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. 6307 _____ Михайлов И. Т.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы:

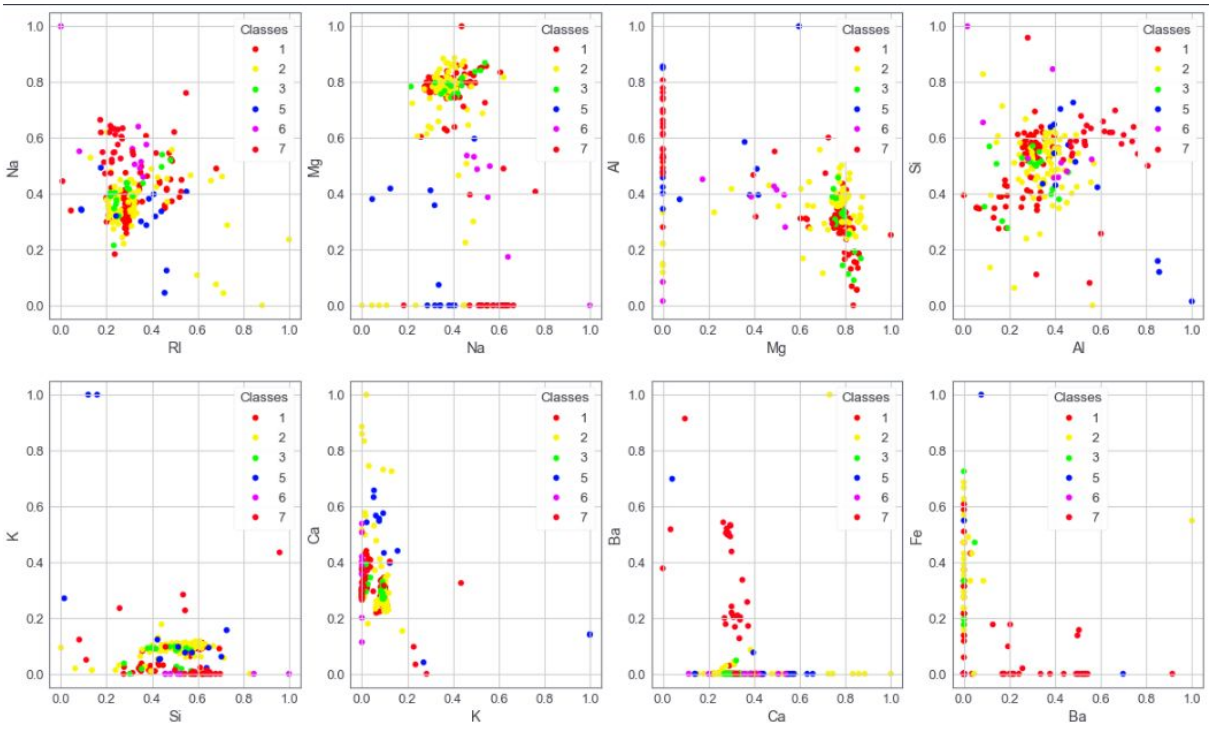
Ознакомиться с методами понижения размерности данных из библиотеки Scikit Learn.

Ход выполнения работы:

Загрузка данных

Датасет загружен в датафрейм, данные разделены на описательные признаки и признак, отображающий класс. Данные нормированы к интервалу [0, 1].

Диаграммы рассеяния для пар признаков:



Метод главных компонент

Проведено понижение размерности пространства до размерности 2.

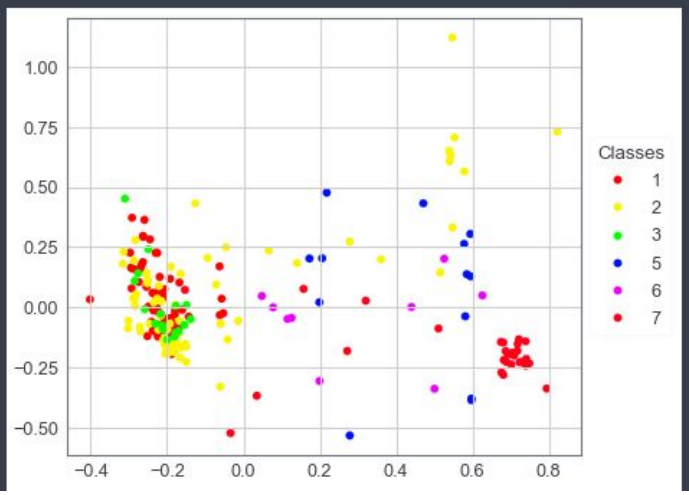
Полученное значение объясненной дисперсии в процентах и полученные собственные числа, соответствующие компонентам:

```
print(pca.explained_variance_ratio_)
print(pca.explained_variance_ratio_.cumsum())
print(pca.singular_values_)
```

```
[0.45429569 0.17990097]
[0.45429569 0.63419666]
[5.1049308  3.21245688]
```

Диаграмма рассеяния после метода главных компонент:

```
plt.scatter(pca_data[:,0],pca_data[:,1],c=labels,cmap='hsv')
plt.legend(*scatter.legend_elements(),loc="center left", title="Classes", bbox_to_anchor=(1, 0.5))
plt.show()
```



По диаграмме рассеяния можно сказать, что между полученными компонентами нет сильной связи (по крайней мере нет линейной зависимости).

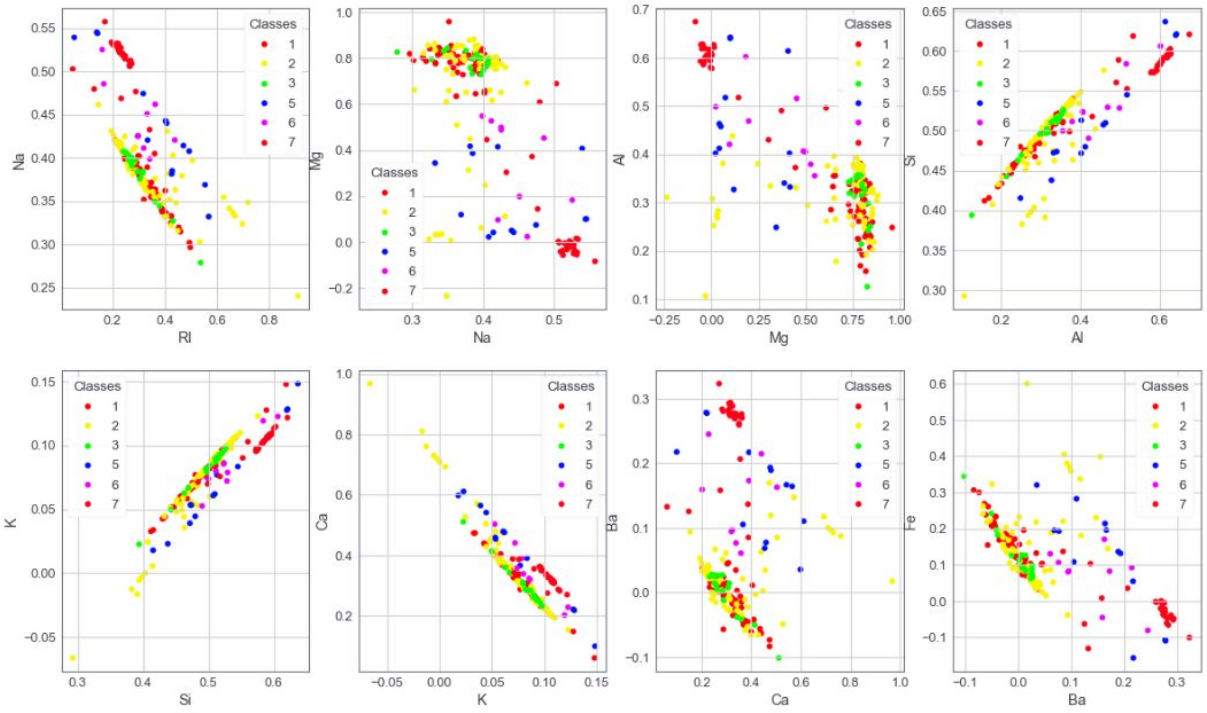
При понижении размерности пространства до 2 сохраняется 63,4% дисперсии. Для сохранения приемлемой точности нужно использовать сохранить хотя бы 4 измерения (85,9%), а для хорошей точности 5 измерений (92,7%):

```
pca = PCA(n_components = 5)
pca_data = pca.fit(data).transform(data)

print(pca.explained_variance_ratio_)
print(pca.explained_variance_ratio_.cumsum())
print(pca.singular_values_)
```

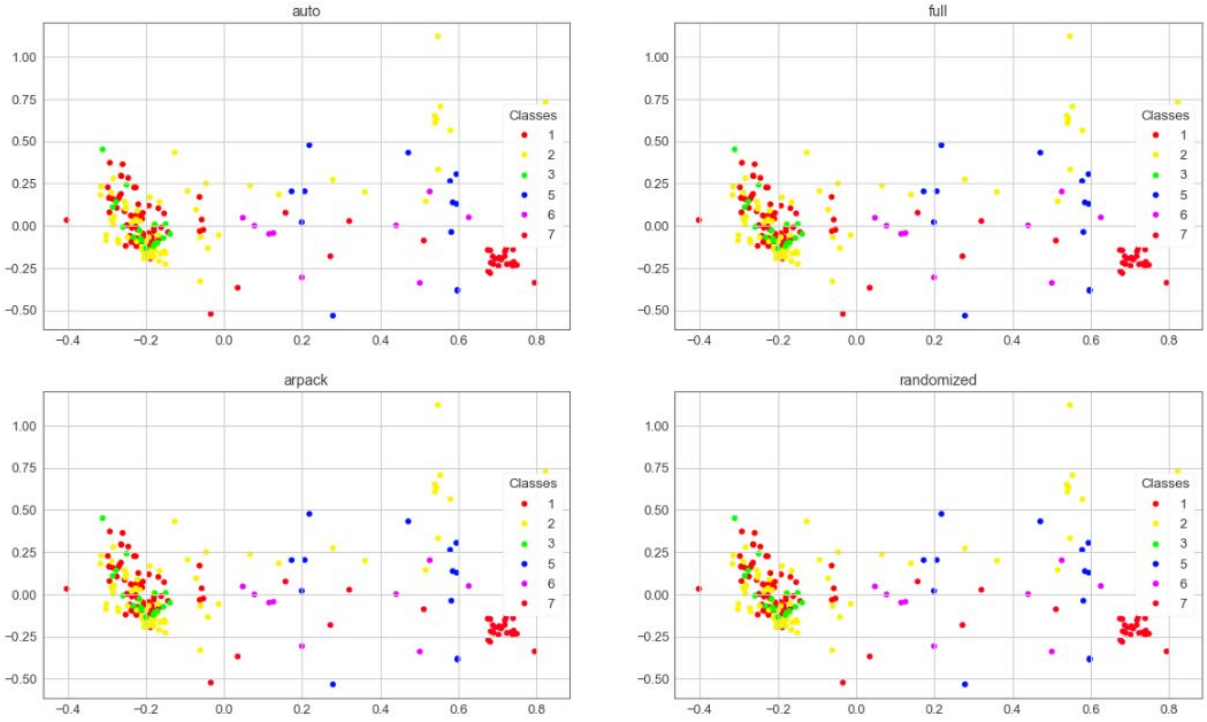
```
[0.45429569 0.17990097 0.12649459 0.09797847 0.06862398]
[0.45429569 0.63419666 0.76069126 0.85866973 0.92729371]
```

Данные после восстановления методом inverse_transform:



На диаграмме рассеяния видно, что разброс случайной величины сильно изменился (потеря данных 36,6%). Также было наглядно продемонстрировано, что метод понижения размерности PCA ищет линейные зависимости между признаками.

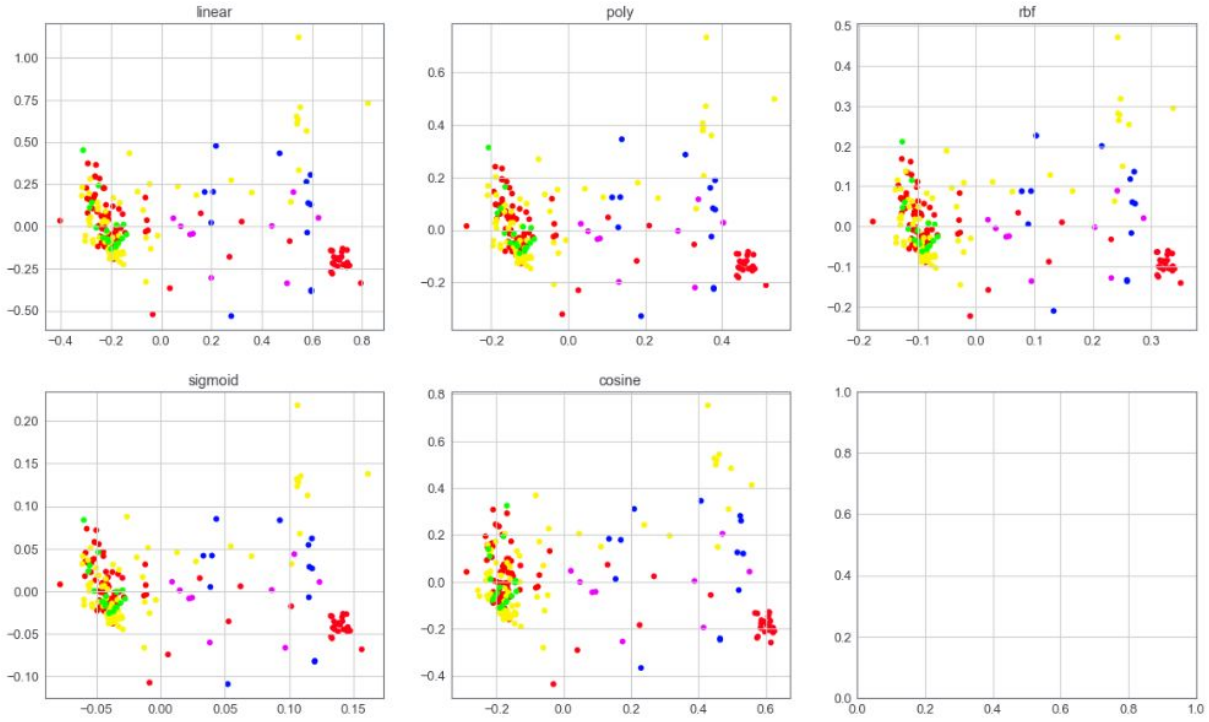
Пронаблюдаем результаты метода главных компонент в зависимости от различных значений параметра svd_solver:



Диаграммы рассеяния не имеют видимых отличий.

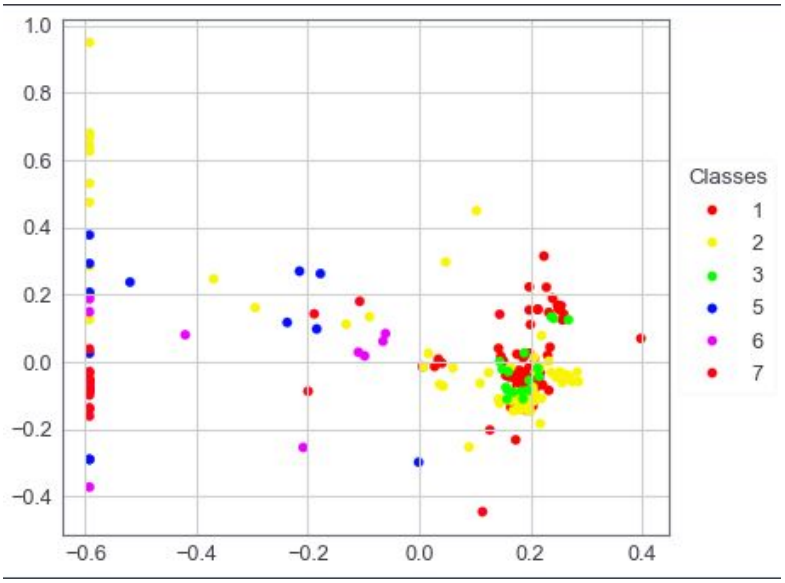
Модификации метода главных компонент

Исследуем **KernelPCA** для различных параметров kernel:



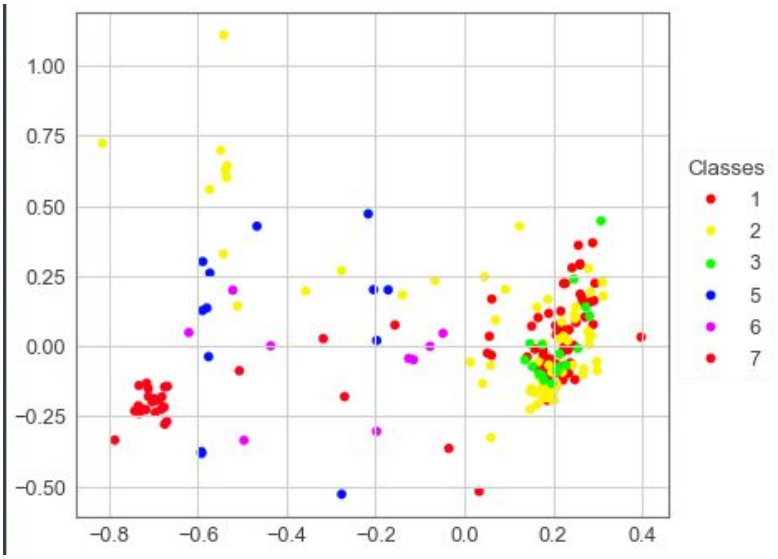
При значении параметра равном ‘linear’ результат полностью совпадает с PCA. Результаты при других значениях параметров отличаются незначительно.

Метод **SparcePCA** работает аналогично PCA, но дает разряженные компоненты. Однако при вызове данной функции с параметрами по умолчанию диаграмма признаков не совпадает с диаграммой PCA:



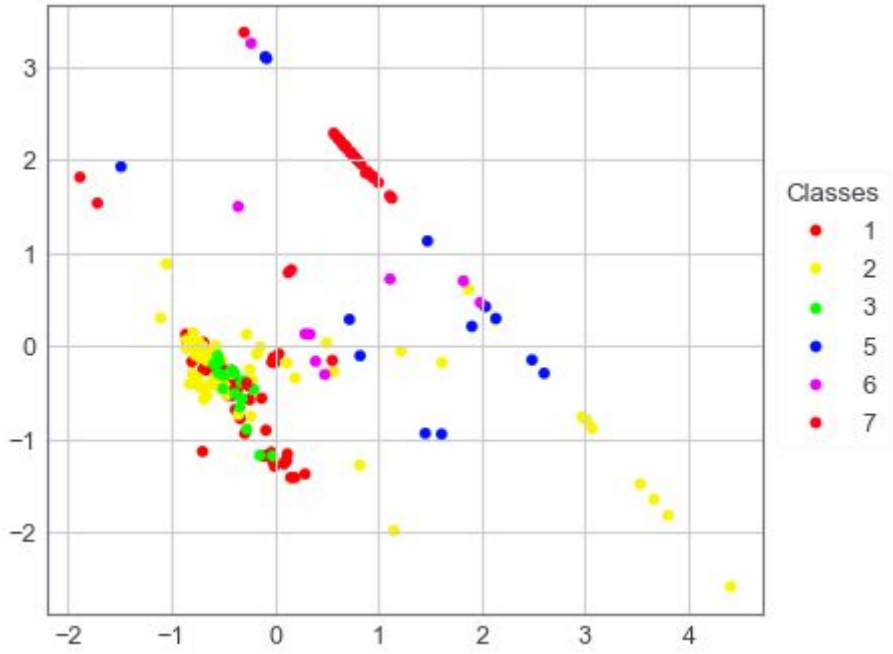
Это связано с тем, что по умолчанию параметр alpha, отвечающий за разреженность равен 1.

Если вызвать SparcePCA с параметром alpha = 0, мы получим результат аналогичный методу PCA:



Факторный анализ

Факторный анализ служит для нахождения скрытых зависимостей между компонентами. Для двух компонент результатом факторного анализа будет:



Метод главных компонент применяется главным образом для сокращения объема данных, а факторный анализ для нахождения зависимостей между признаками.

Вывод

Были изучены методы понижения размерности данных из библиотеки Scikit Learn. Такие как метод главных компонент (PCA), Kernel PCA, Sparce PCA, факторный анализ. Эти методы далее могут быть использованы для подготовки и обработки данных для использования методов машинного обучения.