COURSE NAME: Hillel - Machine Learning. Lesson 9 (27.03.2023 19:15,) 9. PCA
LECTURER: Yevhen Krasnokutsky
STUDENT: Maryna Lysenko

HW 8 REPORT “PCA use cases”.

- Get the dataset from Kaggle (any tabular dataset you want)
- Make a simple classifier/regressor on the dataset
- Reasonably reduce dataset dimensionality
    - Plot explained variance
    - Explain the chosen number of components
- Retrain the same classifier/regressor on the dataset with reduced dimensionality
- Compare accuracies / MSEs and speed of the two approaches (with and without dimensionality reduction)

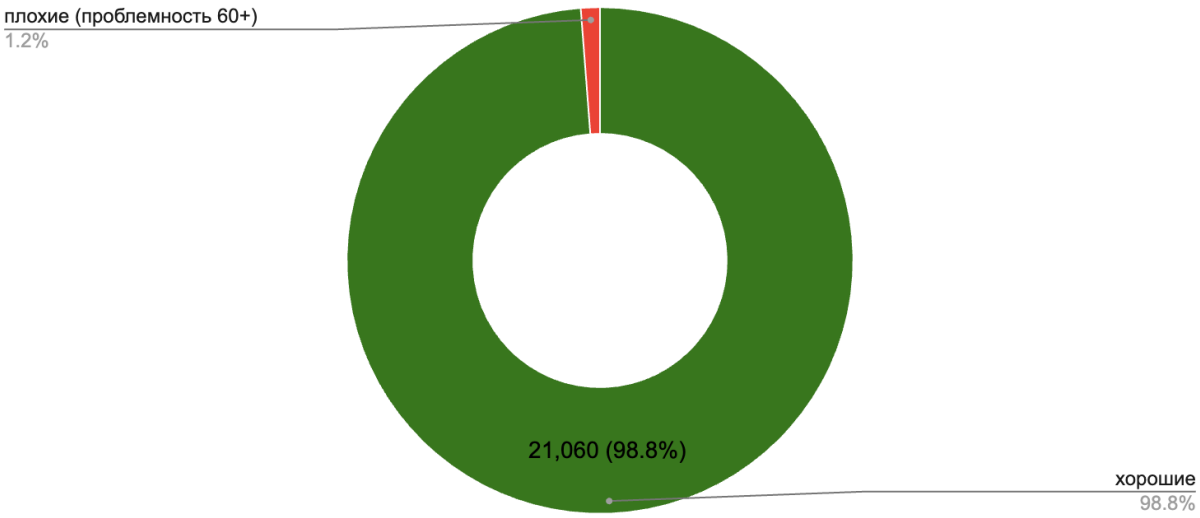
Выбранный датасет

Dataset "Credit Card Approval Prediction"

Задача состоит в построении модели машинного обучения для оценки кредитного риска, то есть прогнозирования того, будет ли заявитель (модель будет применяться одна и к онбордингу новых и к пересчету для повышения лимита уже действующей базы) "хорошим" или "плохим" клиентом. Определение "хорошего" или "плохого" клиента не дано, и необходимо использовать техники, такие как анализ винтажа, для создания меток. В этой задаче также стоит столкнуться с проблемой несбалансированных данных. Обратить внимани, что более сложные методы машинного обучения могут быть менее прозрачными, что затрудняет объяснение принятого решения клиенту и регуляторам.

Подготовка данных

- Проведен EDA
- Проведена предподготовка данных, преобразования фичей
- Выбран и подготовлен target как флаг выхода на просрочку 60+, удалены наблюдения, которые чисто технически еще не могли войти в прострочку
- Проведена балансировка данных при помощи Synthetic Minority Over-Sampling Technique(SMOTE), так как конверсия выхода на просрочку 60+ по нашей базе составляла всего лишь 1,2%



Выбор базовой модели

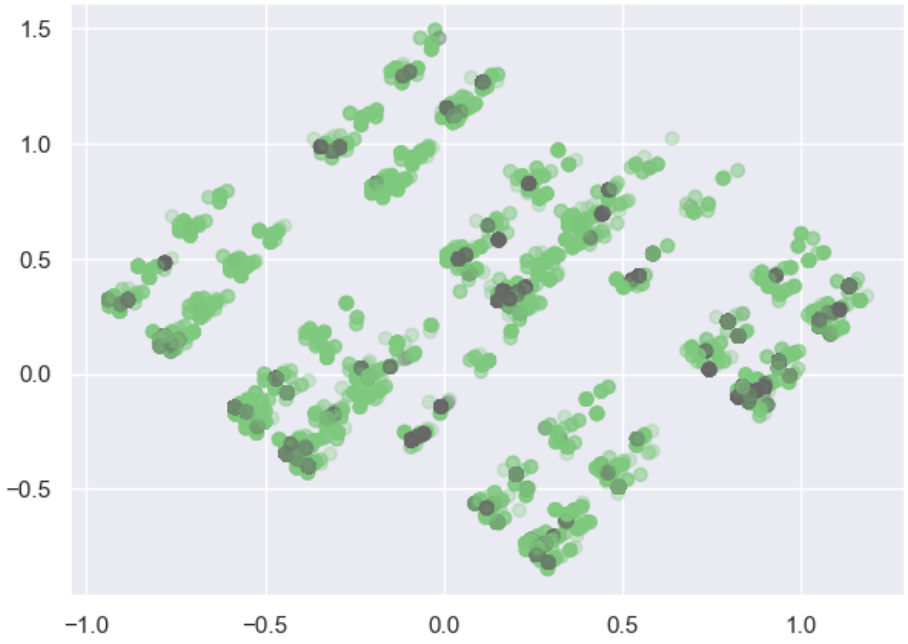
- Построено несколько моделей для выбора той, на которой будет проводиться експеримент сравнения результата с и без PCA

Alghorithm	Accuracy Score
CatBoost	75.32%
XGBClassifier	74.88%
DecisionTreeClassifier	72.55%
RandomForestClassifier	71.31%
LogisticRegression	64.74%
SVC	64.01%

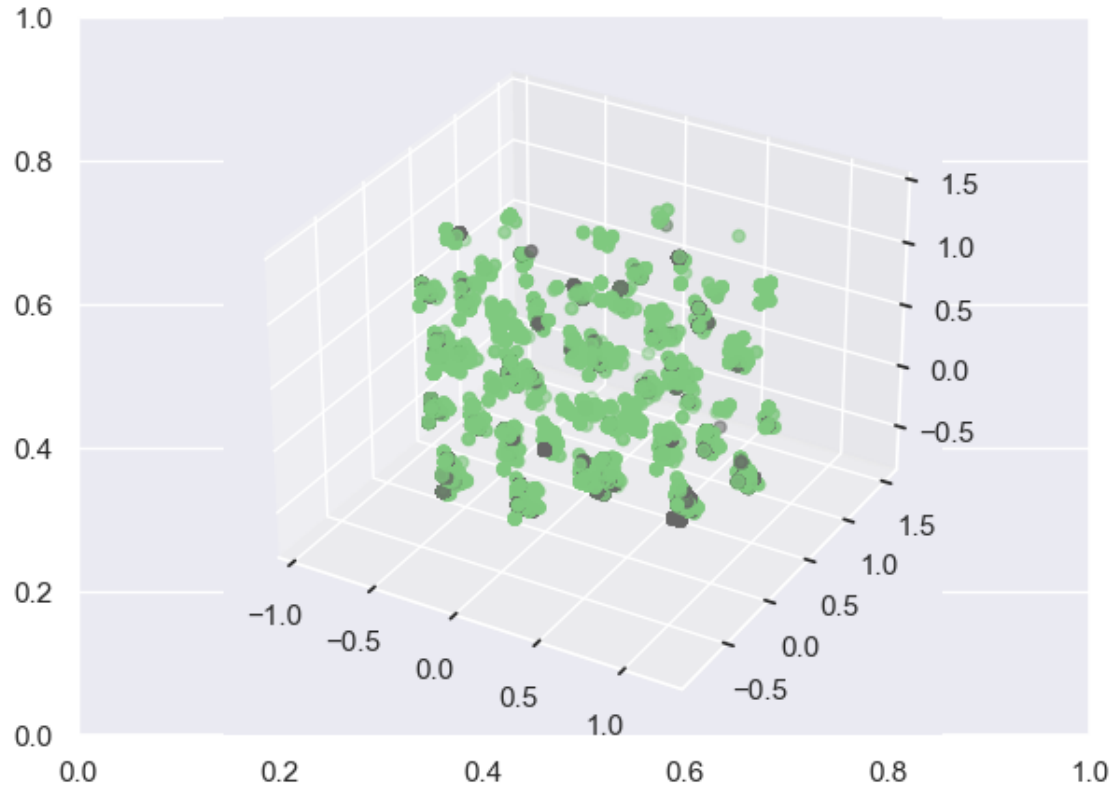
Для эксперимента выбрана модель CatBoost

PCA

2 компоненты:

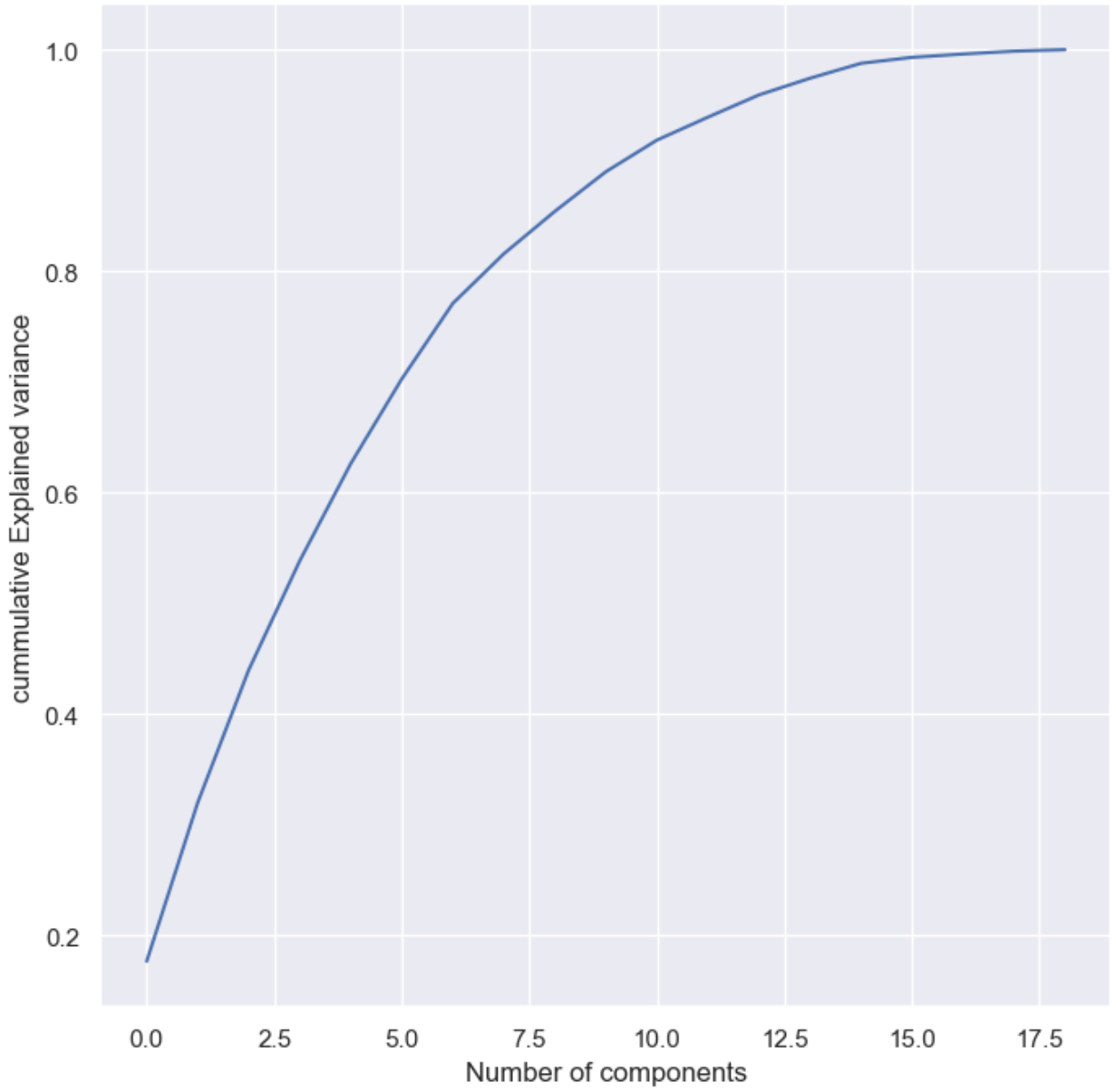


3 компоненты:



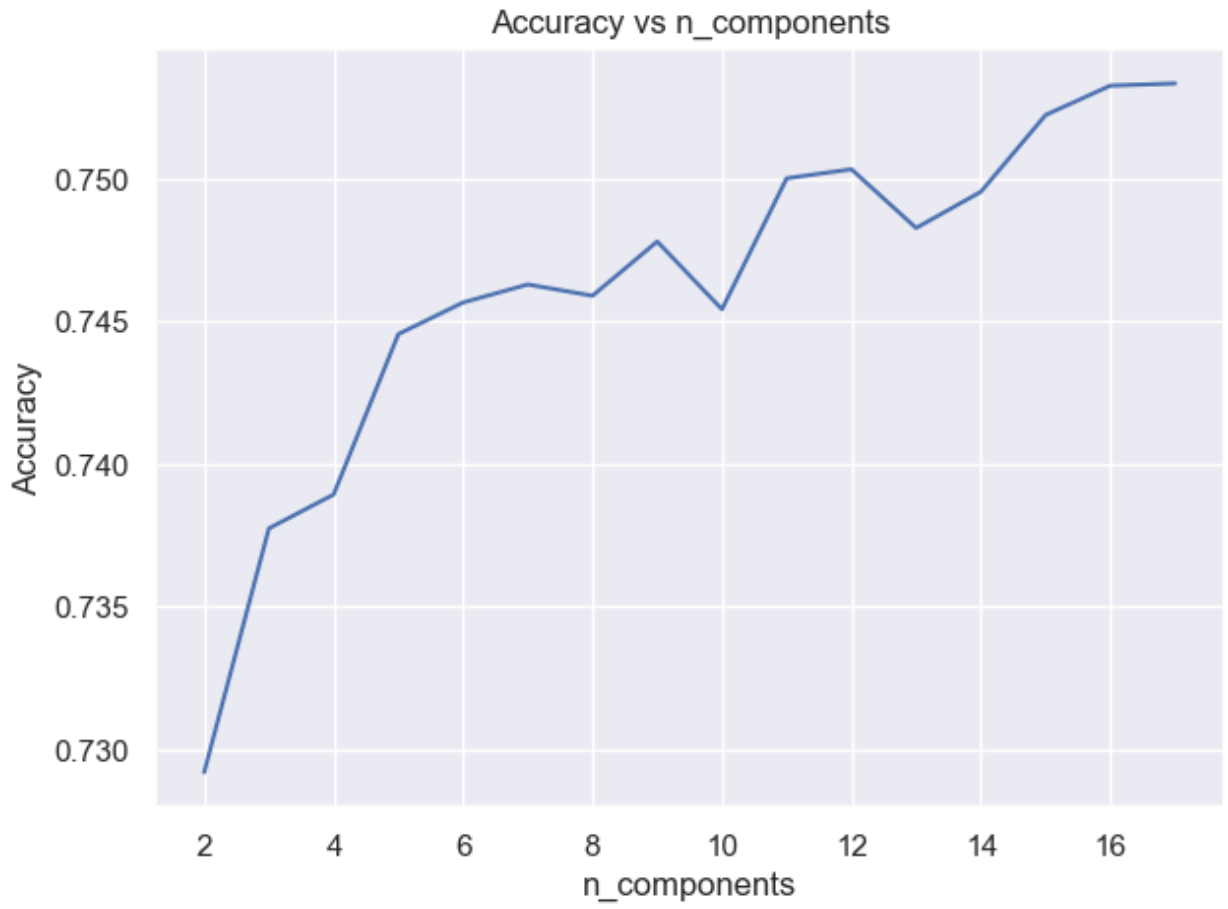
Как видно из визуализаций PCA различить границы между целевыми классами нельзя, но у нас есть "облака" где нет плохого класса. Итак, похоже, что производительность моделей классификации упадет не очень сильно если снизить размерность данных, но мой класс задач не терпит падения точности и лучше обучаться дольше, чем потерять качество.

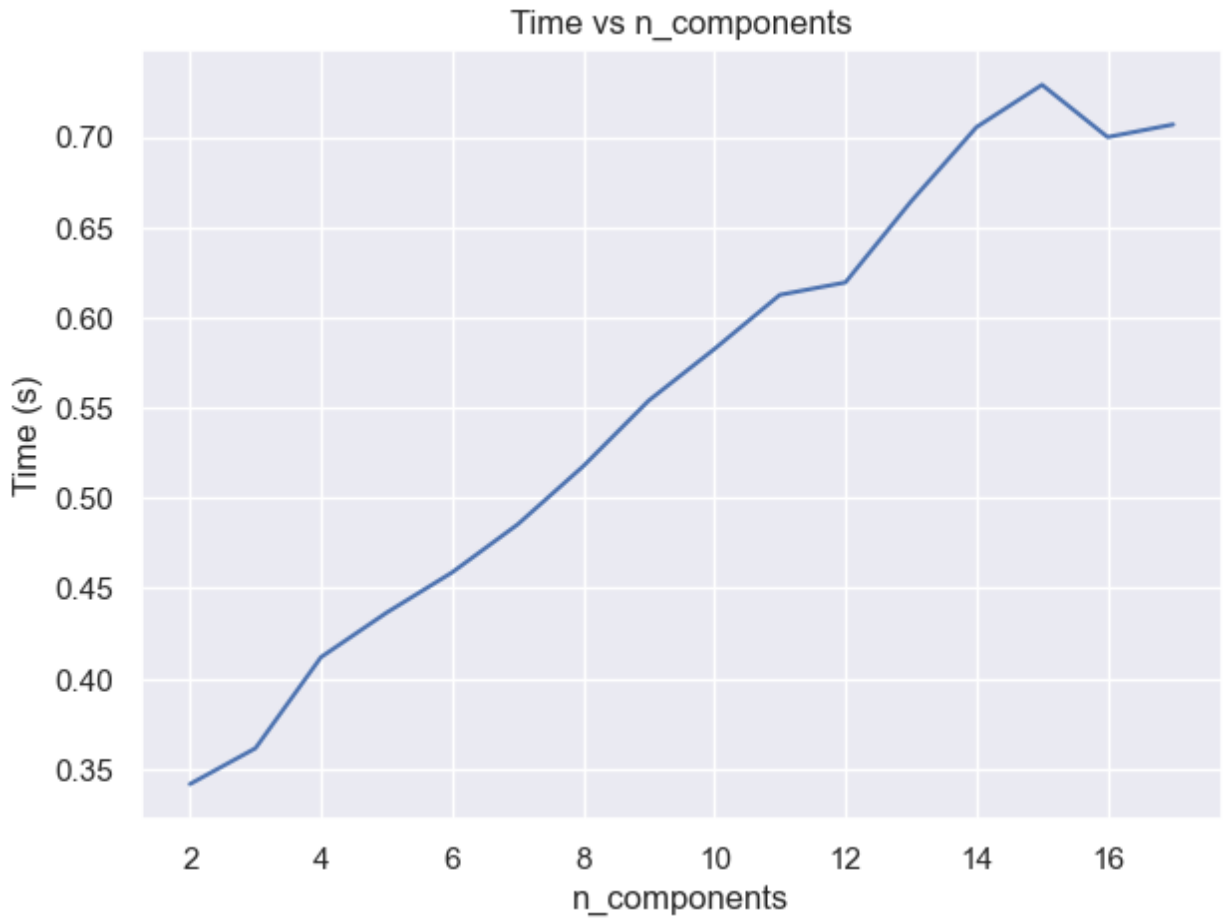
Reasonably reduce dataset dimensionality



Для моей задачи потеря в точности - это критично, поэтому я выберу 16 компонент чтобы практически все описать

Но до этого эксперимента ради посмотрю что будет с точностью и скоростью, если выбираем все возможные уменьшения размерности

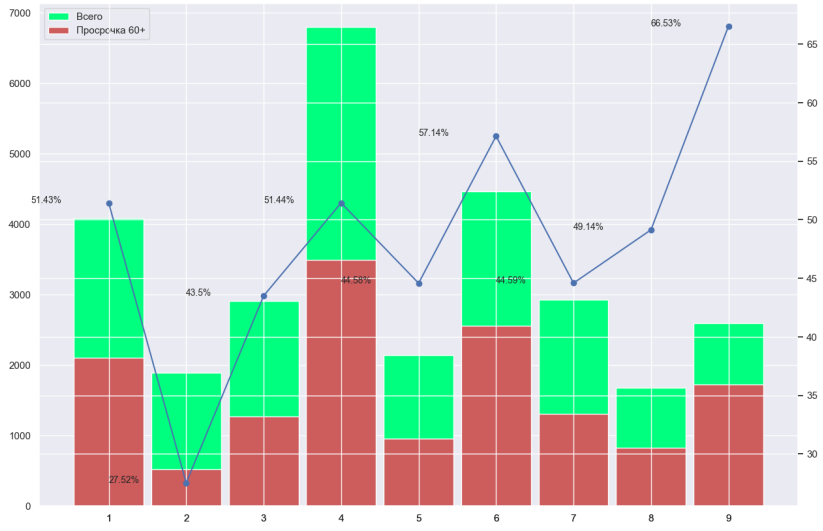

Accuracy vs n_components
0.750
0.745
0.740
0.735
0.730
Accuracy
2
4
6
8
10
12
14
16
n_components



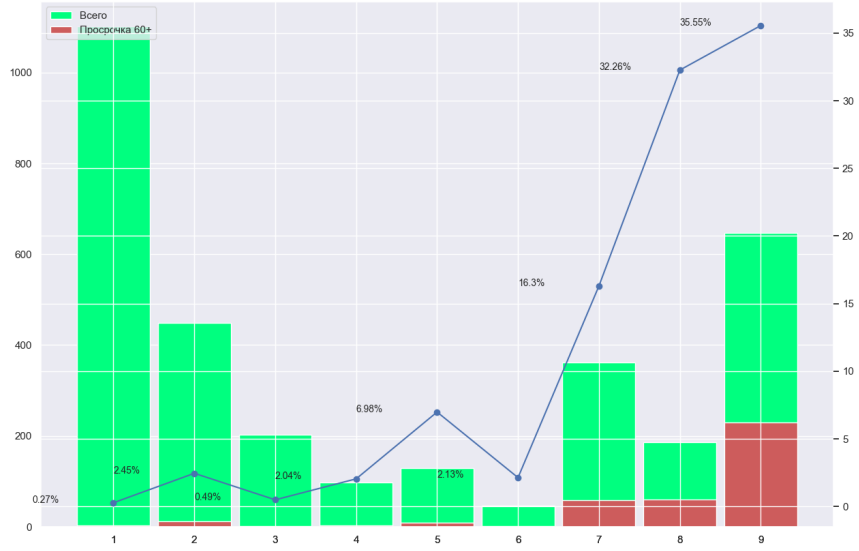
Распределение по децилям

Для итоговой модели на 16 компонент провозу сравнение распределение базы по децилям на выборке трейн и тест

Трейн



Тест



1. Модель работает не стабильно,судя по скачку проблемности в первом дециле трейн выборки

2. Модель имеет положительное смещение в сторону хорошего дециля (на выборке тест первый дециль не имеет столь высокой проблемности)
3. Распределение заявок на тестовой выборке не соответствует распределению на обучающей имея более сильный перекос в сторону первого дециля
4. Модель имеет тенденцию роста проблемности с ростом дециля и неплохо разделяет заявки
5. Очень большое отличие проблемностей в децилях по трейн и тест (при этом выборки разделялись случайно, а не по временному срезу, причина такой особенности не понятна)

Conclusions:

1. Скорость обучения падает с увеличением количества компонент
2. Еслибы бизнесово не было бы критичны потери в точности, то можно было бы получить точность буквально на 2 процентные позиции ниже со всего 2 компонентами и в пару раз быстрее скоростью обучения
3. На основании графика по explained variance было корректно выбрано количество компонент и без значимых потерь в точности в этой задаче можно уменьшить размерность с 19 до 16 компонент