



## **Лабораторная работа №5**

*по дисциплине: Системы искусственного интеллекта*

Вариант: 15

Выполнил: Неграш Андрей, Р33301

Преподаватель: Кугаевских Александр Владимирович

Санкт-Петербург, 2022

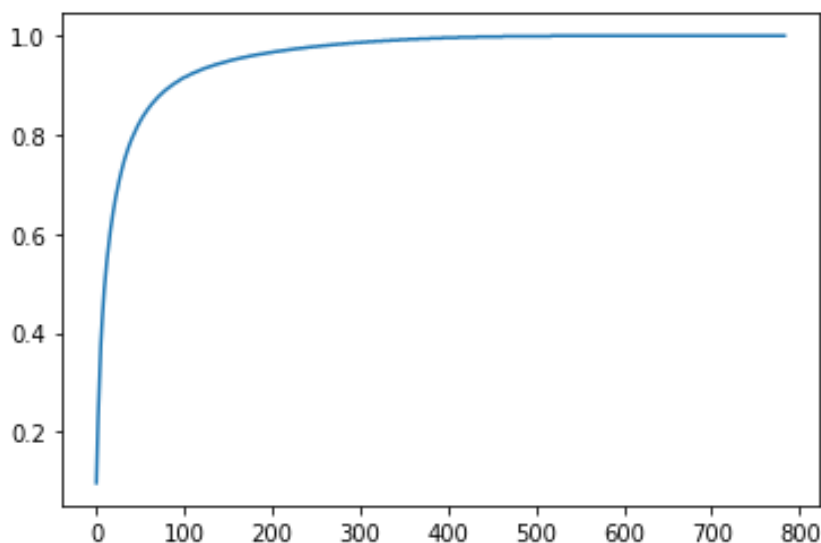
## Задание

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 объема 60000). Определите, какое минимальное количество главных компонент необходимо использовать, чтобы доля объясненной дисперсии превышала 0.80+номер\_в\_списке%10.
2. Построить график зависимости доли объясненной дисперсии от количества используемых ГК.
3. Введите количество верно классифицированных объектов класса номер\_в\_списке%9 для тестовых данных.
4. Введите вероятность отнесения 5 любых изображений из тестового набора к назначенному классу
5. Определите Accuracy, Precision, Recall or F1 для обученной модели
6. Сделайте вывод про обученную модель

## Выполнение

График зависимости доли объясненной дисперсии:



Матрица ошибок (confusion matrix):

```
[[1022  0  32  11  4  92  22  13  4  6]
 [  1 1258  16  27  8  2  14  8  16  3]
 [ 21  12 921  30  42  13  59  21  36  20]
 [ 30  7  58 916  15  81  15  16  89  8]
 [ 3  9  22  7 894  10  26  11  55 150]
 [123  6  59 193  30 463  45  27  79  53]
 [ 27  7  61  23  21  40 963  4  9  18]
 [ 21  23  19  2  30  11  8 1046  20  50]
 [ 21  7  71 116  6  52  11  7 836  44]
 [ 12  29  22  41 320  62  27 114  67 498]]
```

```

# количество объектов класса 2 отнесенных ко 2 классу
# second 2 -> count of predicted 2s
cl = 15 % 9
print(cl)
m_confusion[cl][cl]

```

---

```

6
963

from random import sample
indexes = sample(range(len(X_test)), k=5)
print(indexes)
test_objects = [X_test_transf[i] for i in indexes]
for number, proba in zip(indexes, clf.predict_proba(test_objects)):
print(f"Image #{number}:class={y_pred[number]},proba={round(max(proba), 4)}")

```

---

```

[5594, 5041, 7644, 6059, 2842]
Image #5594: class=5, proba=0.3143
Image #5041: class=9, proba=0.5371
Image #7644: class=6, proba=0.7081
Image #6059: class=9, proba=0.509
Image #2842: class=4, proba=0.3397

```

Accuracy, Precision, Recall or F1:

```

from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

```

---

	precision	recall	f1-score	support
0	0.80	0.85	0.82	1206
1	0.93	0.93	0.93	1353
2	0.72	0.78	0.75	1175
3	0.67	0.74	0.70	1235
4	0.65	0.75	0.70	1187
5	0.56	0.43	0.49	1078
6	0.81	0.82	0.82	1173
7	0.83	0.85	0.84	1230
8	0.69	0.71	0.70	1171
9	0.59	0.42	0.49	1192
accuracy			0.73	12000
macro avg	0.72	0.73	0.72	12000
weighted avg	0.73	0.73	0.73	12000

## Код

Код доступен на GitHub: [https://github.com/ANegrash/AI\\_systems\\_lab5](https://github.com/ANegrash/AI_systems_lab5)

## Вывод

В ходе данной лабораторной работы мне нужно было решить задачу многоклассовой классификации, используя в качестве тренировочного набора данных - набор данных MNIST, содержащий образы рукописных цифр.