

Факультет программной инженерии и компьютерной техники

Системы искусственного интеллекта

Лабораторная работа №5

Преподаватель: Болдырева Елена Александровна

Выполнил: Кульбако Артемий Юрьевич Р33112

# Задание

Цель: решить задачу многоклассовой классификации, используя в качестве тренировочного набора данных - набор данных MNIST, содержащий образы рукописных цифр.

1. Используйте метод главных компонент для набора данных MNIST (train dataset объемом 60000). Определите, какое минимальное количество главных компонент необходимо использовать, чтобы доля объясненной дисперсии превышала 0.80 + номер\_в\_списке % 10. Построить график зависимости доли объясненной дисперсии от количества используемых ГК.

2. Введите количество верно классифицированных объектов класса номер\_в\_списке%9 для тестовых данных.

3. Введите вероятность отнесения 5 любых изображений из тестового набора к назначенному классу.

4. Определите Accuracy, Precision, Recall или F1 для обученной модели.

5. Сделайте вывод про обученную модель.

# Выполнение

Немного дополним код, чтобы было легче подбирать количество компонент – напишем функцию, которая будет выводить, достигла ли доля объяснённой дисперсии необходимого значения:

from sklearn.decomposition import PCA

gold\_disp = 0.8 + 265570 % 10 # необходимая долю объясненной дисперсии

pca = PCA(n\_components=50, svd\_solver='full') # перебираем n\_components, пока один из элементов explained\_variance не станет больше gold\_disp

modelPCA = pca.fit(X\_train)

X\_train = modelPCA.transform(X\_train)

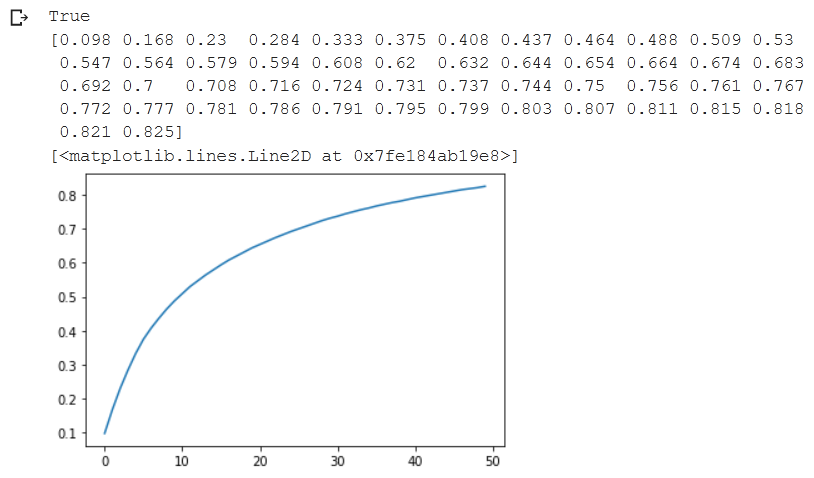
explained\_variance = np.round(np.cumsum(pca.explained\_variance\_ratio\_),3)

print(any(i >= gold\_disp for i in explained\_variance)) # функция, которая выведет True если Д.О.С. достигла нужного значения

print(explained\_variance)

plt.plot(np.arange(50), explained\_variance, ls = '-')

Минимально необходимое количество компонент = 50.



Определим число объектов, отнесённых к нужному классу:

var\_class = 265570 % 9

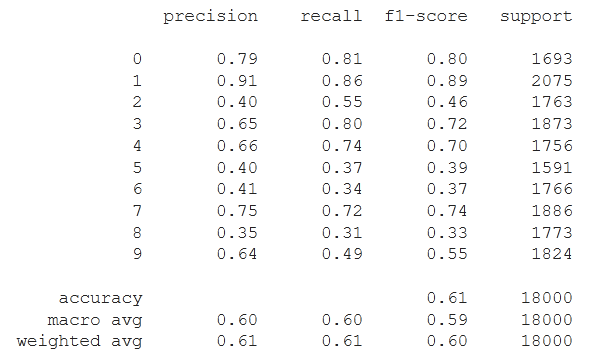
CM[var\_class][var\_class]

Результат = 1362

Выведем остальные параметры модели с помощью средств библиотеки sklearn:

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))



# Вывод

В лабораторной работе я реализовал многоклассовую классификацию с помощью методы опорных векторов для набора данных, состоящих из рукописных цифр, научился просматривать параметры обученной модели – это базовые навыки, необходимые для развития в сфере ML.