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»

Тема: Понижение размерности пространства признаков

Студент гр. 6304 _____ Иванов Д.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы

Ознакомиться с методами понижения размерности данных из библиотеки Scikit Learn

Загрузка данных

1. Загружен датасет, выделены описательные признаки и признак отображающий класс, после этого выполнена нормировка исходных данных к интервалу [0, 1].

```
df = pd.read_csv('datasets/glass.csv')
var_names = list(df)
labels = df.to_numpy('int')[:,-1]
data = df.to_numpy('float')[:,:-1]
preprocessing.minmax_scale(data)
data = preprocessing.minmax_scale(data)
```

2. Построены диаграммы рассеяния данных (рис. 1).

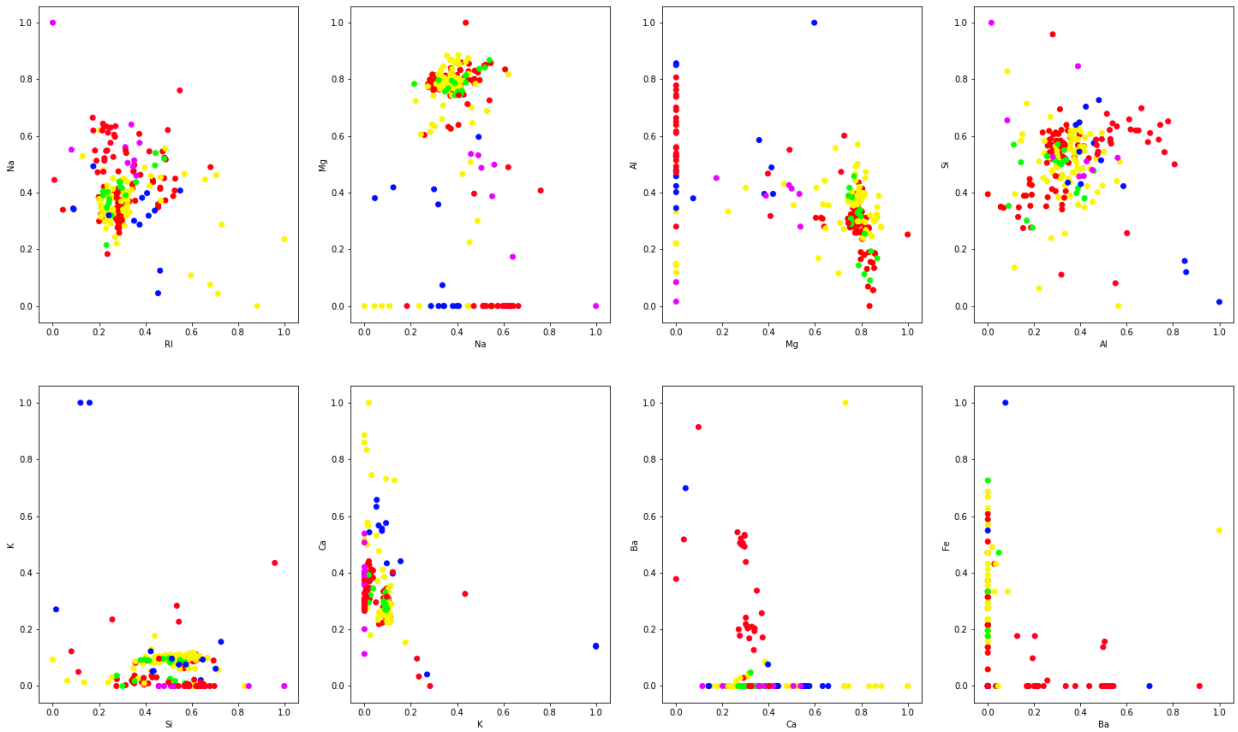

Рис. 1 — Диаграммы рассеяния исходных данных

Метод главных компонент

1. Выполнено понижение размерности пространства до размерности 2, построена диаграмма рассеяния (рис. 3)

- Объясненная дисперсия: (0.454 0.180)
- Собственные числа: (5.105 3.212)

Полученные результаты показывают, что для используемого набора данных две компоненты охватывают лишь ~63% дисперсии, что может быть недостаточным. Кроме того, диаграмма рассеяния показывает, что между данными отсутствует явная линейная взаимосвязь.

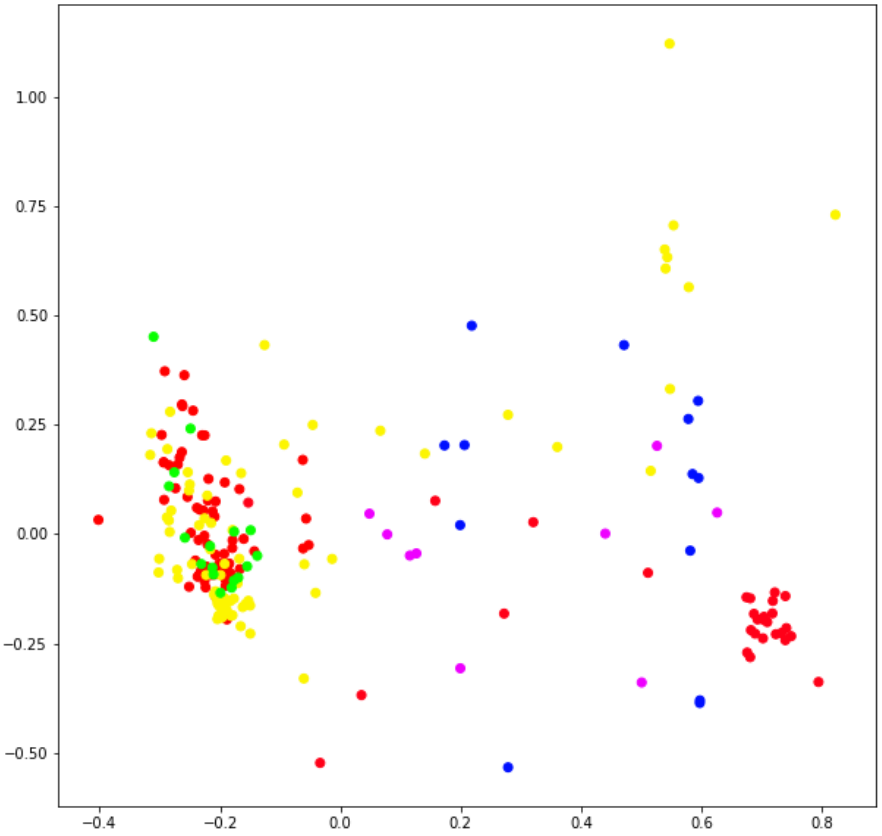

Рис. 2 — Диаграмма рассеяния двух компонент

2. Выполнено исследование зависимости объясненной дисперсии от количества компонент (рис. 5)

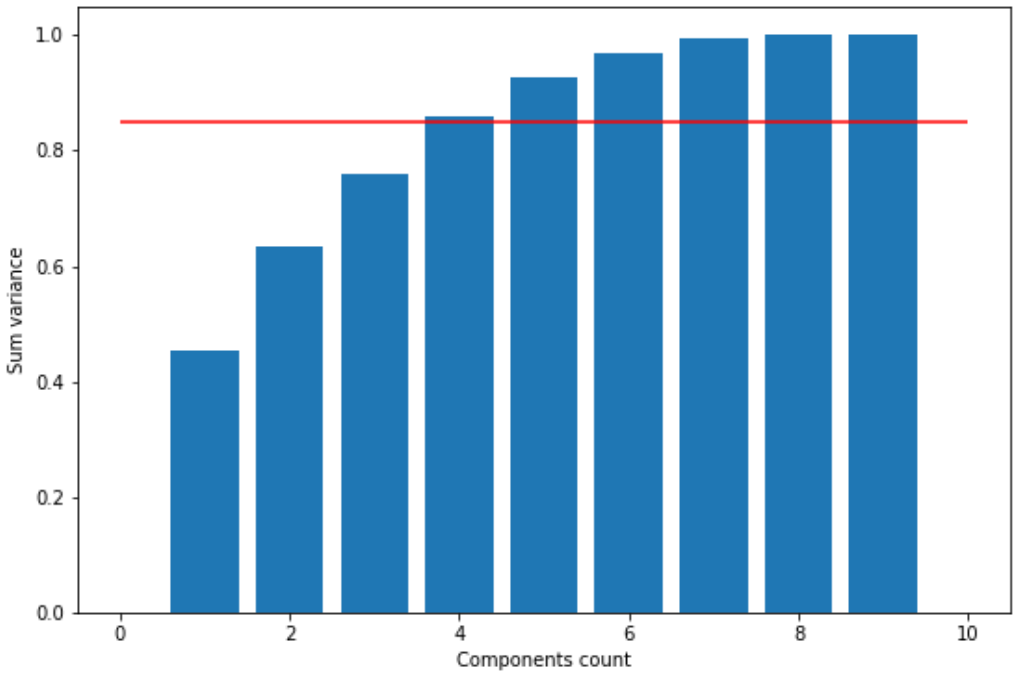

Рис. 3 — Зависимость объясненной дисперсии от количества компонент

Минимальное количество компонент, при котором компоненты объясняют не менее 85% дисперсии, равно 4.

3. Посредстом метода *inverse_transform* восстановлены исходные данные. Проведено сравнение математического ожидания и дисперсии (табл. 1). По полученным данным видно, что математическое ожидания признака восстанавливается полностью, однако некоторые признаки при обратном преобразовании теряют, по сравнению с другими, большую долю СКО: до 58.3%.

Таблица 1 — Сравнение мат. ожидания и дисперсии исходных и восстановленных данных

	Среднее			СКО			
	Исходные	Восстановленные	Отличие	Исходные	Восстановленные	Отличие	% потери
RI	0,317	0,317	0,000	0,133	0,130	0,003	2,3
Na	0,403	0,403	0,000	0,123	0,069	0,054	44
Mg	0,598	0,598	0,000	0,321	0,321	0,001	0,03
Al	0,360	0,360	0,000	0,156	0,136	0,020	12,8
Si	0,507	0,507	0,000	0,138	0,130	0,009	6,5
K	0,080	0,080	0,000	0,105	0,044	0,061	58,1
Ca	0,328	0,328	0,000	0,132	0,129	0,004	3,03
Ba	0,056	0,056	0,000	0,158	0,132	0,026	16,5
Fe	0,112	0,112	0,000	0,191	0,189	0,002	1,0

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

Параметр *svd_solver* может принимать следующие значения: *auto*, *full*, *arpack*, *randomized*.

Результат работы при различных значениях параметра изображен на рисунке 4.

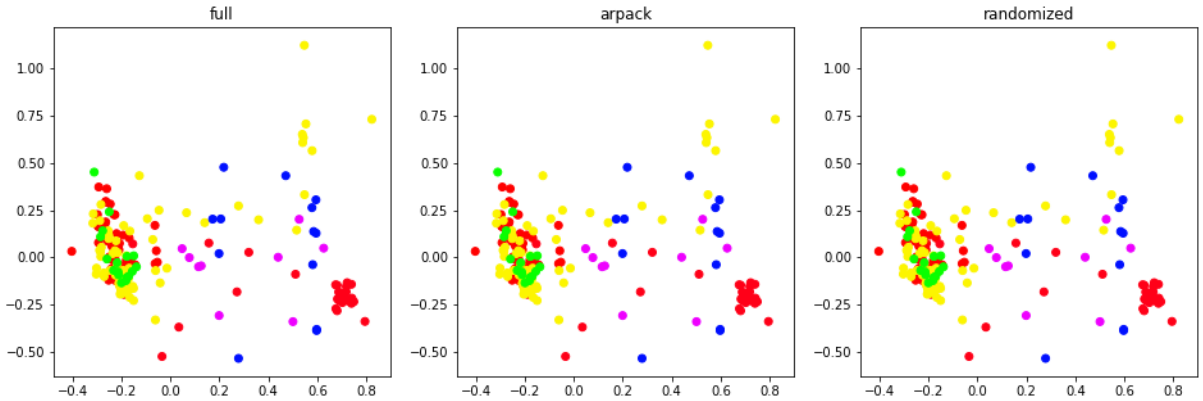

Рис. 4 — Диаграмма рассеяния первых двух главных компонент с различными параметрами *svd_solver*

Модификации метода главных компонент

KernelPCA – вычисление главных компонент с использованием ядер. Возможные ядра: линейное, полиномиальное, радиально-базисная функция (RBF), сигмоида, косинус, предварительно вычисленное.

KernelPCA ведет себя аналогично PCA, при значении параметра *kernel* равного “linear”. После настройки на данных в поле *lambdas_* объекта *KernelPCA* содержатся собственные числа $(26.060, 10.320, 7.2560, 5.620)$, являющиеся квадратами сингулярных чисел, хранящихся в поле *singular_values*: $(5.105, 3.212, 2.694, 2.371)$. Сравнение объясненных дисперсий для различных ядер приведено в таблице 2.

Таблица 2 — Сравнение ядер KernelPCA

Ядро	PC1	PC2	PC3	PC4	Сумма	Объясненная дисперсия
linear	26,060	10,320	7,256	5,620	49,257	0.859
poly	10,918	4,319	3,119	2,368	20,724	0.849
rbf	5,351	2,018	1,496	1,111	9,976	0.850
sigmoid	1,006	0,400	0,274	0,216	1,896	0.862
cosine	18,314	6,475	4,696	3,578	33,064	0.860

SparsePCA – анализ разреженных компонент, которые могут оптимально восстановить данные. Могут использоваться два метода: наименьшая угловая регрессия (*least angle regression*) или координатный спуск (*coordinate descent*) для решения задачи Lasso (метод регрессионного анализа, который выполняет как выбор переменных, так и регуляризацию, чтобы повысить точность прогнозирования и интерпретируемость создаваемой статистической модели)

Применение разных методов не дает визуально различимых отличий (рис. 5), кроме того, получаются одинаковые компоненты. Однако варьирование параметра *alpha*, который влияет на разреженность данных

дает видимые различия (рис. 6). Также получаются различные компоненты (таб. 3).

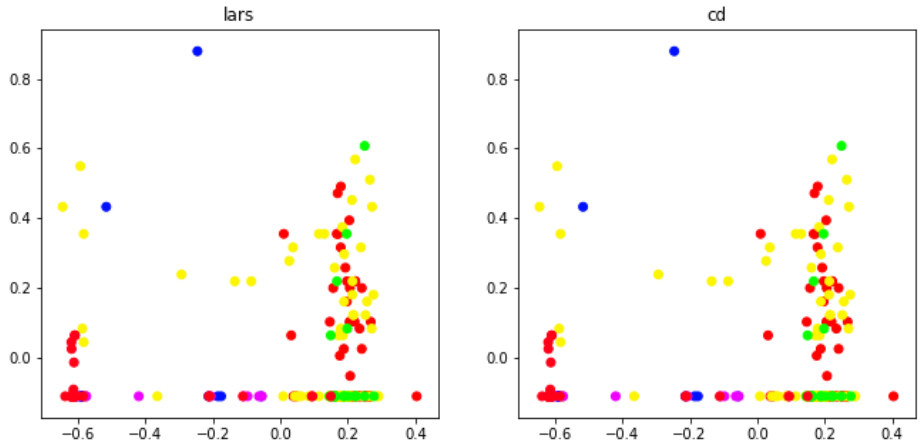

Рис. 5 — Диаграмма рассеяния первых двух главных компонент с использованием различных методов

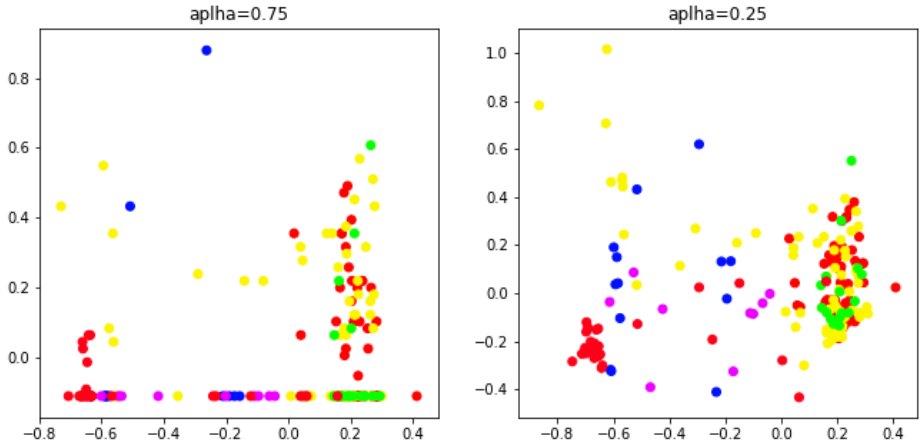

Рис. 6 — Диаграмма рассеяния первых двух главных компонент с варьировании параметра *alpha*

Таблица 3 — Главные компоненты SparsePCA при различных α

$\alpha = 0.75$	PC1	0	0	0,985	-0,017	0	0	-0,005	-0,129	0
	PC2	0	0	0	0	0	0	0	0	1
$\alpha = 0.25$	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	0,654

Факторный анализ

1. Выполнено понижение размерности с использованием факторного анализа.

```
transformer = FactorAnalysis(n_components=2)
fa_2 = transformer.fit_transform(data)
```

2. Сравнение результатами PCA и FA показывает серьезные различия в в полученных данных. (рис. 7)

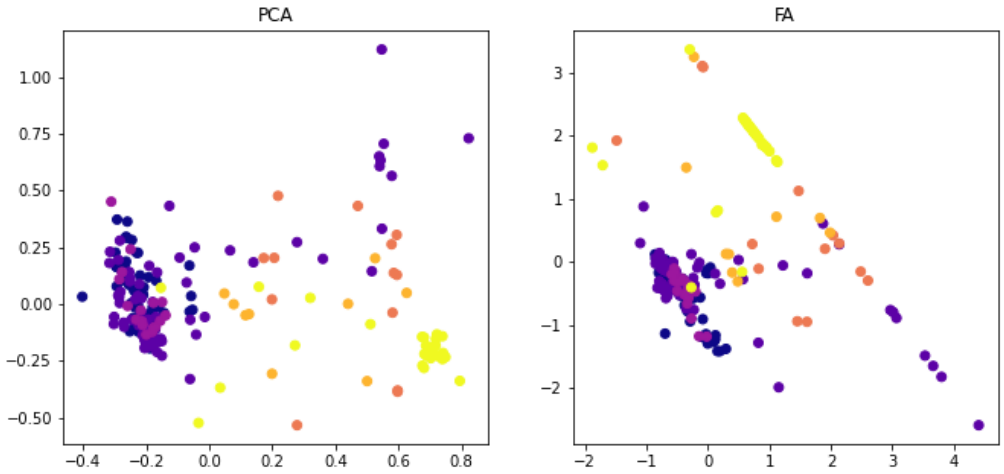

Рис. 7 — Диаграмма рассеяния результатов PCA и FA

3. Различия PCA и FA:
    a. Компоненты PCA объясняют максимальную дисперсию, а факторный анализ – ковариацию данных.
    b. Компоненты PCA полностью ортогональны друг другу (независимы), в то время как факторный анализ не требует не накладывает таких жестких ограничений.
    c. Компонент PCA является линейной комбинацией наблюдаемой переменной, а при факторном анализе наблюдаемые переменные являются линейной комбинацией фактора.
    d. PCA – метод уменьшения размерности, факторный анализ – метод поиска скрытых переменных(факторов).

Вывод

Были изучены методы понижения размерности данных Factor Analysis и Principal component analysis из библиотеки Scikit Learn.

В ходе работы было выяснено, что разное количество компонент в PCA объясняет разное количество дисперсии данных.

KernelPCA используется для поиска нелинейных зависимостей в данных. Однако на предложенных данных объясненная дисперсия была около 85% независимо от ядра. SparsePCA выгодно использовать на данных большого размера.

Также был изучен факторный анализ и его различия с PCA.