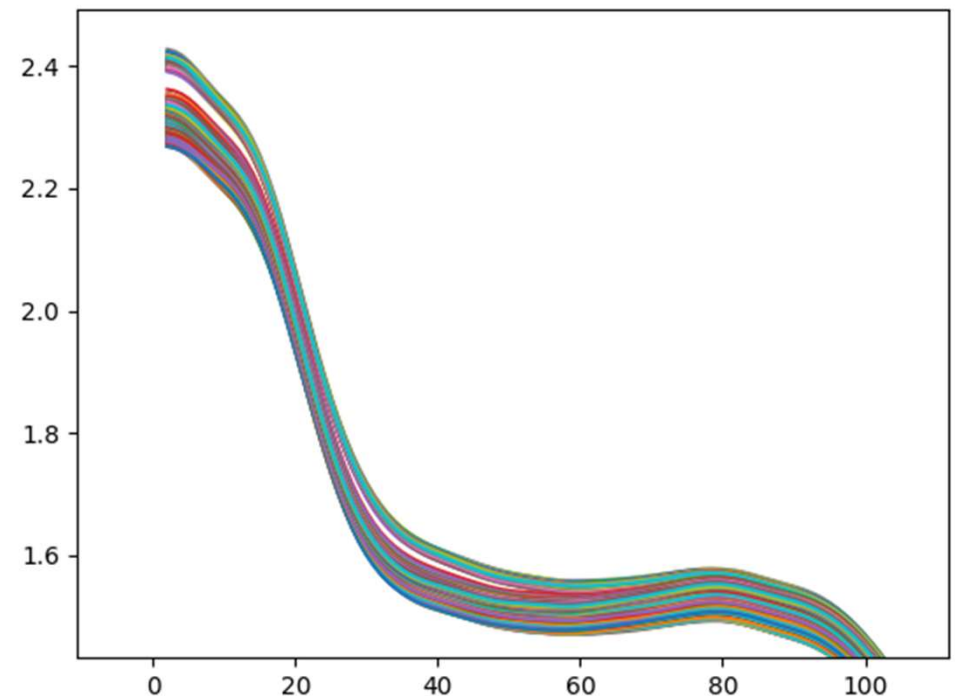
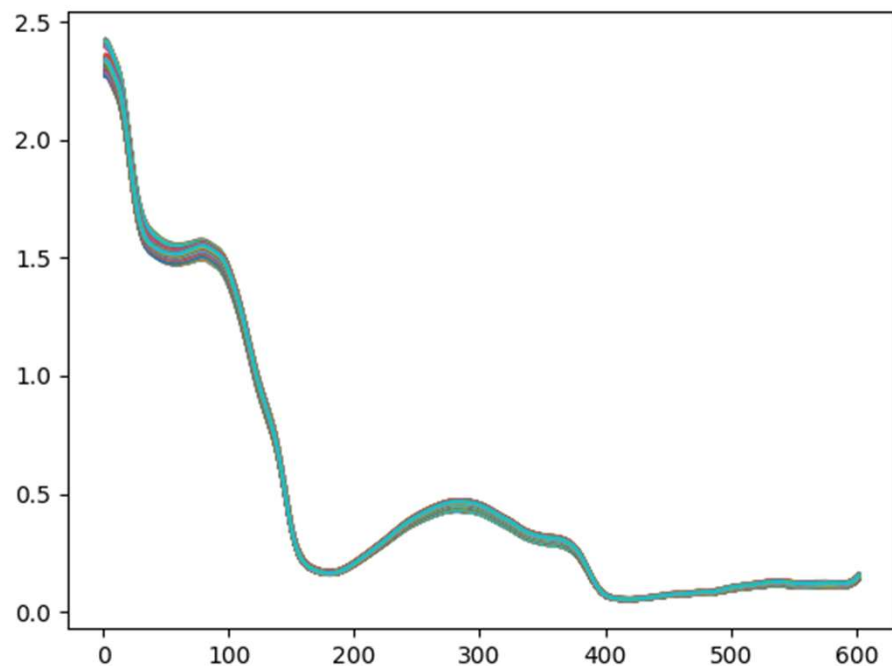


# РСА - “Изобретён Карлом Пирсоном в 1901 году.”

[https://en.wikipedia.org/wiki/Principal\\_component\\_analysis](https://en.wikipedia.org/wiki/Principal_component_analysis)

(pca\_milk.py)

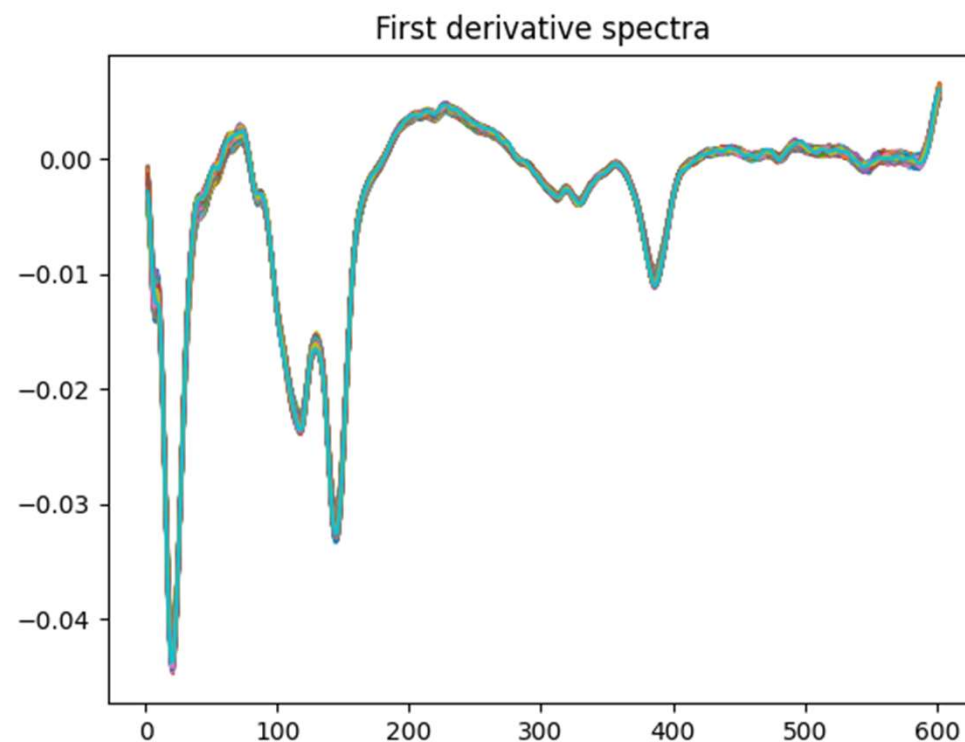
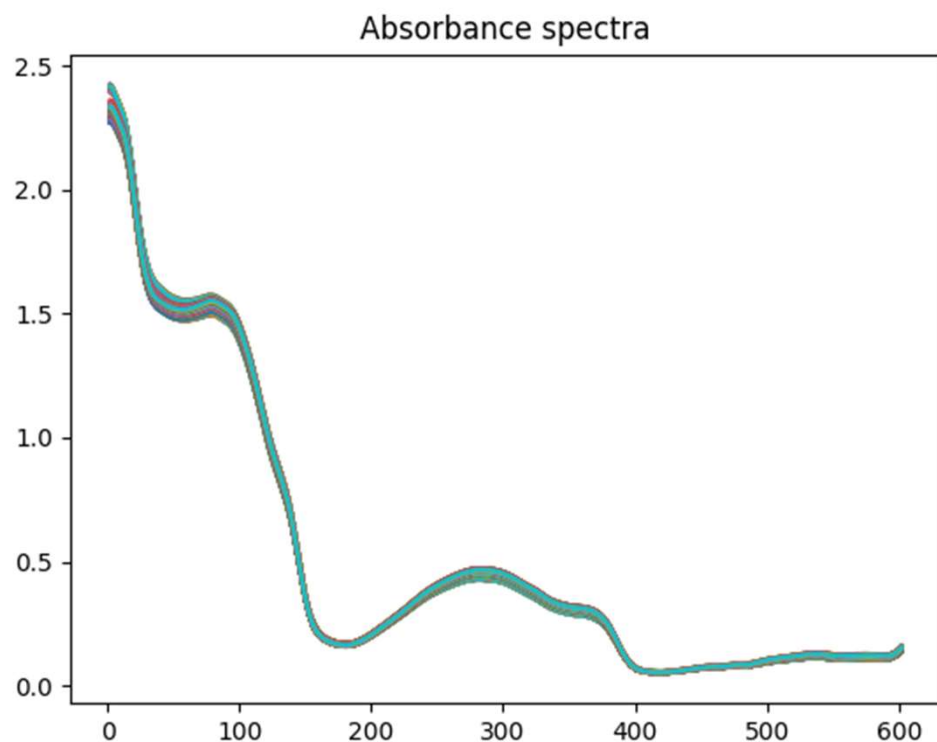


NIR spectra from milk samples with a varying concentration of lactose.

<https://nirpyresearch.com/classification-nir-spectra-principal-component-analysis-python/>

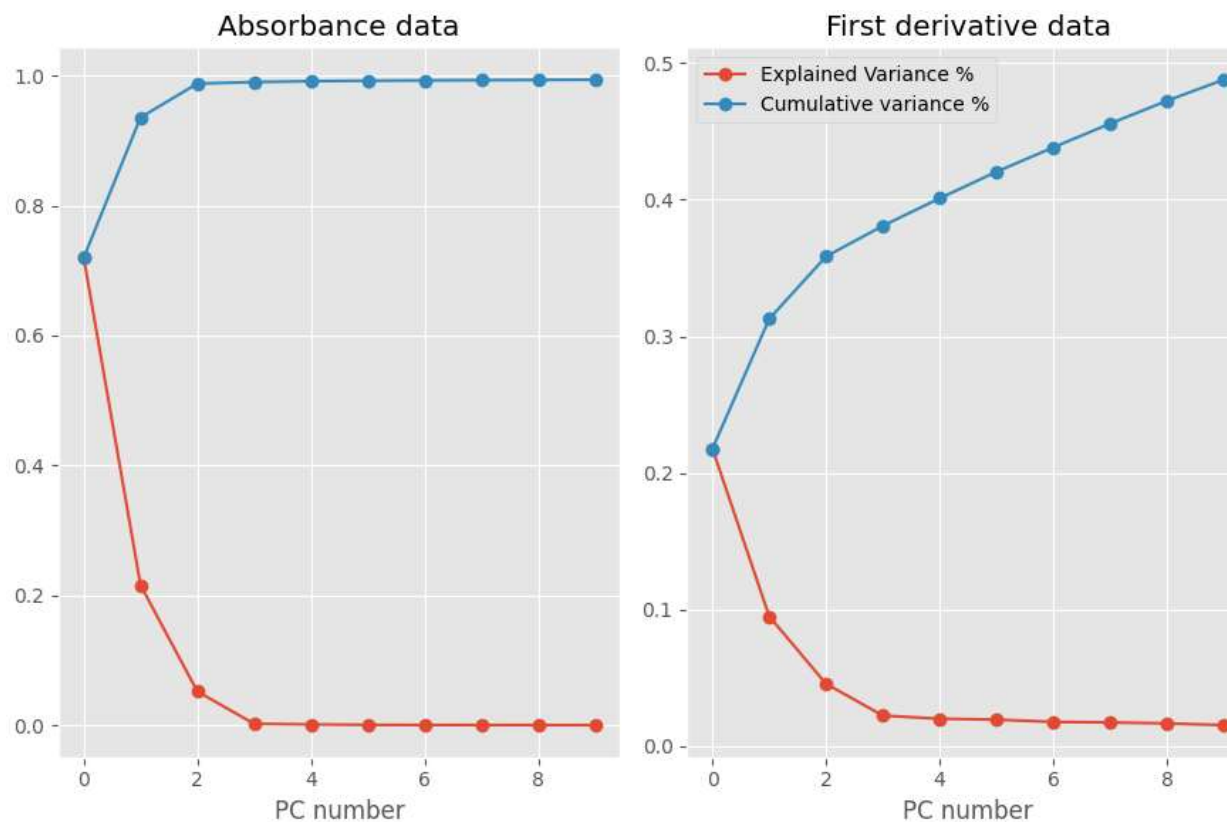
## Классификация спектров

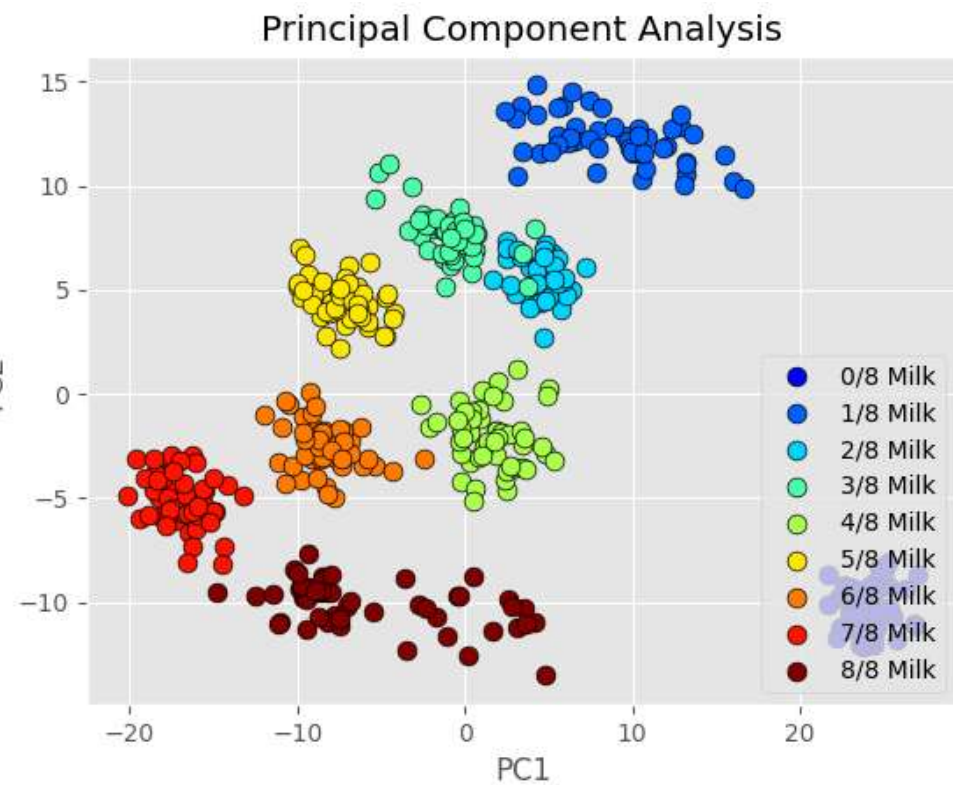
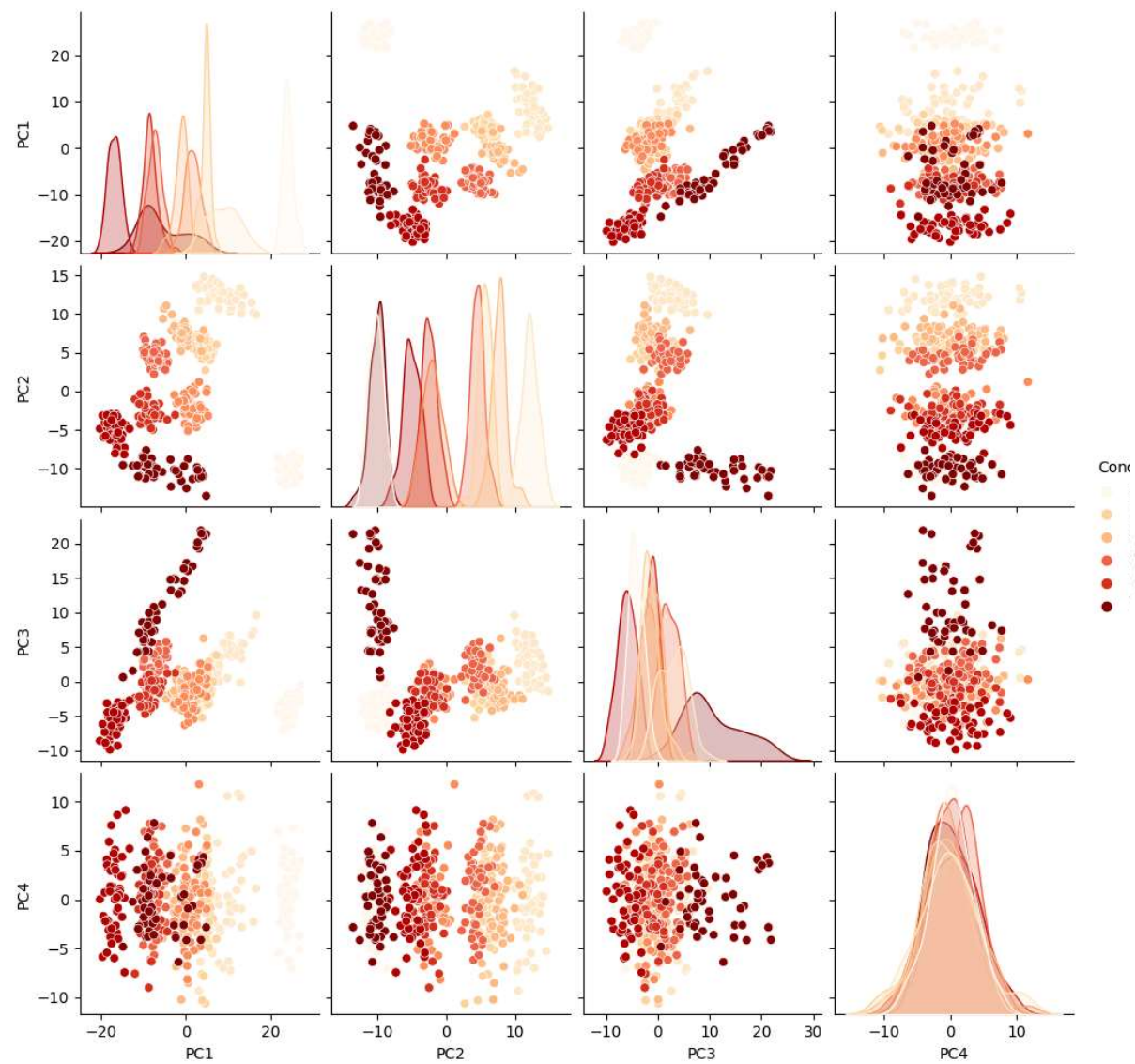
- – информативнее производная
- – желательно сглаживание



## Классификация спектров

- – информативнее производная
- – желательно сглаживание
- – ОПТИМАЛЬНОЕ КОЛИЧЕСТВО КОМПОНЕНТ

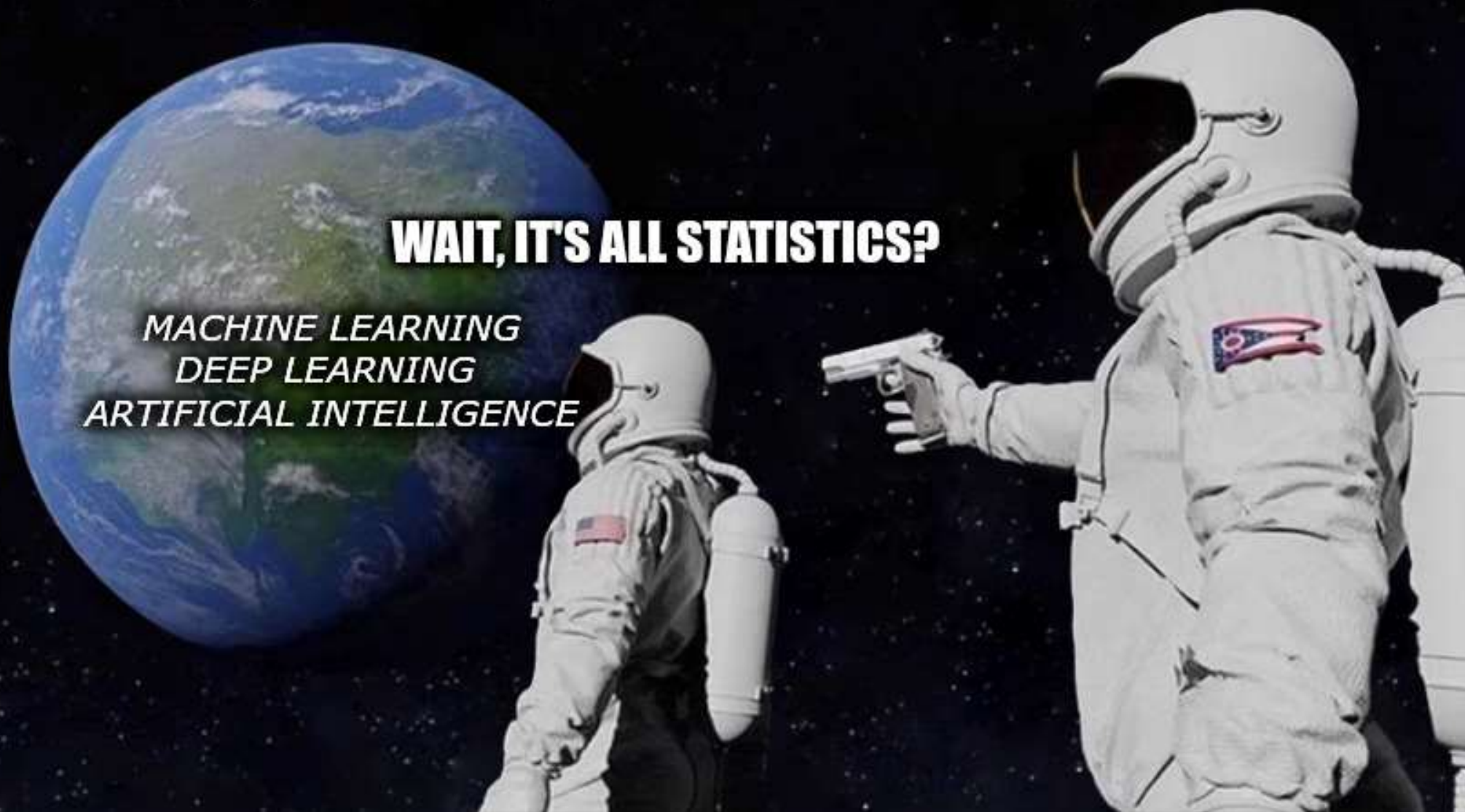




**ALWAYS HAS BEEN**

**WAIT, IT'S ALL STATISTICS?**

*MACHINE LEARNING  
DEEP LEARNING  
ARTIFICIAL INTELLIGENCE*

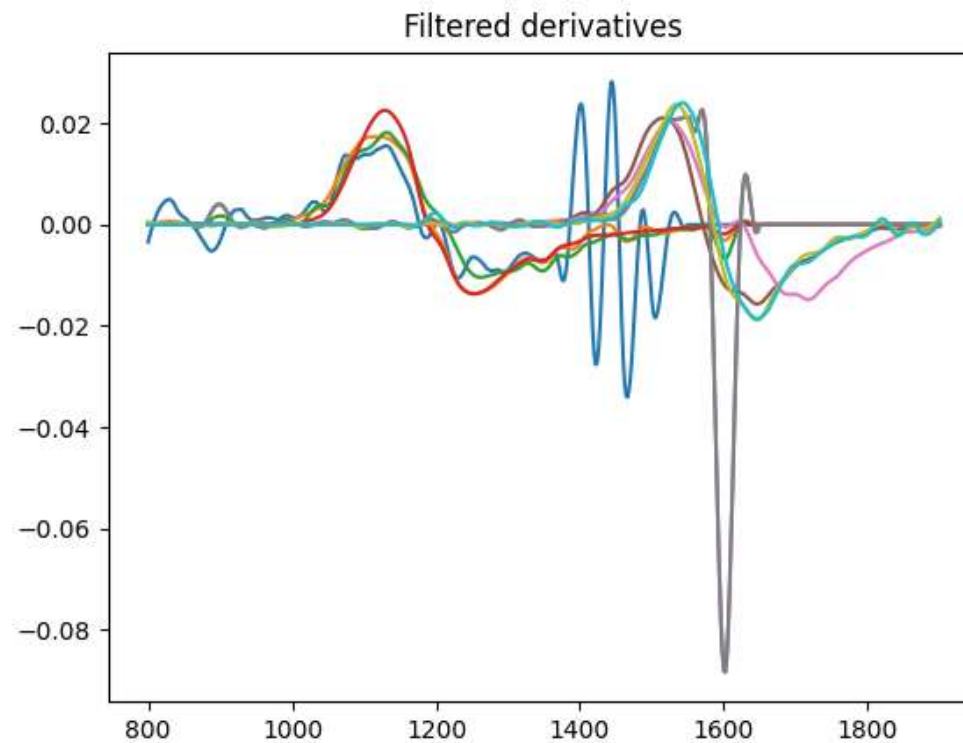
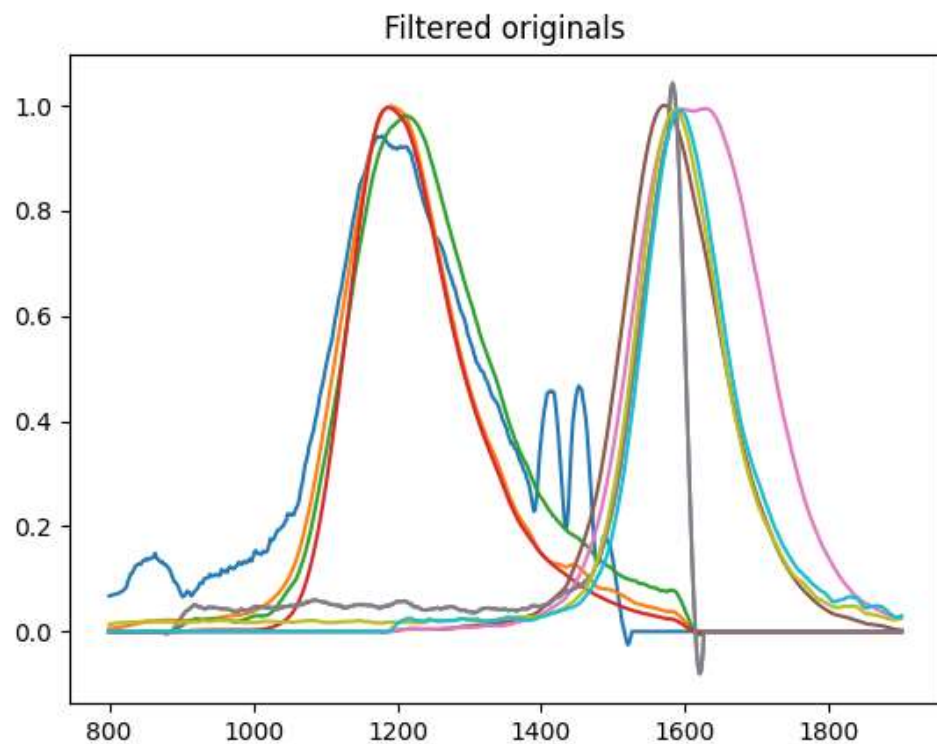


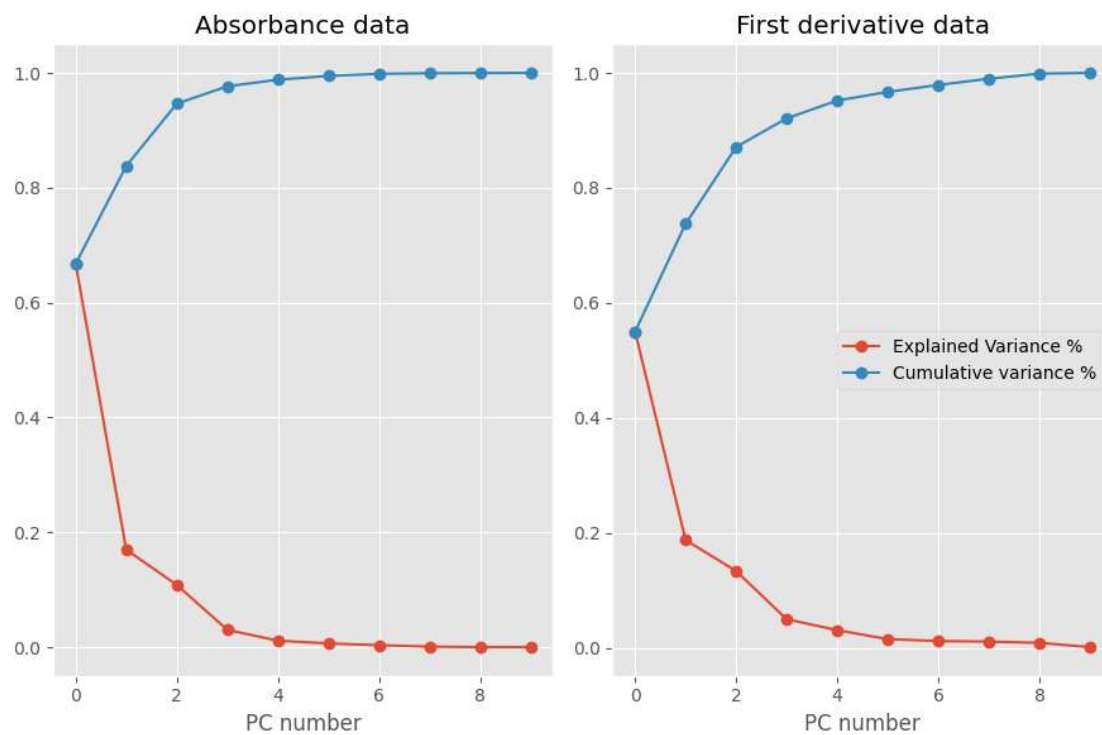


# PCA & PbS

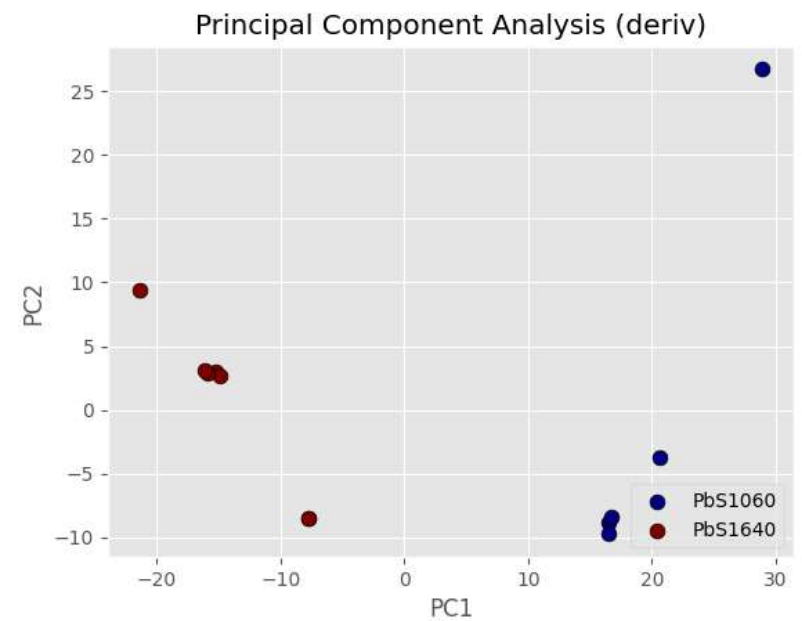
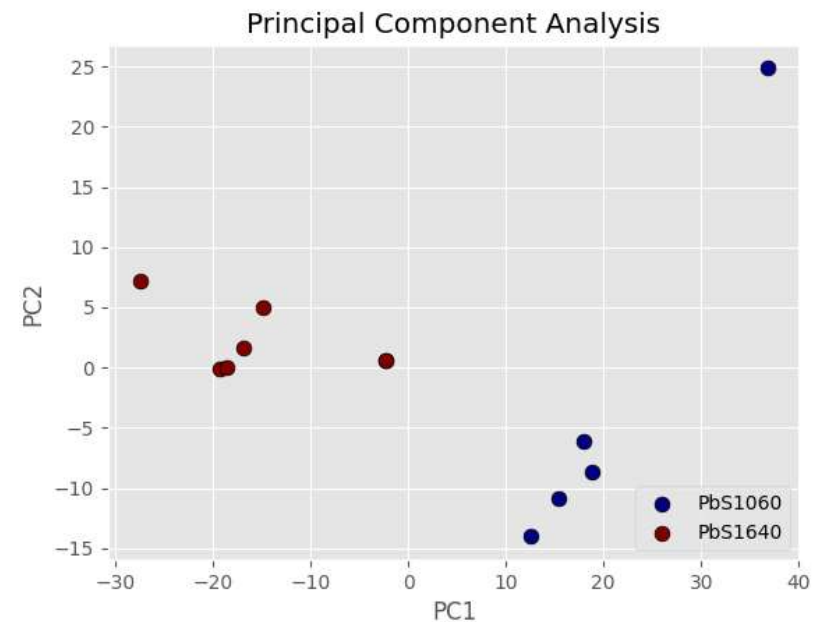
pca\_pbs.py

- Filtered (для производной фильтрация и шумы критичны)

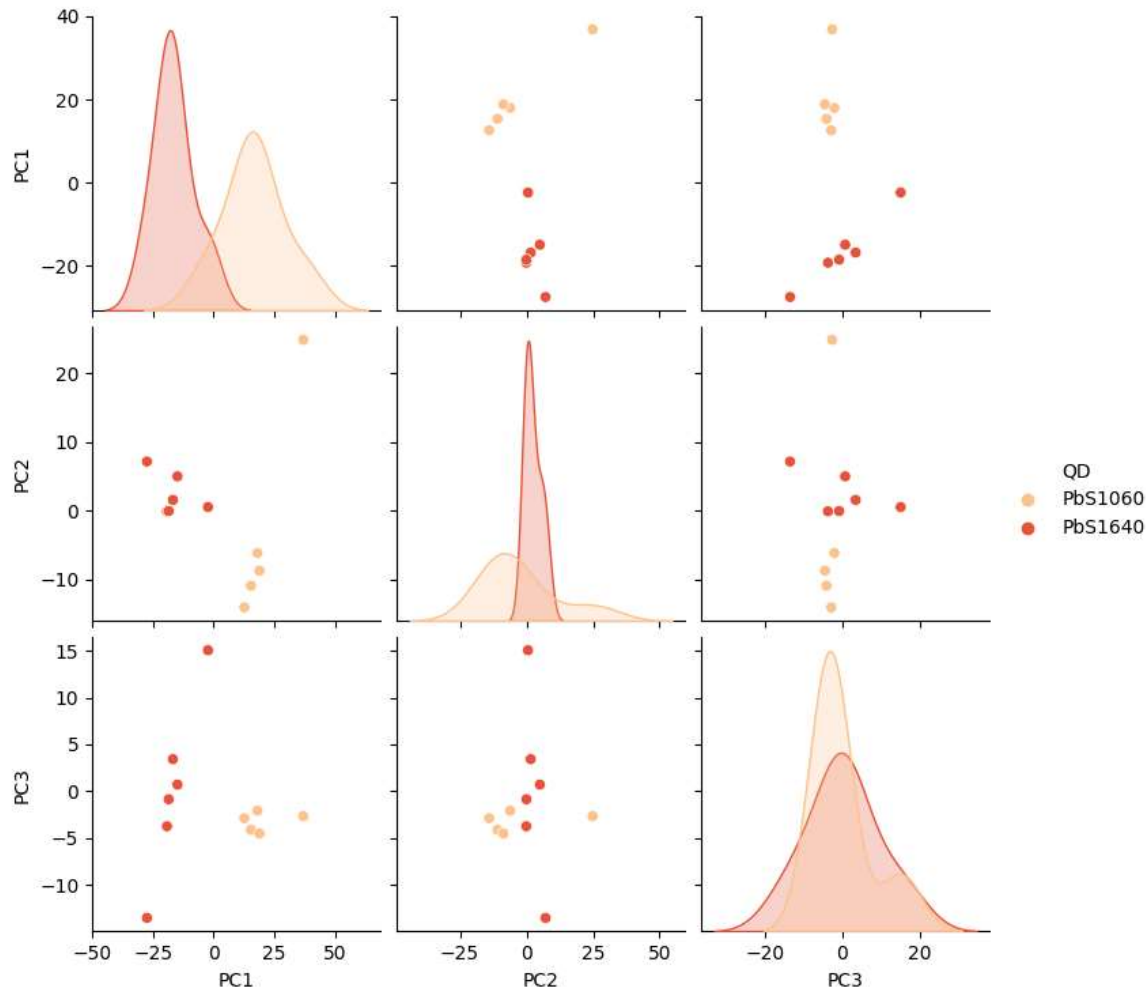




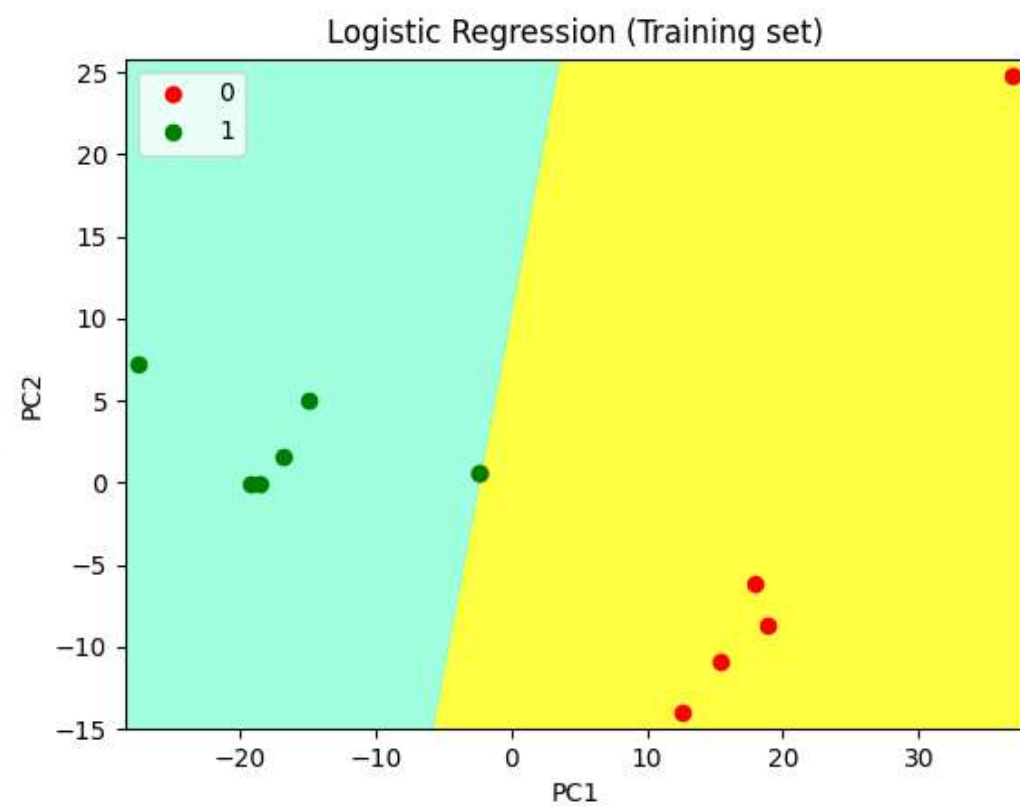
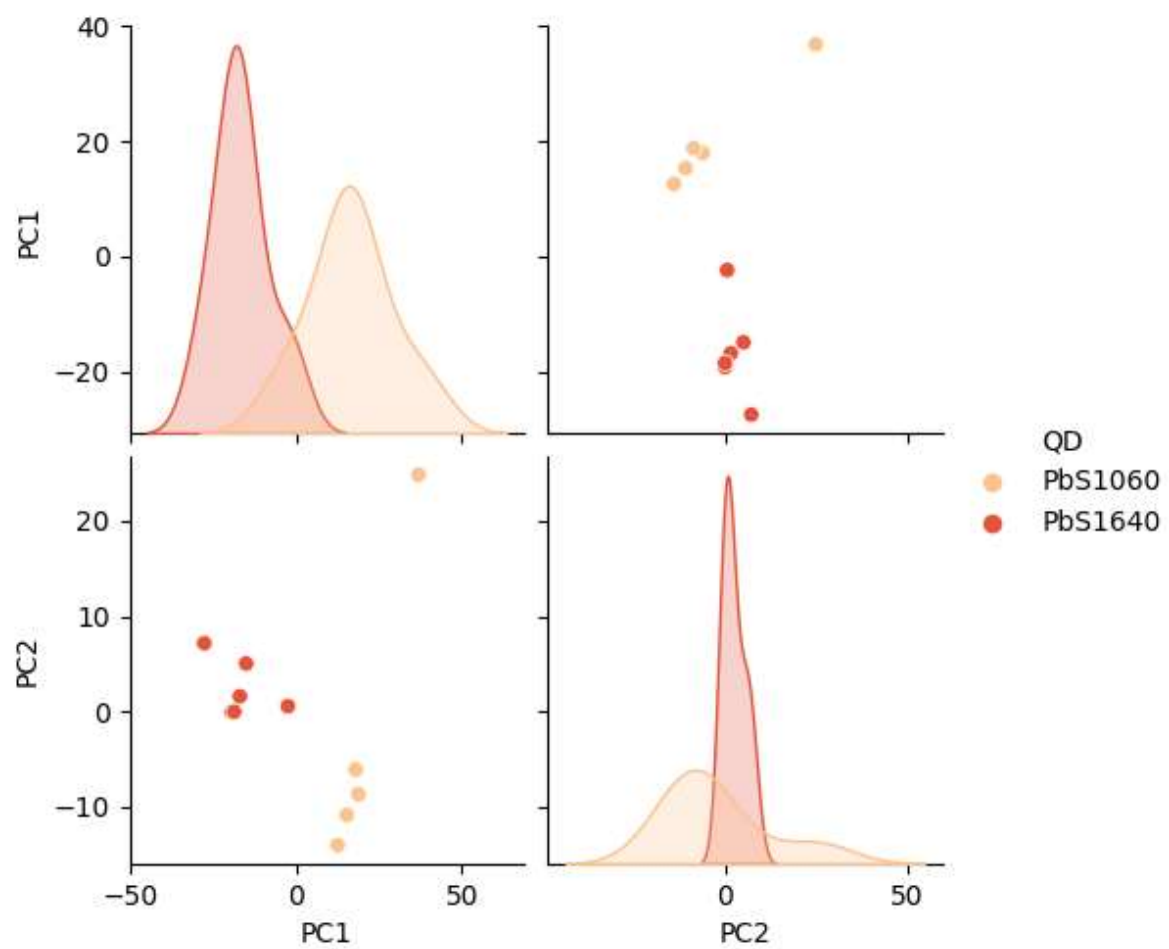
- Из-за шумов производная (nfeat2) не факт что лучше исходной (nfeat1)



# Layered kernel density estimate (KDE)







Это все хорошо, но где  
распознавание? Где обучение?)

<https://www.geeksforgeeks.org/principal-component-analysis-with-python/>

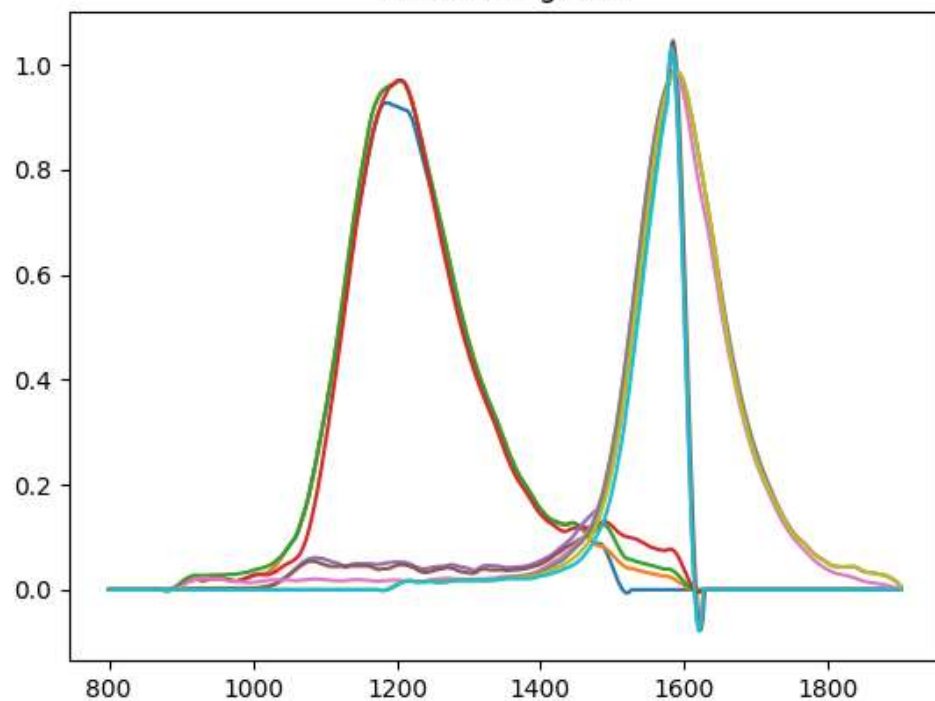
- Логистическая регрессия
- 2 компоненты PCA
- 10 – обучение, 2 – проверка.
- Добавил медианную фильтрацию

(pca\_pbs\_logreg\_realtest.py)

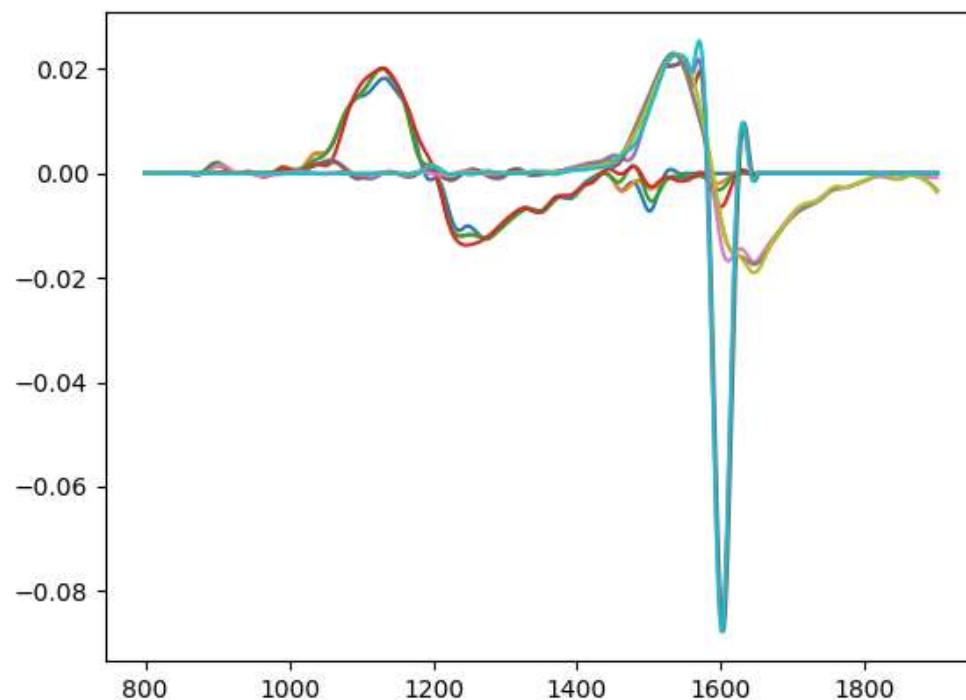
nfeat = nfeat1  
test = test1

nfeat = nfeat2  
test = test2

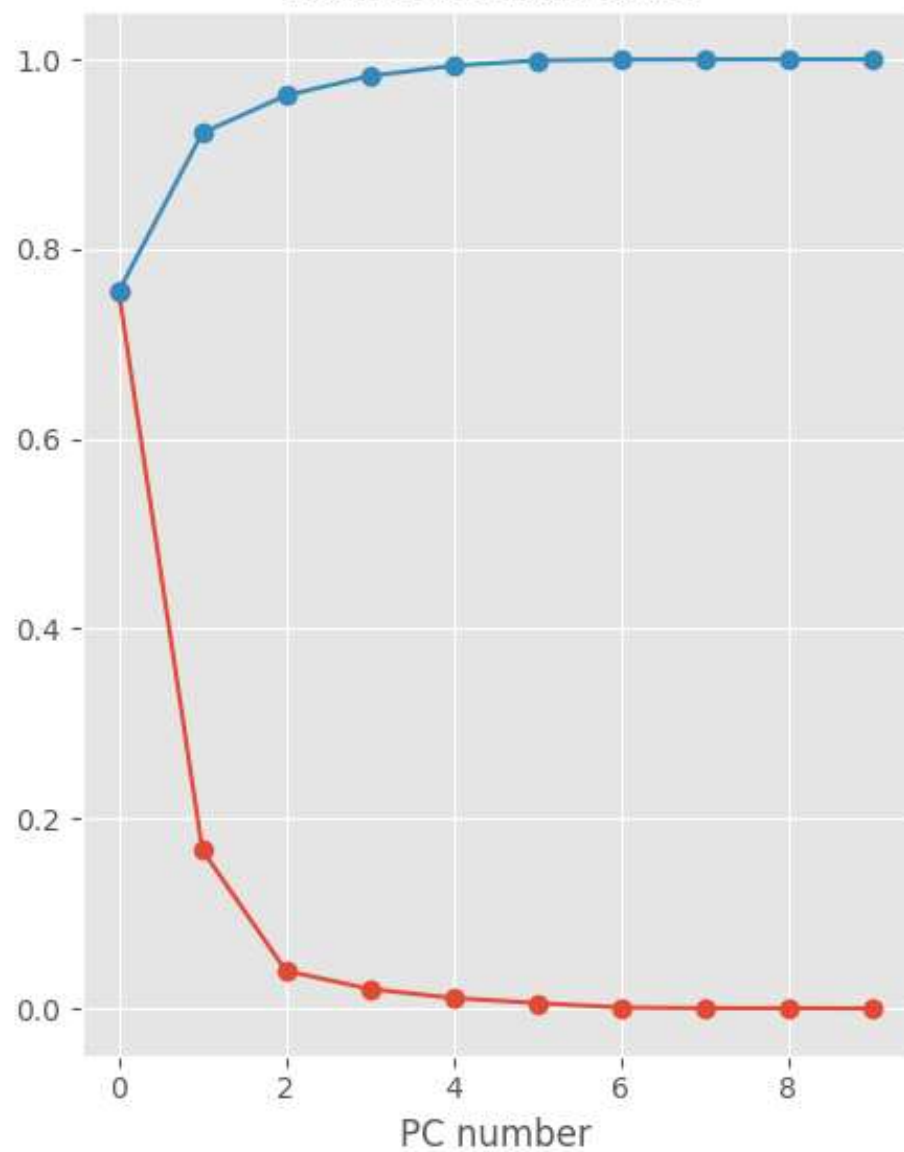
Filtered originals



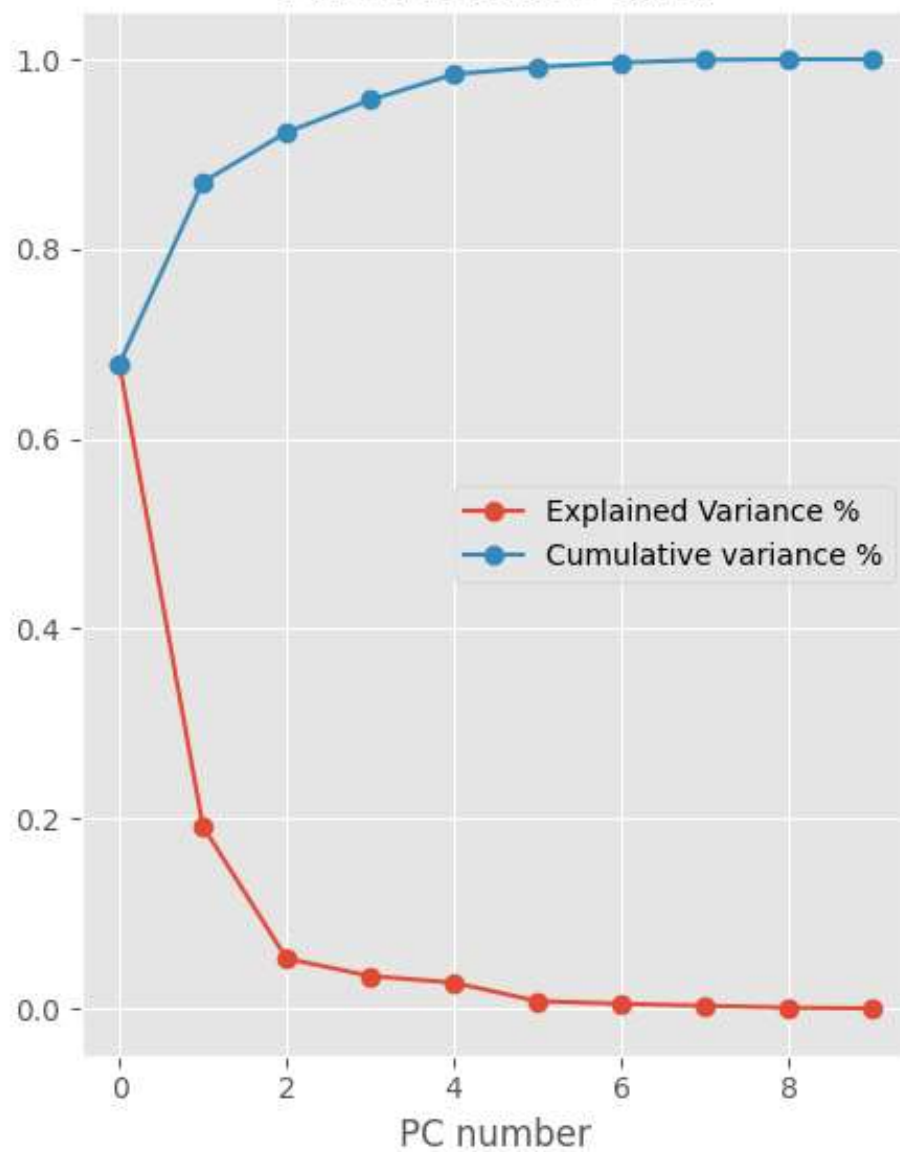
Filtered derivatives



Luminescence data

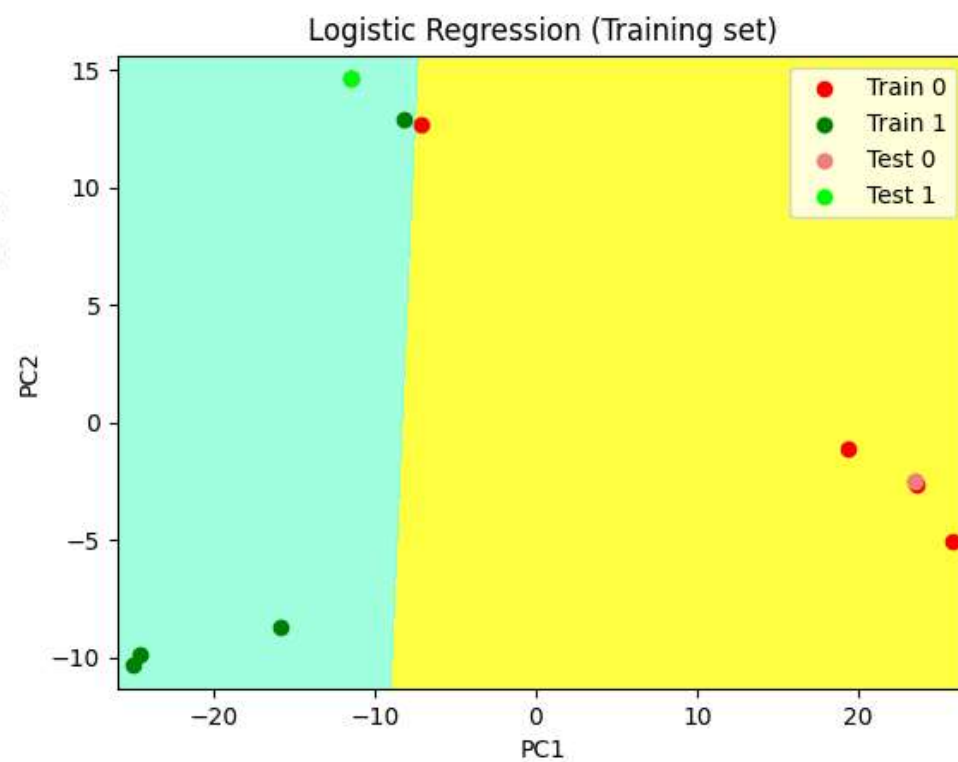
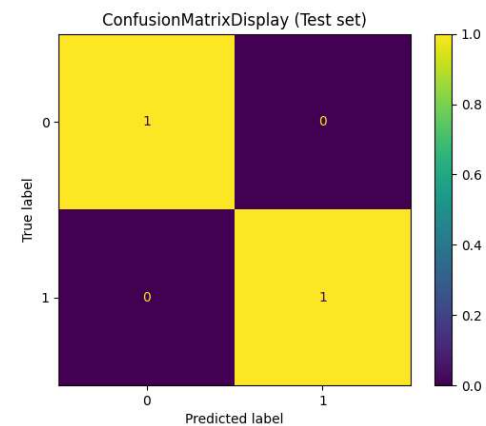
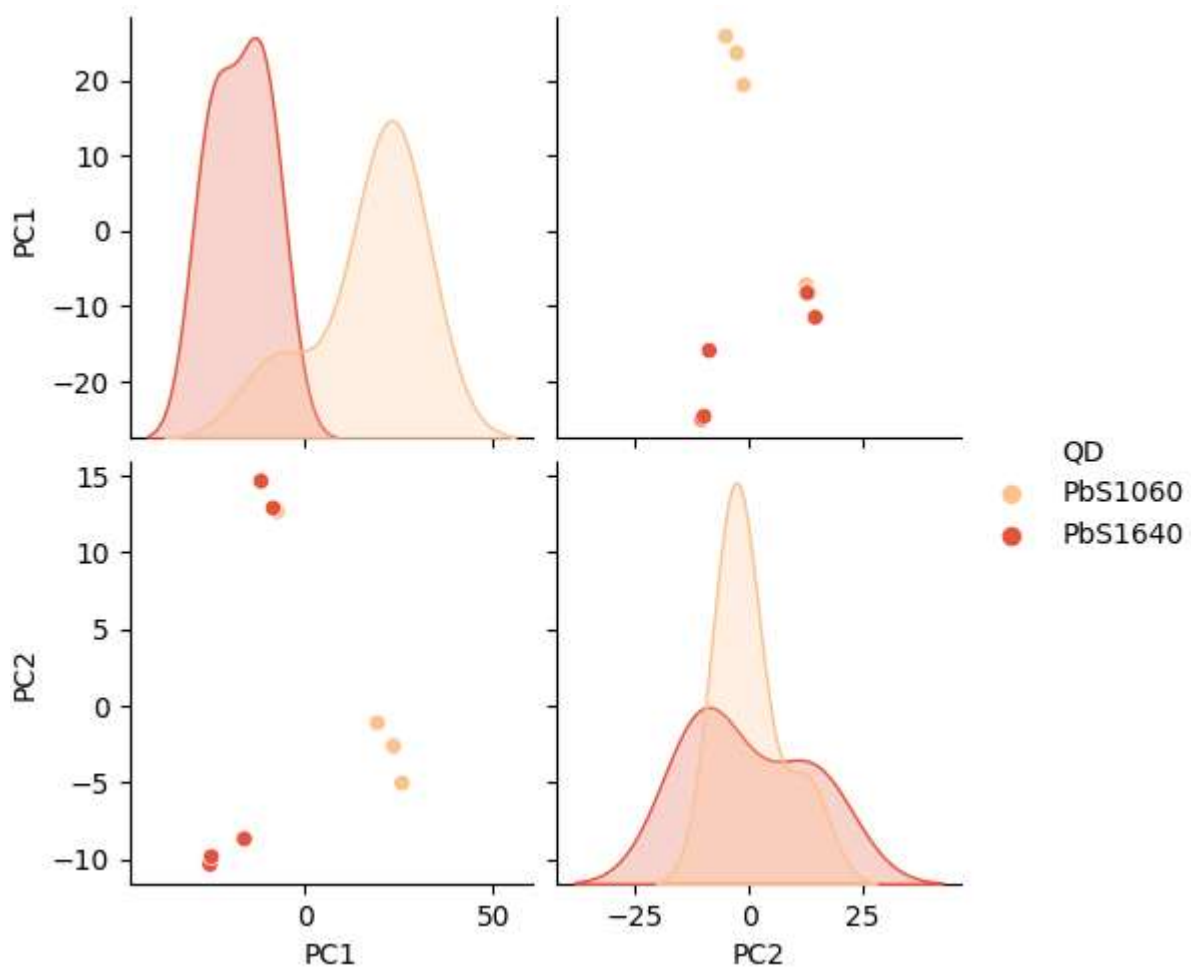


First derivative data



nfeat = nfeat1  
test = test1

- Оригинал спектра



nfeat = nfeat2  
test = test2

- Производная спектра

