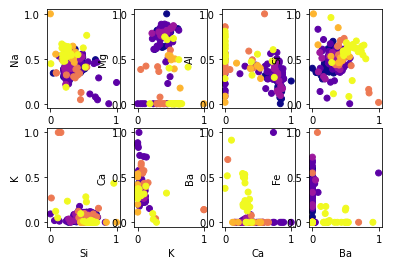
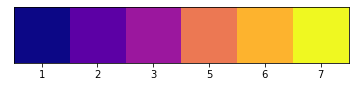
* 1. **Загрузка данных**

1. Данные были загружены в датафрейм и разделены на описательные признаки (*Ri*, *Na*, *Mg*, *Al*, *Si*, *K*, *Ca*, *Ba*, *Fe*) и признак, отображающий класс (*Type*).
2. Данные были нормированы к интервалу *[0; 1]* при помощи *minmax\_scale*.
3. Были построены диаграммы рассеяния для пар признаков.



1. Было установлено соответствие между цветом точки и классом признака (для однозначности использована цветовая схема *plasma*).

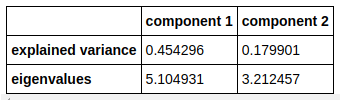


1. **Метод главных компонент**

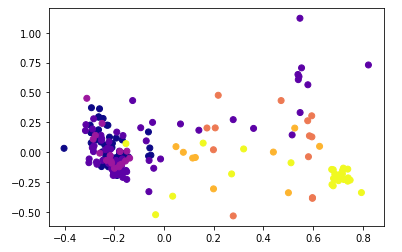
Метод главных компонент представлякт из себя способ линейного уменьшения размерности и проецирования их в пространство более низкой размерности. Входные данные центрируются (вычитанием из среднего), но не масштабируются для каждого объекта перед применением *SVD*.

PCA линейно преобразует данные в новые признаки, которые не коррелируют друг с другом.

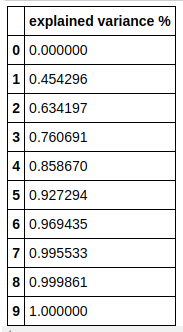
1. С использованием метода главных компонент (*PCA*) было проведено понижение размерности пространства до размерности 2.
2. Были выведены значения объясненной дисперсии в процентах и собственные числа, соответствующие компонентам.



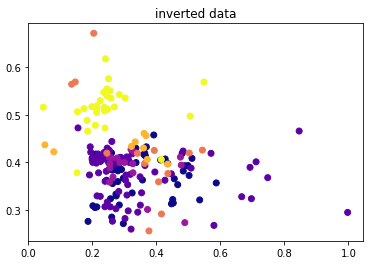
1. Была построена диаграмма рассеяния данных после применения метода главных компонент.



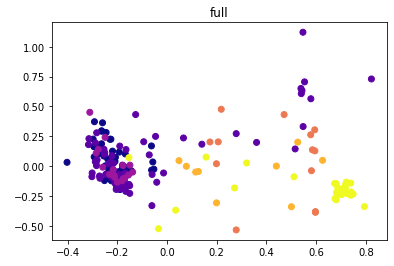
1. По диаграмме рассеивания видно, что линейной зависимости данных нет.
2. При помощи изменения количества компонент, было определено количество, при котором компоненты объясняют не менее 85% дисперсии данных (4).

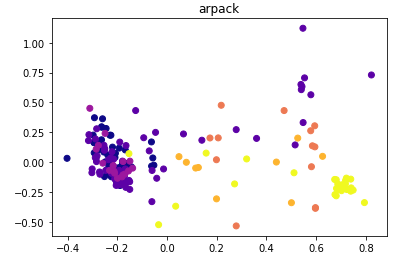


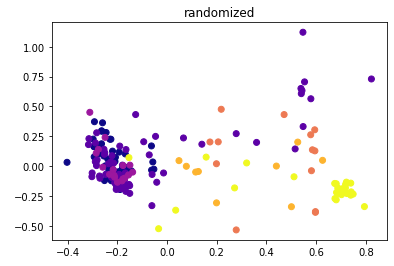
1. С использованием метода *inverse\_transform* данные были восстановлены данные, а также выполнено сравнение с исходными данными. Была построена диаграмма рассеяния после восстановления данных. Можно сделать вывод о том, что, при размере объясненной дисперсии в 85%, различия между исходными и восстановленными данными достаточно малы.



1. Было выполнено исследование метода *PCA* при различных значениях *svd\_solver*. Из построенных диаграмм рассеивания видно, что на данном наборе данных значение *svd\_solver* почти не влияет на конечный результат. При значении ‘*full*’ выполняется разложение алгоритмом *LAPACK*, а затем определяется количество компонент. При значении ‘*arpack*’ выполняется разложение алгоритмом *ARPACK* для заданного числа компонент. При значении ‘*randomized*’ выполняется разложение по методу Халко.

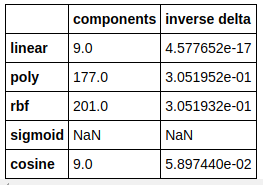


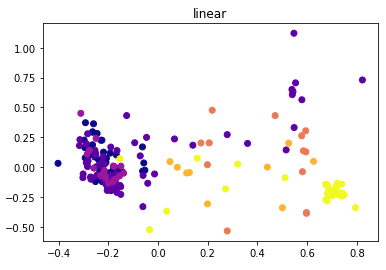


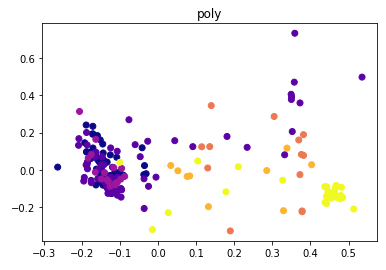
1. **Модиф****икации метода главных компонент**

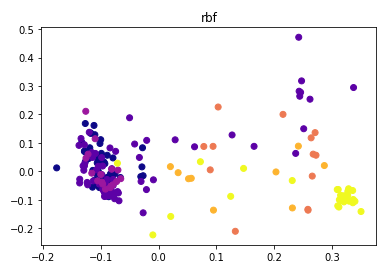
Алгоритм *KernelPCA* производит уменьшение нелинейной размерности за счет использования ядер. **Ядро может быть** '*линейное*', '*poly*', '*rbf*', '*сигмоидное*', '*косинусное*', '*предварительно вычисленное*'*,* по умолчанию используется *линейное* ядро.

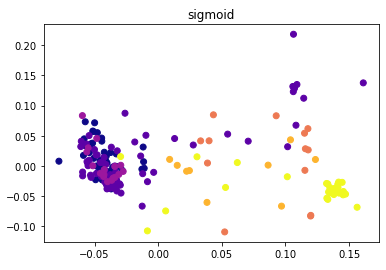
1. По аналогии с *PCA* было проведено исследование *KernelPCA* при различных параметрах *kernel*. В результате исследования можно сделать вывод о том, что собственные числа и количество компонент отличаются из-за разных функций ядра, однако значения объясненной дисперсии различаются в небольших пределах от 1 до 2%.

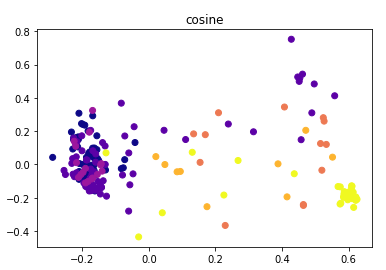






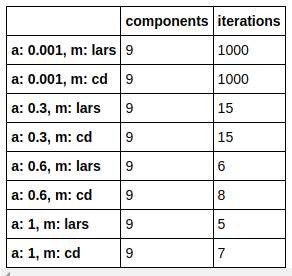


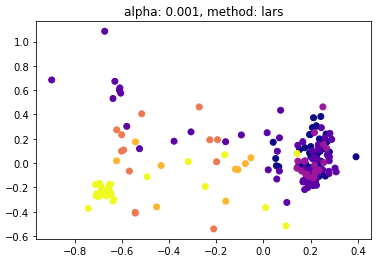


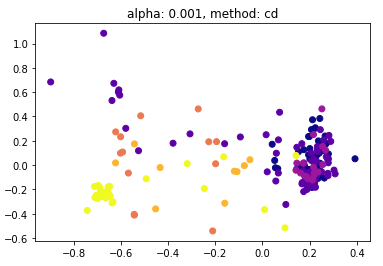


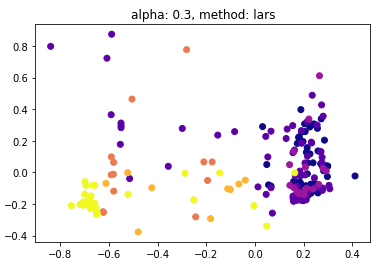
1. *KernelPCA* ведет себя также, как *PCA* при использовании *linear* ядра.
2. Аналогично был исследован *SparsePCA*.

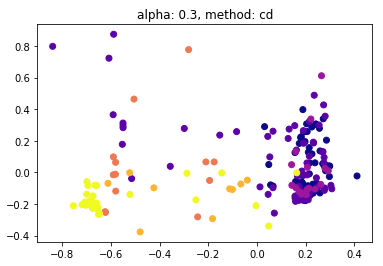
Этот алгоритм находит набор разреженных компонентов, которые могут оптимально реконструировать данные.

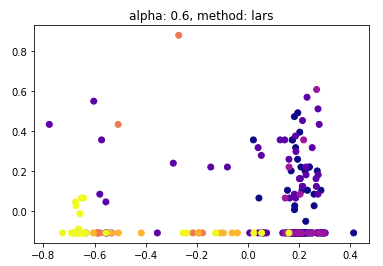


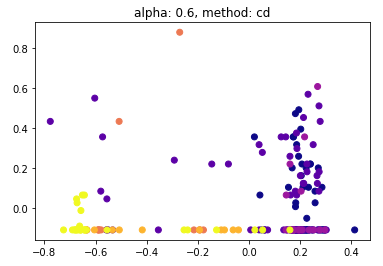


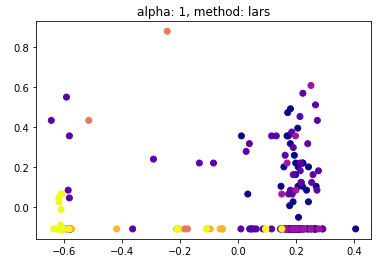


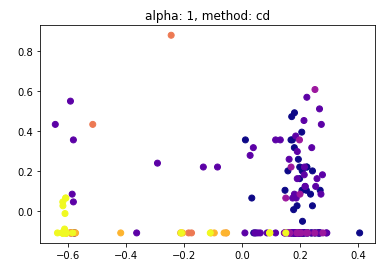








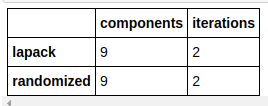




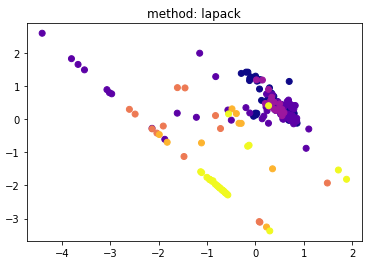
1. Параметр *alpha* контролирует разреженность данных, а параметр *method* — метод работы алгоритма. Понижая *alpha* до 0, мы приблизим результат работы алгоритма к результату работы обычного *PCA*. Влияние параметра *method* на результат работы алгоритма для данного габора данных несущественно.
2. **Факторный анализ**

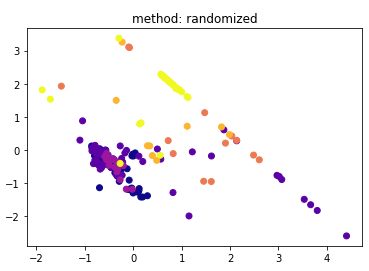
Предполагается, что наблюдения вызваны линейным преобразованием неких факторов и добавлением гауссовского шума. Метод оценивает вероятность преобразования скрытых признаков в наблюдаемые.

1. Было проведено понижение размерности с импользованием факторного анализа *FactorAnalysis*.



1. Были построены диаграммы рассеяния для двух возможных методов применения факторного анализа: *lapack* и *randomized*.





1. Компонент *PCA* представляет собой линейную комбинацию наблюдаемой переменной. В свою очередь в *FactorAnalysis* наблюдаемые переменные представляют собой линейные комбинации ненаблюдаемых переменных или фактора. Также можно сделать вывод, что, если *PCA* - это метод уменьшения размерности, то *FactorAnalysis* фокусируется на поиске скрытых переменных. Влияние параметра *method* на результат работы алгоритма для данного габора данных несущественно.

**Выводы**

В процессе выполнения лабораторной работы были изучены методы понижения размерности и факторного анализа данных из библиотеки *Scikit Learn*.

Для *PCA* (метод главных компонент) был выявлен факт того, что количество компонент напрямую влияет на процент объяснённой дисперсии.

*KernelPCA* применяется для снижения нелинейной размерности за счет использования различных ядер. Может использоваться для поиска нелинейных зависимостей в данных. Для данных, рассмторенных в данной лабораторной работе, смена ядра на результат практически не влияла.

*SparsePCA* находит набор разреженных компонентов, которые могут оптимально восстановить данные. Степень разреженности регулируется коэффициентом *alpha*.

Были изучены отличия факторного анализа от метода главных компонент. Основное преимущество факторного анализа перед ним *PCA* заключается в том, что он может независимо моделировать дисперсию во всех направлениях входного пространства.