Ушаков Владимир Александрович, группа МПиКИ

Лабораторная работа № 4

Знакомство с классификатором KNN

**Цель работы**

1. Решить задачу классификации исходного изображения с помощью метода k ближайших соседей (геометрические фигуры). Оценить точность полученной модели. Возможно использование преобученной нейронной сети.

**Код программы (внесённые изменения в шаблон кода выделены)**

import pandas as pd  
import pandas as pd  
import numpy as np  
import matplotlib.image as mplimg  
  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.model\_selection import train\_test\_split  
from sklearn.decomposition import PCA  
  
from PIL import Image  
#%%  
DATA\_PATH = 'data/figures.csv'  
RANDOM\_STATE = 42  
#%%  
def get\_img\_vect(path):  
 img = mplimg.imread(f'data/output/{path}')  
 return img[:, :, 0]  
  
def to\_gray(path):  
 img = Image.open(f'data/output/{path}').convert('LA')  
 img.save(f'data/gray/{path}')  
#%%  
df = pd.read\_csv(DATA\_PATH)  
#%%  
df['fig\_path'].apply(to\_gray)  
#%%  
df['vect\_fig'] = df['fig\_path'].apply(get\_img\_vect)  
df['target'] = df['fig\_name'].replace(['Circle', 'Triangle', 'Square'], [0, 1, 2])  
#%%  
pca = PCA(n\_components=3)  
pca.fit\_transform(df.loc[0, 'vect\_fig'])  
#%%  
pca.explained\_variance\_ratio\_  
#%%  
df['pca'] = df['vect\_fig'].apply(lambda x: PCA(3).fit\_transform(x))  
df.loc[0, 'pca']  
#%%  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(  
 df['pca'], df['target'], test\_size=0.2, random\_state=RANDOM\_STATE  
)  
X\_train = np.array([i.flatten() for i in X\_train])  
X\_test = np.array([i.flatten() for i in X\_test])  
#%%  
mdl = KNeighborsClassifier(3, weights='distance', leaf\_size=20)  
mdl.fit(X\_train, y\_train)  
mdl.score(X\_train, y\_train)  
#%%  
mdl.score(X\_test, y\_test)

**Результаты выполнения задания**

В результате выполнения работы была достигнута точность 81%

# Выводы

В результате работы была успешно обучена модель KNN и получена точность классификации 81%