РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

Дисциплина: Информационный анализ данных

Студент: Ильин Никита

Группа: НФИбд-01-19

Москва 2022

Вариант №16

Датасет - winequality-red.csv

Задание

1. Используя функционал библиотеки Pandas, считайте заданный набор данных из репозитария UCI. Набор данных задан ссылкой на страницу набора данных и названием файла с данными, который доступен из папки с данными (data folder).
2. Проведите исследование набора данных, выявляя числовые признаки. Если какие-то из числовых признаков были неправильно классифицированы, то преобразуйте их в числовые. Если в наборе для числовых признаков присутствуют пропущенные значения ('?'), то заполните их медианными значениями признаков.
3. Определите столбец, содержащий метку класса (отклик). Если столбец, содержащий метку класса (отклик), принимает более 10 различных значений, то выполните дискретизацию этого столбца, перейдя к 4-5 диапазонам значений.
4. При помощи класса `SelectKBest` библиотеки scikit-learn найдите в наборе два признака, имеющих наиболее выраженную взаимосвязь с (дискретизированным) столбцом с меткой класса (откликом). Используйте для параметра `score_func` значения `chi2` или `f_classif`.
5. Для найденных признаков и (дискретизированного) столбца с меткой класса (откликом) вычислите матрицу корреляций и визуализируйте ее в виде тепловой карты (heat map).
6. Визуализируйте набор данных в виде диаграммы рассеяния на плоскости с координатами, соответствующими найденным признакам, отображая точки различных классов разными цветами. Подпишите оси и рисунок, создайте легенду набора данных.
7. Оставляя в наборе данных только числовые признаки, найдите и выведите на экран размерность метода главных компонент (параметр `n_components`), для которой доля объясняемой дисперсии будет не менее 97.5%.

8. Пользуясь методом главных компонент (PCA), снизьте размерность набора данных до двух признаков и изобразите полученный набор данных в виде диаграммы рассеяния на плоскости, образованной двумя полученными признаками, отображая точки различных классов разными цветами. Подпишите оси и рисунок, создайте легенду набора данных.

1. Используя функционал библиотеки Pandas, считайте заданный набор данных из репозитария UCI. Набор данных задан ссылкой на страницу набора данных и названием файла с данными, который доступен из папки с данными (data folder).

Ввод [6]:
```
url = \
    "https://archive.ics.uci.edu/ml/"+\
    "machine-learning-databases/wine-quality/winequality-red.csv"
```

Ввод [7]:
```
import pandas as pd
import numpy as np

# считываем данные в объект DataFrame
my_data = pd.read_csv(url, sep=";")
my_data
```

Out[7]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

2. Проведите исследование набора данных, выявляя числовые признаки. Если какие-то из числовых признаков были неправильно классифицированы, то преобразуйте их в числовые. Если в наборе для числовых признаков присутствуют пропущенные значения ('?'), то заполните их медианными значениями признаков.

Ввод [8]:
```
my_data = my_data.replace('?', np.NaN)
```

Ввод [9]:
```
print('Число записей = %d' % (my_data.shape[0]))
print('Число признаков = %d' % (my_data.shape[1]))

print('Число пропущенных значений:')
for col in my_data.columns:
    print('\t%s: %d' % (col,my_data[col].isna().sum()))
```

```
Число записей = 1599
Число признаков = 12
Число пропущенных значений:
        fixed acidity: 0
        volatile acidity: 0
        citric acid: 0
        residual sugar: 0
        chlorides: 0
        free sulfur dioxide: 0
        total sulfur dioxide: 0
        density: 0
        pH: 0
        sulphates: 0
        alcohol: 0
        quality: 0
```

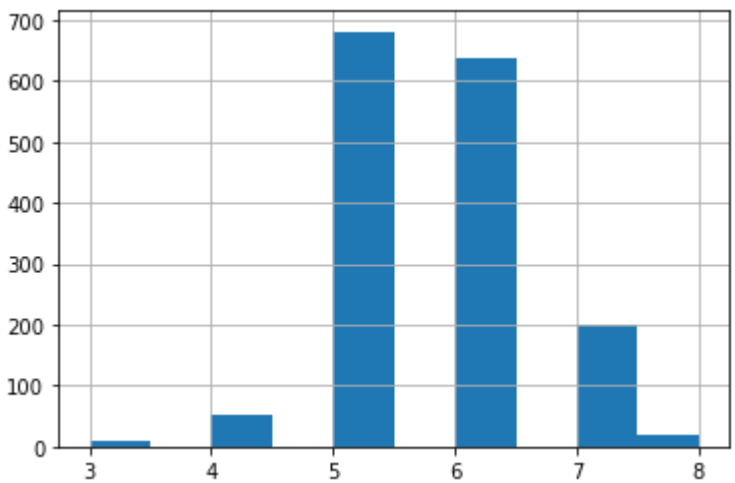
Пропущенных значений нет

3. Определите столбец, содержащий метку класса (отклик). Если столбец, содержащий метку класса (отклик), принимает более 10 различных значений, то выполните дискретизацию этого столбца, перейдя к 4-5 диапазонам значений.

так как quality единственый класс с числовыми значениями int - он и является меткой класса.

Ввод [10]:
```
my_data['quality'].hist(bins=10)
my_data['quality'].value_counts(sort=False)
```

Out[10]:
```
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
```



Дискретизация не требуется, т.к. столбец принимает только 6 значений

4. При помощи класса `SelectKBest` библиотеки scikit-learn найдите в наборе два признака, имеющих наиболее выраженную взаимосвязь с (дискретизированным) столбцом с меткой класса (откликом). Используйте для параметра `score_func` значения `chi2` или `f_classif`.

Ввод [11]:
```
# отбор признаков при помощи одномерных статистических тестов
from sklearn.feature_selection import SelectKBest,chi2

print("\nИсходный набор данных:\n",my_data.head())
array = my_data.values
X = array[:,0:11] # входные переменные (11 признаков)
Y = array[:,11]   # выходная переменная - качество (оценка между 0 и 10)

# отбор признаков
test = SelectKBest(score_func=chi2, k=2)
fit = test.fit(X, Y)

# оценки признаков
print("\nОценки признаков:\n",fit.scores_)

cols = test.get_support(indices=True)
my_data_new = my_data.iloc[:,cols]
print("\nОтобранные признаки:\n",my_data_new.head())
```

```
Исходный набор данных:
    fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5

Оценки признаков:
 [1.12606524e+01 1.55802891e+01 1.30256651e+01 4.12329474e+00
 7.52425579e-01 1.61936036e+02 2.75555798e+03 2.30432045e-04
 1.54654736e-01 4.55848775e+00 4.64298922e+01]

Отобранные признаки:
    free sulfur dioxide  total sulfur dioxide
0                 11.0                  34.0
1                 25.0                  67.0
2                 15.0                  54.0
3                 17.0                  60.0
4                 11.0                  34.0
```

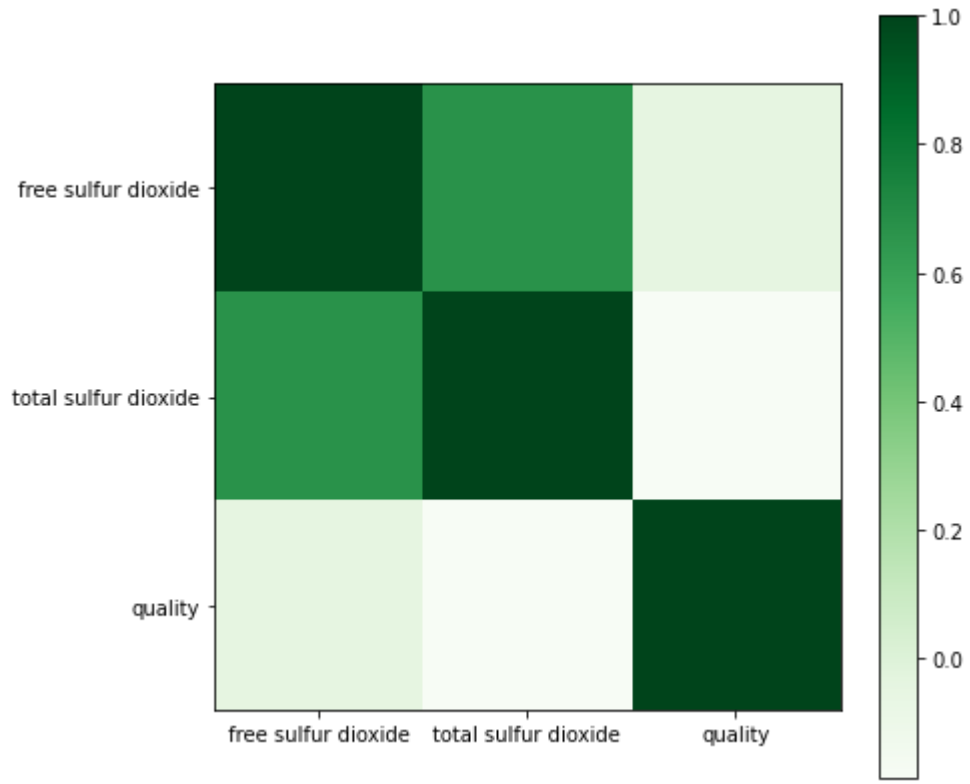
5. Для найденных признаков и (дискретизированного) столбца с меткой класса (откликом) вычислите матрицу корреляций и визуализируйте ее в виде тепловой карты (heat map).

Ввод [12]:
```
# важность признаков с классификатором Extra Trees
from sklearn.ensemble import ExtraTreesClassifier

array = my_data.values
X = array[:,0:11] # входные переменные (11 признаков)
Y = array[:,11]   # выходная переменная - качество (оценка между 0 и 10)

# отбор признаков
model = ExtraTreesClassifier()
model.fit(X, Y)
print(model.feature_importances_)
```

```
[0.07660584 0.09875045 0.08227485 0.07910332 0.07527924 0.07348791
 0.10091277 0.08489166 0.07512999 0.10294128 0.15062269]
```

total sulfur dioxide и free sulfur dioxide - наиболее важные признаки quality - метка класса

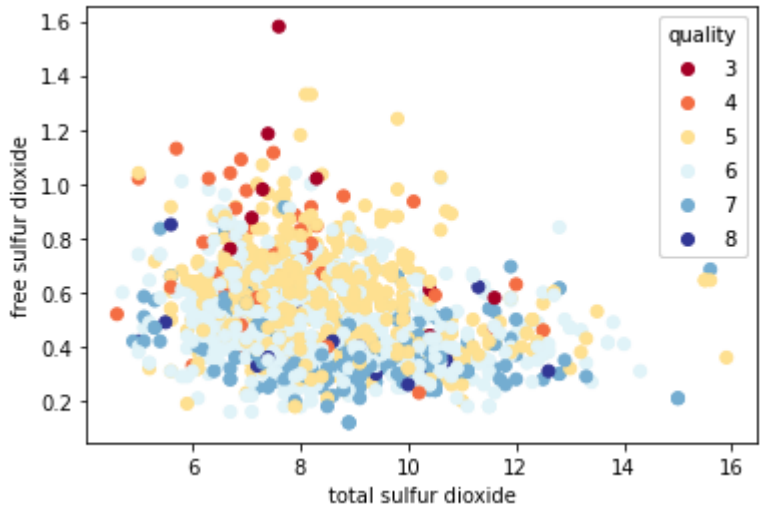
Ввод [13]:
```
import matplotlib.pyplot as plt

corr_matrix = my_data[['free sulfur dioxide','total sulfur dioxide','quality']].c
corr_matrix

plt.figure(figsize=(7, 7))
plt.imshow(corr_matrix, cmap='Greens')
plt.colorbar()  # добавим шкалу интенсивности цвета

plt.xticks(range(len(corr_matrix.columns)), corr_matrix.columns)
plt.yticks(range(len(corr_matrix)), corr_matrix.index);
```



6. Визуализируйте набор данных в виде диаграммы рассеяния на плоскости с координатами, соответствующими найденным признакам, отображая точки различных классов разными цветами. Подпишите оси и рисунок, создайте легенду набора данных.

Ввод [14]:
```
X = np.array(my_data.drop('quality', axis=1))
y = np.array(my_data["quality"])
fig, ax = plt.subplots()
s = ax.scatter(X[:,0], X[:,1], c = y, cmap = plt.cm.RdYlBu)
leg = ax.legend(*s.legend_elements(), title='quality')
ax.add_artist(leg)
plt.xlabel('total sulfur dioxide')
plt.ylabel('free sulfur dioxide')
```

Out[14]:
```
Text(0, 0.5, 'free sulfur dioxide')
```



7. Оставляя в наборе данных только числовые признаки, найдите и выведите на экран размерность метода главных компонент (параметр `n_components`), для которой доля объясняемой дисперсии будет не менее 97.5%.

Ввод [15]:
```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)

pcad = pca.fit_transform(my_data) # numpy array

print( "*** Первые 5 строк данных:" )
for x in range(0,5):
  print( pcad[x] )

print( "*** Дисперсии компонент:\n", pca.explained_variance_ratio_ )
```

```
*** Первые 5 строк данных:
[-13.22202658  -2.03192212]
[22.04025471  4.40179054]
[ 7.16536169 -2.50832073]
[13.42836949 -1.94603248]
[-13.22202658  -2.03192212]
*** Дисперсии компонент:
 [0.94607951 0.04834835]
```

Ввод [16]:
```
for r in range(1,5):
  pca = PCA( n_components = r )
  pca.fit( my_data )
  print( "r =",r,"\tДисперсия =",
        sum(pca.explained_variance_ratio_)*100,"%" )
```

```
r = 1    Дисперсия = 94.60795135347404 %
r = 2    Дисперсия = 99.44278609084158 %
r = 3    Дисперсия = 99.70238517736415 %
r = 4    Дисперсия = 99.85467385482455 %
```

при r = 2 дисперсия подходит под условие

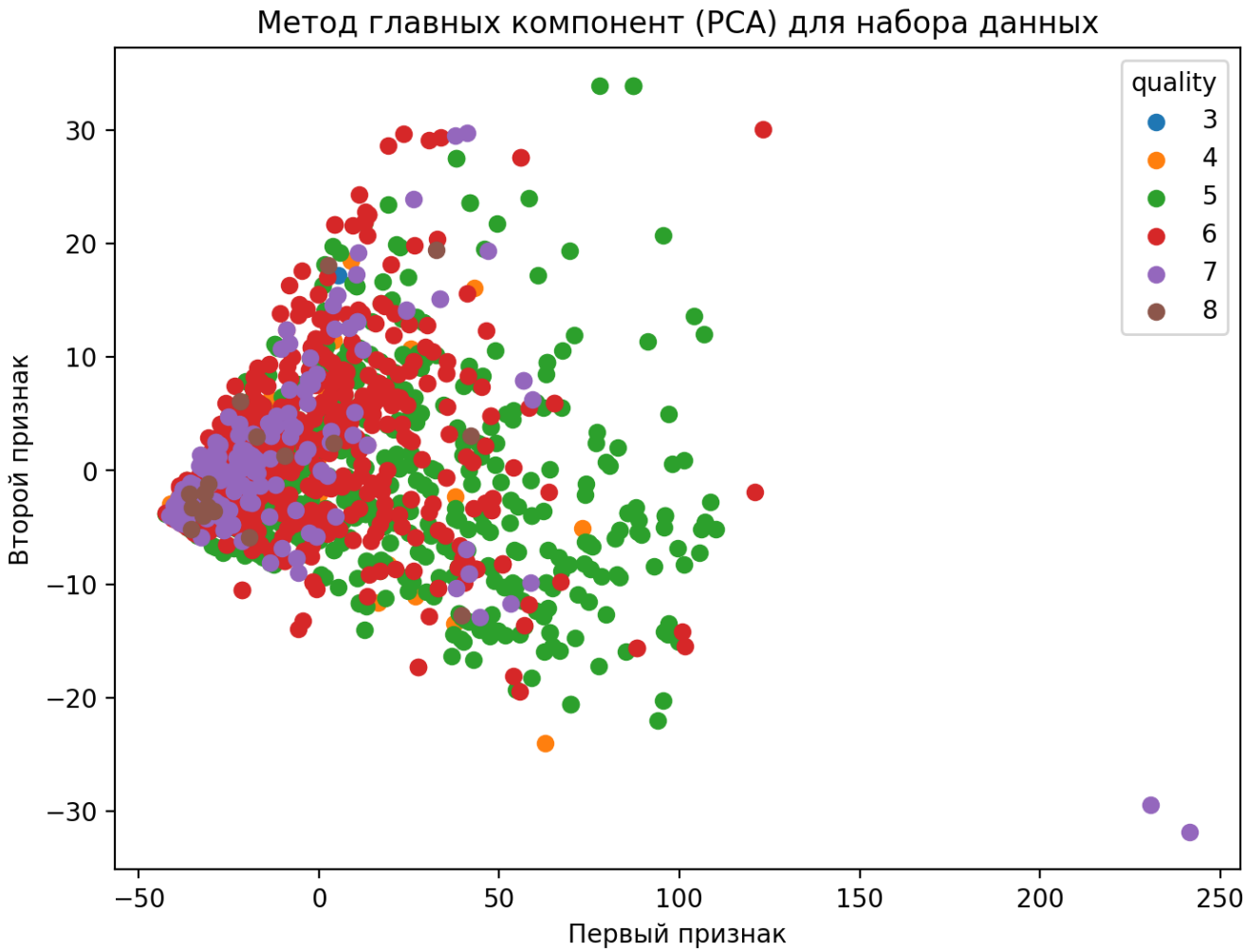
8. Пользуясь методом главных компонент (PCA), снизьте размерность набора данных до двух признаков и изобразите полученный набор данных в виде диаграммы рассеяния на плоскости, образованной двумя полученными признаками, отображая точки различных классов разными цветами. Подпишите оси и рисунок, создайте легенду набора данных.

Ввод [21]:
```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)

pcad = pca.fit_transform(my_data) # numpy array

print( "*** Первые 5 строк данных:" )
for x in range(0,5):
  print( pcad[x] )

print( "*** Дисперсии компонент:\n", pca.explained_variance_ratio_ )
```

```
*** Первые 5 строк данных:
[-13.22202658  -2.03192212]
[22.04025471  4.40179054]
[ 7.16536169 -2.50832073]
[13.42836949 -1.94603248]
[-13.22202658  -2.03192212]
*** Дисперсии компонент:
 [0.94607951 0.04834835]
```

Ввод [32]:
```
target = np.array(my_data['quality'])

plt.figure( figsize=(8, 6), dpi=200 )

plt.scatter(pcad[target==3,0],
         pcad[target==3,1], label = 3)
plt.scatter(pcad[target==4,0],
         pcad[target==4,1], label = 4)
plt.scatter(pcad[target==5,0],
         pcad[target==5,1], label = 5)
plt.scatter(pcad[target==6,0],
         pcad[target==6,1], label = 6)
plt.scatter(pcad[target==7,0],
         pcad[target==7,1], label = 7)
plt.scatter(pcad[target==8,0],
         pcad[target==8,1], label = 8)

plt.title("Метод главных компонент (PCA) для набора данных")
plt.xlabel("Первый признак")
plt.ylabel("Второй признак")
plt.legend(title="quality")
```

Out[32]:
```
<matplotlib.legend.Legend at 0x146f22ee0>
```



Ввод [ ]: