МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

**Лабораторная работа №3**

по дисциплине

«Искусственный интеллект и машинное обучение»

**Выполнил:**

Сердюков Никита Анатольевич

Студент 2 курса группы \_ПИН-б-о-22-1

Направления подготовки

09.03.03 Прикладная информатика

очной формы обучения

Ставрополь, 2023 г.

Тема: МЕТРИЧЕСКИЕ МЕТОДЫ КЛАССИФИКАЦИИ.

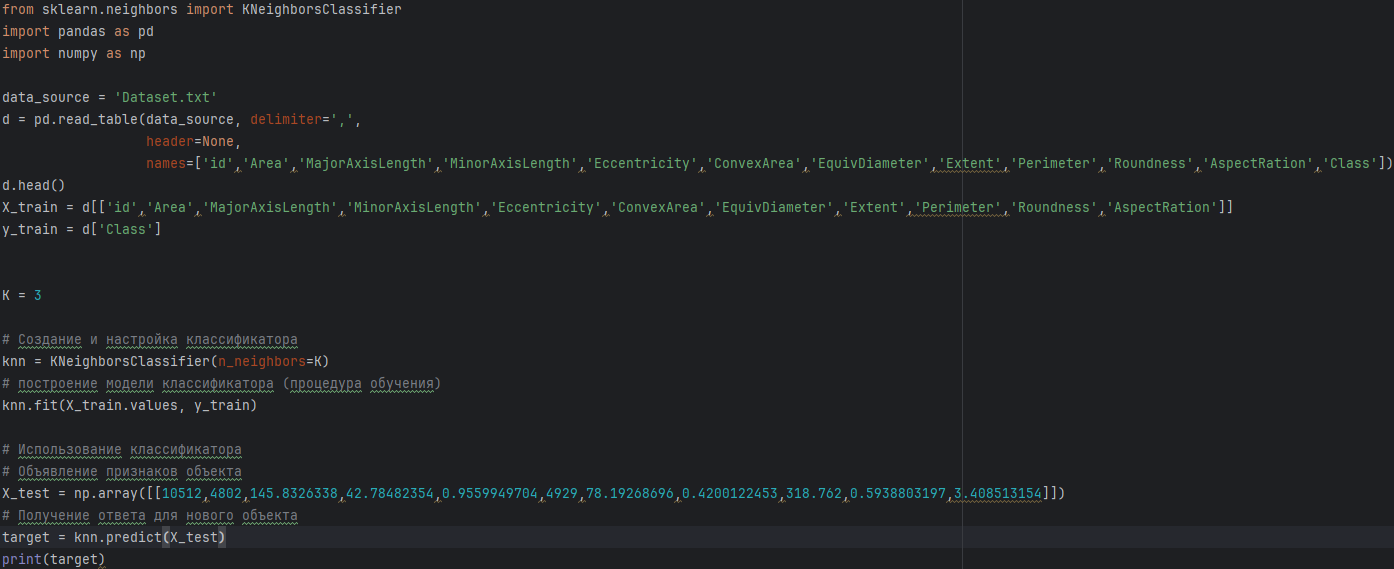
Цель работы: изучение принципов построении информационных систем с использованием метрических методов классификации.

Выполнение работы:

Датасет: Rice type classification

Описание датасета: Это набор данных, созданный для классификации риса.. Это модифицированный набор данных: Jasmine - 1, Gonen - 0.

1. Построение классификатора с заданием K (количества ближайших соседей) пользователем



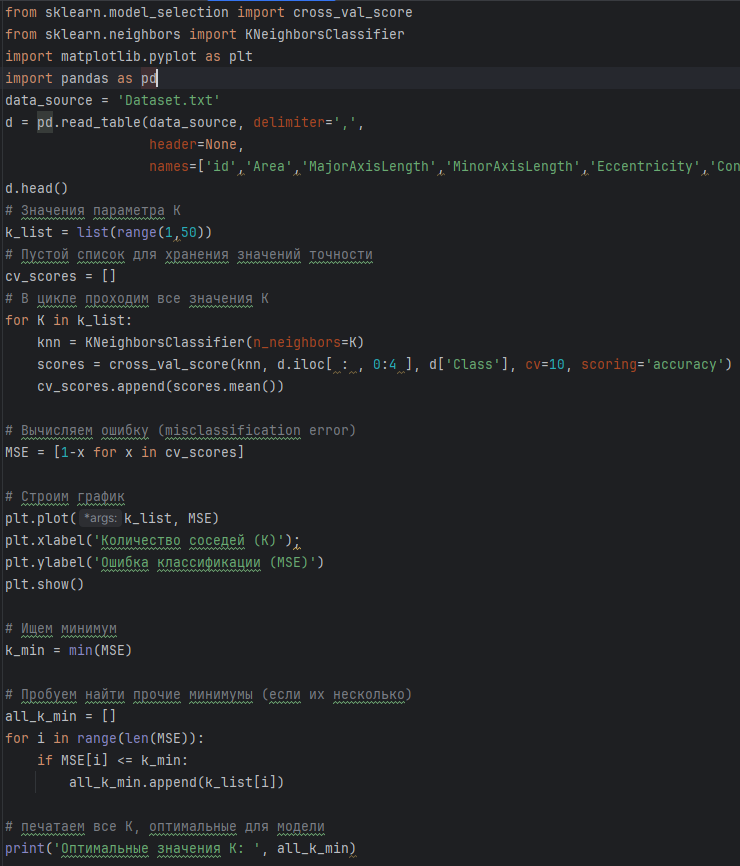


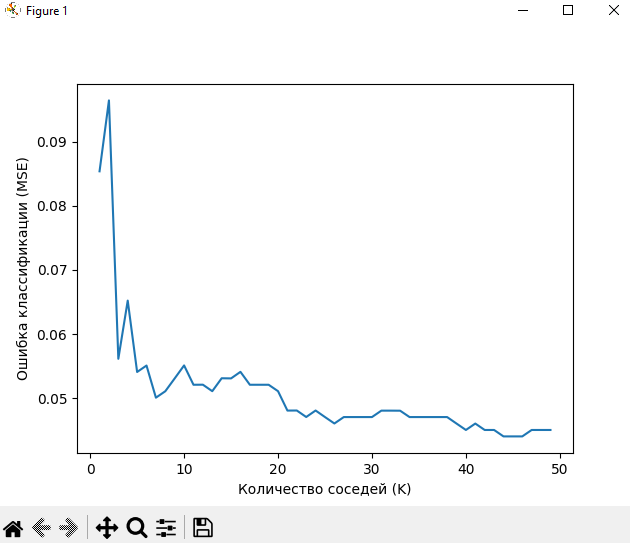
1. Вычисление оценки hold-out для различнх значений K, а также для различных долей обучающей и тестирующей подвыборок;





1. Применение метода hist() для визуализации распределения нескольких признаков





1. В моем наборе данных оптимальным число K является приближенным к максимальному количеству соседей

Листинг:

3.1:

from sklearn.neighbors import KNeighborsClassifier  
import pandas as pd  
import numpy as np  
  
data\_source = 'Dataset.txt'  
d = pd.read\_table(data\_source, delimiter=',',  
 header=None,  
 names=['id','Area','MajorAxisLength','MinorAxisLength','Eccentricity','ConvexArea','EquivDiameter','Extent','Perimeter','Roundness','AspectRation','Class'])  
d.head()  
X\_train = d[['id','Area','MajorAxisLength','MinorAxisLength','Eccentricity','ConvexArea','EquivDiameter','Extent','Perimeter','Roundness','AspectRation']]  
y\_train = d['Class']  
  
  
K = 3  
  
# Создание и настройка классификатора  
knn = KNeighborsClassifier(n\_neighbors=K)  
# построение модели классификатора (процедура обучения)  
knn.fit(X\_train.values, y\_train)  
  
# Использование классификатора  
# Объявление признаков объекта  
X\_test = np.array([[10512,4802,145.8326338,42.78482354,0.9559949704,4929,78.19268696,0.4200122453,318.762,0.5938803197,3.408513154]])  
# Получение ответа для нового объекта  
target = knn.predict(X\_test)  
print(target)

3.2:

import pandas as pd  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import accuracy\_score  
from sklearn.neighbors import KNeighborsClassifier  
  
data\_source = 'Dataset.txt'  
d = pd.read\_table(data\_source, delimiter=',',  
 header=None,  
 names=['id','Area','MajorAxisLength','MinorAxisLength',  
 'Eccentricity','ConvexArea','EquivDiameter','Extent','Perimeter','Roundness','AspectRation','Class'])  
d.head()  
  
  
X\_train, X\_holdout, y\_train, y\_holdout = train\_test\_split(  
 d.iloc[ :, 0:4 ],  
 d['Class'],  
 test\_size=0.3,  
 random\_state=17)  
knn = KNeighborsClassifier(n\_neighbors=3)  
knn.fit(X\_train, y\_train)  
knn\_pred = knn.predict(X\_holdout)  
accur = accuracy\_score(y\_holdout, knn\_pred)  
print('accuracy: ', accur)

3.3:

from sklearn.model\_selection import cross\_val\_score  
from sklearn.neighbors import KNeighborsClassifier  
import matplotlib.pyplot as plt  
import pandas as pd  
data\_source = 'Dataset.txt'  
d = pd.read\_table(data\_source, delimiter=',',  
 header=None,  
 names=['id','Area','MajorAxisLength','MinorAxisLength','Eccentricity','ConvexArea','EquivDiameter','Extent','Perimeter','Roundness','AspectRation','Class'])  
d.head()  
# Значения параметра K  
k\_list = list(range(1,50))  
# Пустой список для хранения значений точности  
cv\_scores = []  
# В цикле проходим все значения K  
for K in k\_list:  
 knn = KNeighborsClassifier(n\_neighbors=K)  
 scores = cross\_val\_score(knn, d.iloc[ : , 0:4 ], d['Class'], cv=10, scoring='accuracy')  
 cv\_scores.append(scores.mean())  
  
# Вычисляем ошибку (misclassification error)  
MSE = [1-x for x in cv\_scores]  
  
# Строим график  
plt.plot(k\_list, MSE)  
plt.xlabel('Количество соседей (K)');  
plt.ylabel('Ошибка классификации (MSE)')  
plt.show()  
  
# Ищем минимум  
k\_min = min(MSE)  
  
# Пробуем найти прочие минимумы (если их несколько)  
all\_k\_min = []  
for i in range(len(MSE)):  
 if MSE[i] <= k\_min:  
 all\_k\_min.append(k\_list[i])  
  
# печатаем все K, оптимальные для модели  
print('Оптимальные значения K: ', all\_k\_min)

Контрольные вопросы:

1. Метод ближайшего соседа относит объект к классу его ближайшего соседа. Метод k ближайших соседей относит объект к классу, к которому принадлежит большинство из k его ближайших соседей.

2. Основные принципы метода kNN: хранение обучающего набора данных, вычисление расстояний между объектами, выбор числа ближайших соседей k, определение класса целевого объекта на основе классов его ближайших соседей.

3. Количество соседей в методе kNN выбирается эмпирически или с помощью кросс-валидации. Маленькое k может привести к переобучению, а слишком большое k может уменьшить точность классификации.

4. Метод парзеновского окна использует окно определенной ширины вокруг целевого объекта для учета его ближайших соседей при классификации.

5. Принцип метода потенциальных функций заключается в том, что объекты разных классов представляются как потенциалы, а решение о принадлежности объекта к классу принимается на основе потенциалов и расстояний до объектов.

6. В методе kNN параметры, которые можно оптимизировать, включают выбор метрики расстояния, количество ближайших соседей k и веса соседей (если используется).

Вывод: изучил принципы построения информационных систем с использованием метрических методов классификации.