество компонент в PCA объясняют разное количество дисперсии данных.

KernelPCA используется для поиска нелинейных зависимостей данных. На предложенном наборе данных объясненная дисперсия была 85% независимо от ядра.

Также был изучен факторный анализ и его различия с PCA.

.