Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №1
по дисциплине
«Методы машинного обучения»
на тему
«Разведочный анализ данных. Исследование и визуализация данных»

Выполнил:
студент группы ИУ5-23М
Наседкин И. А.

_____

Москва — 2020 г.

Описание задания

Цель лабораторной работы: изучение различных методов визуализация данных.

Краткое описание. Построение основных графиков, входящих в этап разведочного анализа данных. Корреляционный анализ данных. Формирование выводов о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

Задание: • Выбрать набор данных (датасет).

• Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных, например из Scikit-learn. Для лабораторных работ не рекомендуется выбирать датасеты большого размера.

• Создать ноутбук, который содержит следующие разделы: 1. Текстовое описание выбранного Вами набора данных. 2. Основные характеристики датасета. 3. Визуальное исследование датасета. Необходимо использовать не менее 2 различных библиотек и не менее 5 графиков. 4. Информация о корреляции признаков.

• Сформировать отчет и разместить его в своем репозитории на github.

1. Ход выполнения

2. 1) Текстовое описание набора данных

В качестве набора данных мы будем использовать набор данных, являющихся результатами химического анализа вин, выращиваемых в одном и том же регионе Италии тремя разными производителями. Есть 13 измерений различных составляющих вин трех разных типов. https://scikit-learn.org/stable/datasets/index.html

Датасет состоит из одного файла, который мы и будем использовать: wine.data

Есть следующие колонки:

Alcohol - процентное содержание алкоголя

Malic acid - содержание яблочной кислоты

Ash - щелочь

Alcalinity of ash - содержание щелочи

Magnesium - магний

Total phenols - общее число фенолов

Flavanoids - флавоноиды

Nonflavanoid phenols - нефлаваноидные фенолы

Proanthocyanins - проантоцианидины

Color intensity - интенсивность цвета

Hue - оттенок вина

OD280/OD315 of diluted wines - разбавленность вина

Proline - сорт вина

[ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
from sklearn.datasets import load_wine
data2 = load_wine()
def make_dataframe(ds_function):
    ds = ds_function()
```

```
    df = pd.DataFrame(data= np.c_[ds['data'], ds['target']],
                      columns= list(ds['feature_names']) + ['target'])
    return df
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in
  ↪the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

[ ]:
```
type(data2)
```

[ ]:
```
sklearn.utils.Bunch
```

[ ]:
```
# Датасет возвращается в виде словаря со следующими ключами
for x in data2:
    print(x)
```

```
data
target
target_names
DESCR
feature_names
```

[ ]:
```
data2['target_names']
```

[ ]:
```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

[ ]:
```
data2['feature_names']
```

[ ]:
```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

[ ]:
```
data2['data'].shape
```

[ ]:
```
(178, 13)
```

[ ]:
```
data2['target'].shape
```

[ ]: (178,)

[ ]:
```
data1 = make_dataframe(load_wine)
data1.head()
```

[ ]:
```
   alcohol  malic_acid   ash  …  od280/od315_of_diluted_wines  proline
  ↪target
0    14.23        1.71  2.43  …                          3.92   1065.0
  ↪    0.0
1    13.20        1.78  2.14  …                          3.40   1050.0
  ↪    0.0
2    13.16        2.36  2.67  …                          3.17   1185.0
  ↪    0.0
3    14.37        1.95  2.50  …                          3.45   1480.0
  ↪    0.0
4    13.24        2.59  2.87  …                          2.93    735.0
  ↪    0.0

[5 rows x 14 columns]
```

[ ]:
```
#data1 = pd.read_csv('wine (1).data', sep=",")
```

3. 2) Основные характеристики датасета

[ ]:
```
# Первые 5 строк датасета
data1.head()
```

[ ]:
```
   alcohol  malic_acid   ash  …  od280/od315_of_diluted_wines  proline
  ↪target
0    14.23        1.71  2.43  …                          3.92   1065.0
  ↪    0.0
1    13.20        1.78  2.14  …                          3.40   1050.0
  ↪    0.0
2    13.16        2.36  2.67  …                          3.17   1185.0
  ↪    0.0
3    14.37        1.95  2.50  …                          3.45   1480.0
  ↪    0.0
4    13.24        2.59  2.87  …                          2.93    735.0
  ↪    0.0

[5 rows x 14 columns]
```

[ ]:
```
#data.DESCR
```

[ ]:
```
#data.target
```

[ ]:
```
# Размер датасета - 8143 строк, 7 колонок
data1.shape
```

[ ]: (178, 14)

[ ]:
```
total_count = data1.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 178
```

[ ]:
```
# Список колонок
data1.columns
```

[ ]:
```
Index(['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',
       'total_phenols', 'flavanoids', 'nonflavanoid_phenols',
       'proanthocyanins', 'color_intensity', 'hue',
       'od280/od315_of_diluted_wines', 'proline', 'target'],
      dtype='object')
```

[ ]:
```
# Список колонок с типами данных
data1.dtypes
```

[ ]:
```
alcohol                         float64
malic_acid                      float64
ash                             float64
alcalinity_of_ash               float64
magnesium                       float64
total_phenols                   float64
flavanoids                      float64
nonflavanoid_phenols            float64
proanthocyanins                 float64
color_intensity                 float64
hue                             float64
od280/od315_of_diluted_wines    float64
proline                         float64
target                          float64
dtype: object
```

[ ]:
```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data1.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data1[data1[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
alcohol - 0
malic_acid - 0
ash - 0
alcalinity_of_ash - 0
magnesium - 0
total_phenols - 0
flavanoids - 0
nonflavanoid_phenols - 0
```

```
proanthocyanins - 0
color_intensity - 0
hue - 0
od280/od315_of_diluted_wines - 0
proline - 0
target - 0
```

[ ]:
```
# Основные статистические характеристки набора данных
data1.describe()
```

[ ]:

	alcohol	malic_acid	…	proline	target
count	178.000000	178.000000	…	178.000000	178.000000
mean	13.000618	2.336348	…	746.893258	0.938202
std	0.811827	1.117146	…	314.907474	0.775035
min	11.030000	0.740000	…	278.000000	0.000000
25%	12.362500	1.602500	…	500.500000	0.000000
50%	13.050000	1.865000	…	673.500000	1.000000
75%	13.677500	3.082500	…	985.000000	2.000000
max	14.830000	5.800000	…	1680.000000	2.000000

[8 rows x 14 columns]

[ ]:
```
# Определим уникальные значения для целевого признака
data1['target'].unique()
```

[ ]:
```
array([0., 1., 2.])
```

Целевой признак содержит 3 значения.

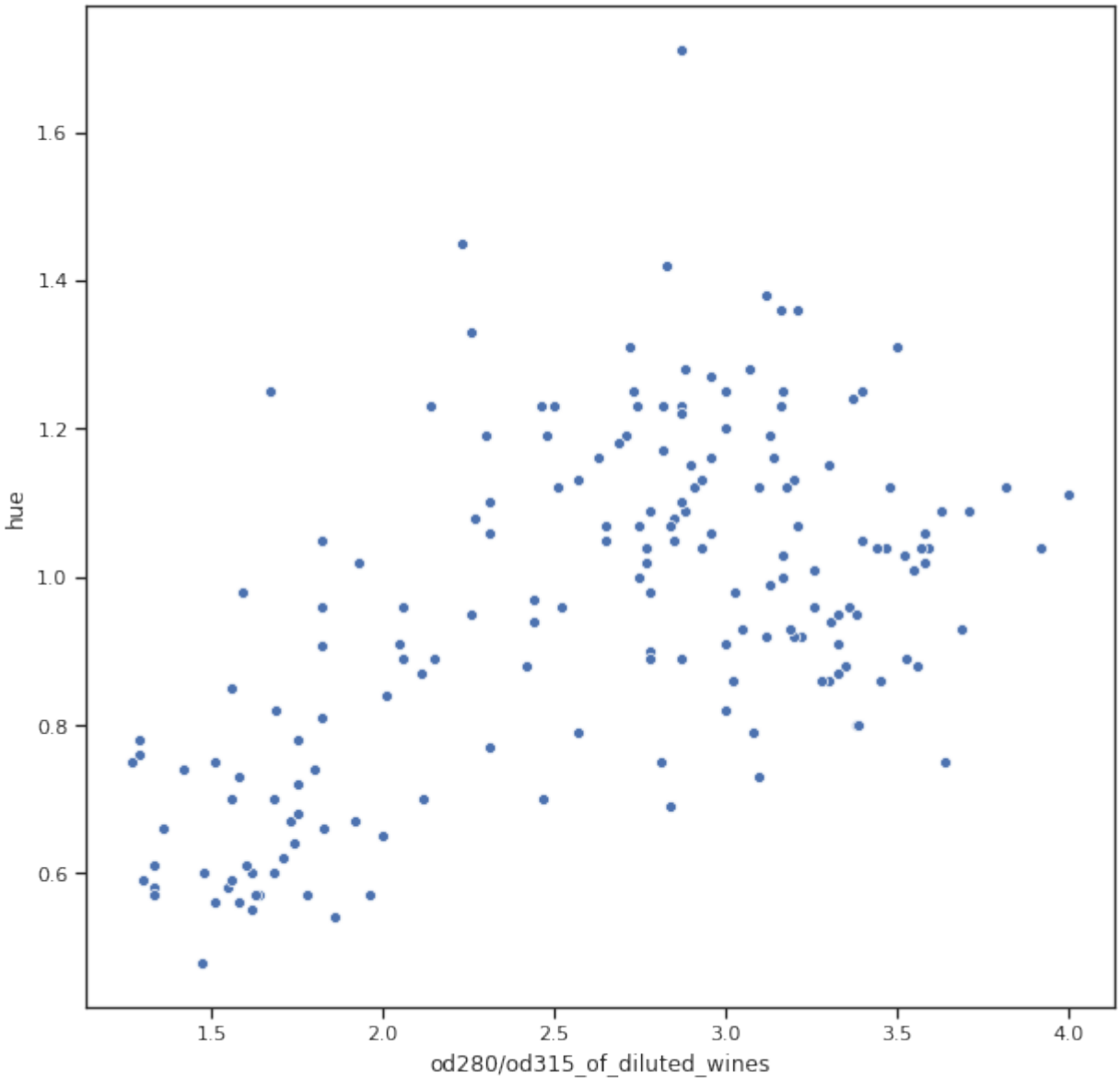
4. 3) Визуальное исследование датасета

4.1. Диаграмма рассеивания

Позволяет построить распределение двух колонок данных и визуально обнаружить наличие зависимости. Не предполагается, что значения упорядочены (например, по времени).

[ ]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='od280/od315_of_diluted_wines', y='hue',
↪data=data1)
```

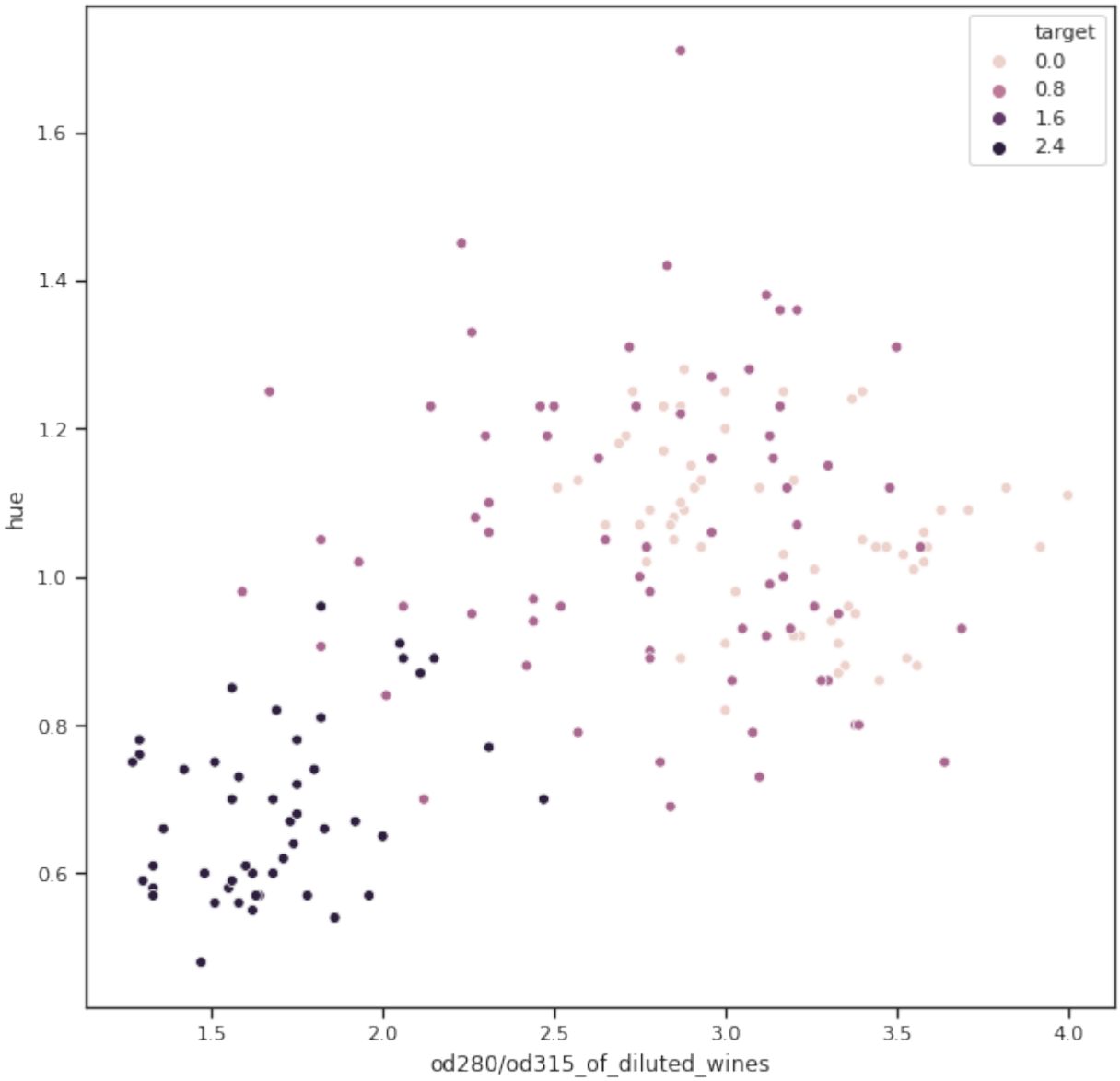
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9223269eb8>
```



Зависимость между разбавленностью и оттенком вина. Насколько на эту зависимость влияет целевой признак. Примечание: почти линейная зависимость имеется между флаваноидами и общим содержанием фенолов, что было замечено мною позднее на множестве графиков.

[ ]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='od280/od315_of_diluted_wines', y='hue',
↪data=data1, hue='target')
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f92502a7e48>
```

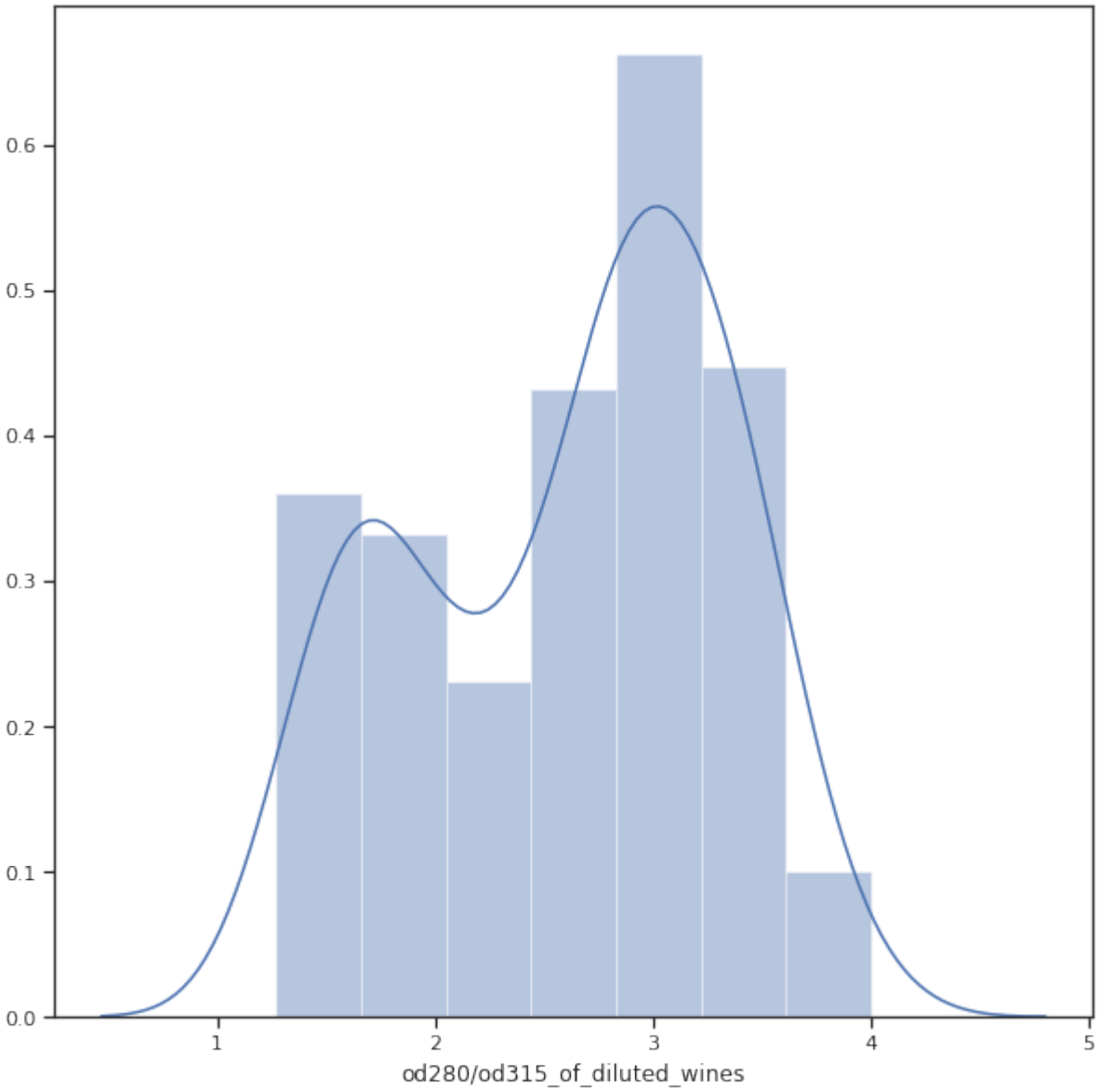


4.2. Гистограмма

Позволяет оценить плотность вероятности распределения данных.

[ ]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data1['od280/od315_of_diluted_wines'])
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f92204a4ac8>
```

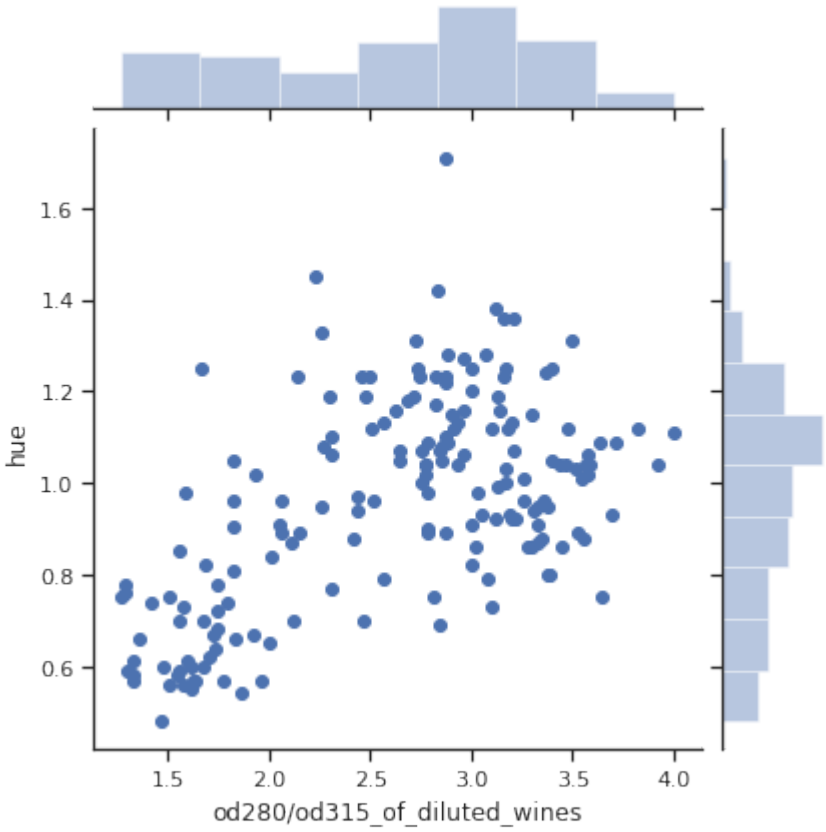


4.3. Jointplot

Комбинация гистограмм и диаграммы рассеивания

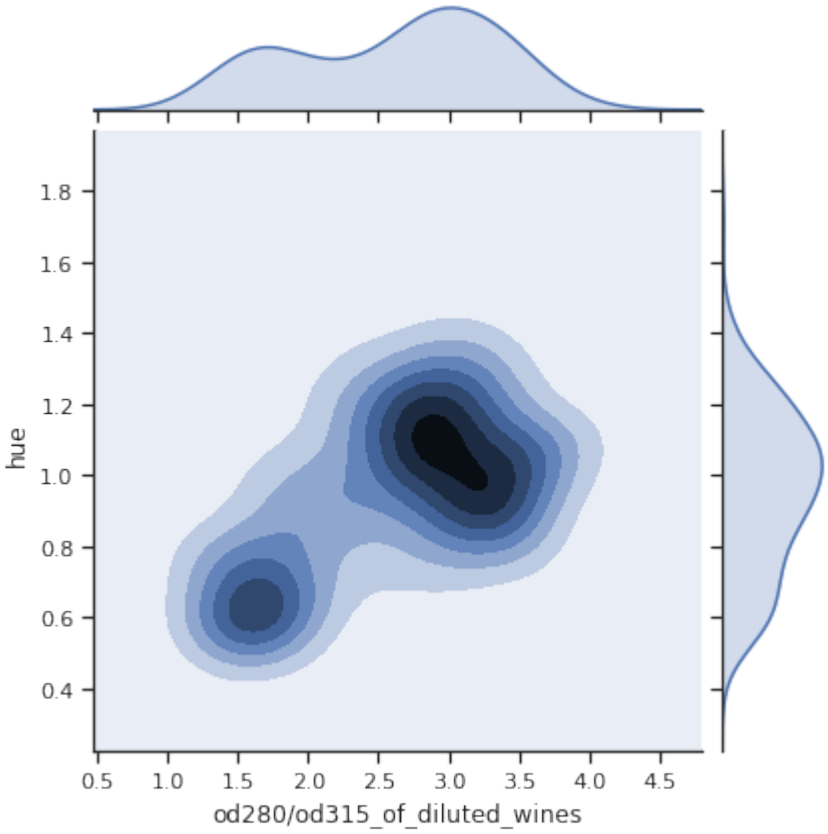
[ ]:
```
sns.jointplot(x='od280/od315_of_diluted_wines', y='hue', data=data1)
```

[ ]:
```
<seaborn.axisgrid.JointGrid at 0x7f922042a940>
```



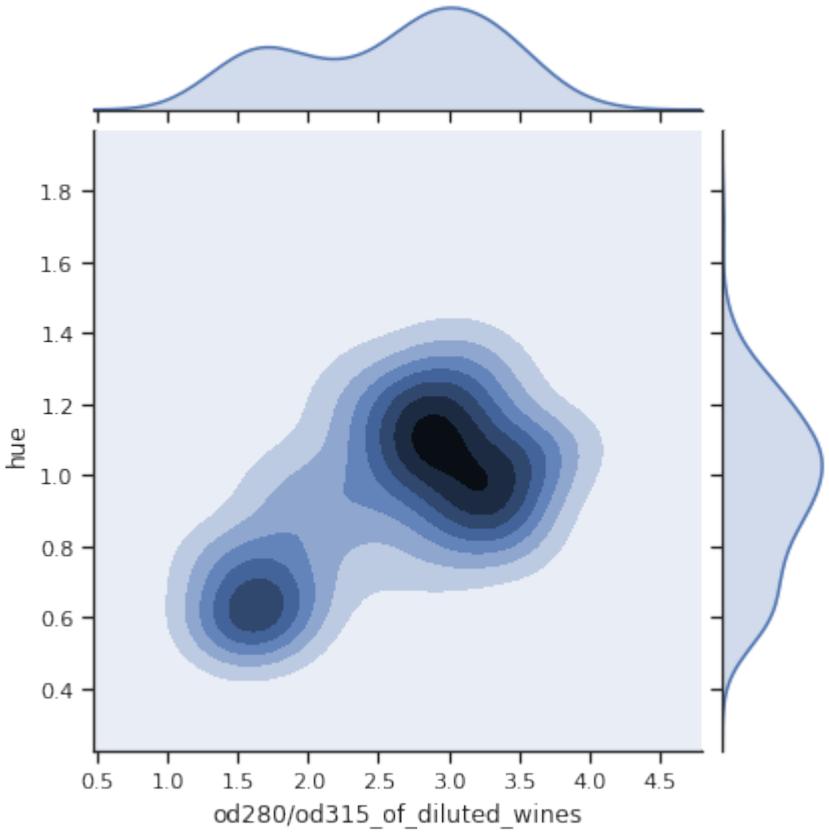
[ ]:
```
sns.jointplot(x='od280/od315_of_diluted_wines', y='hue', data=data1,
↪kind="kde")
```

[ ]:
```
<seaborn.axisgrid.JointGrid at 0x7f92202d3748>
```



[ ]:
```
sns.jointplot(x='od280/od315_of_diluted_wines', y='hue', data=data1,
↪kind="kde")
```

[ ]:
```
<seaborn.axisgrid.JointGrid at 0x7f921ff72a20>
```



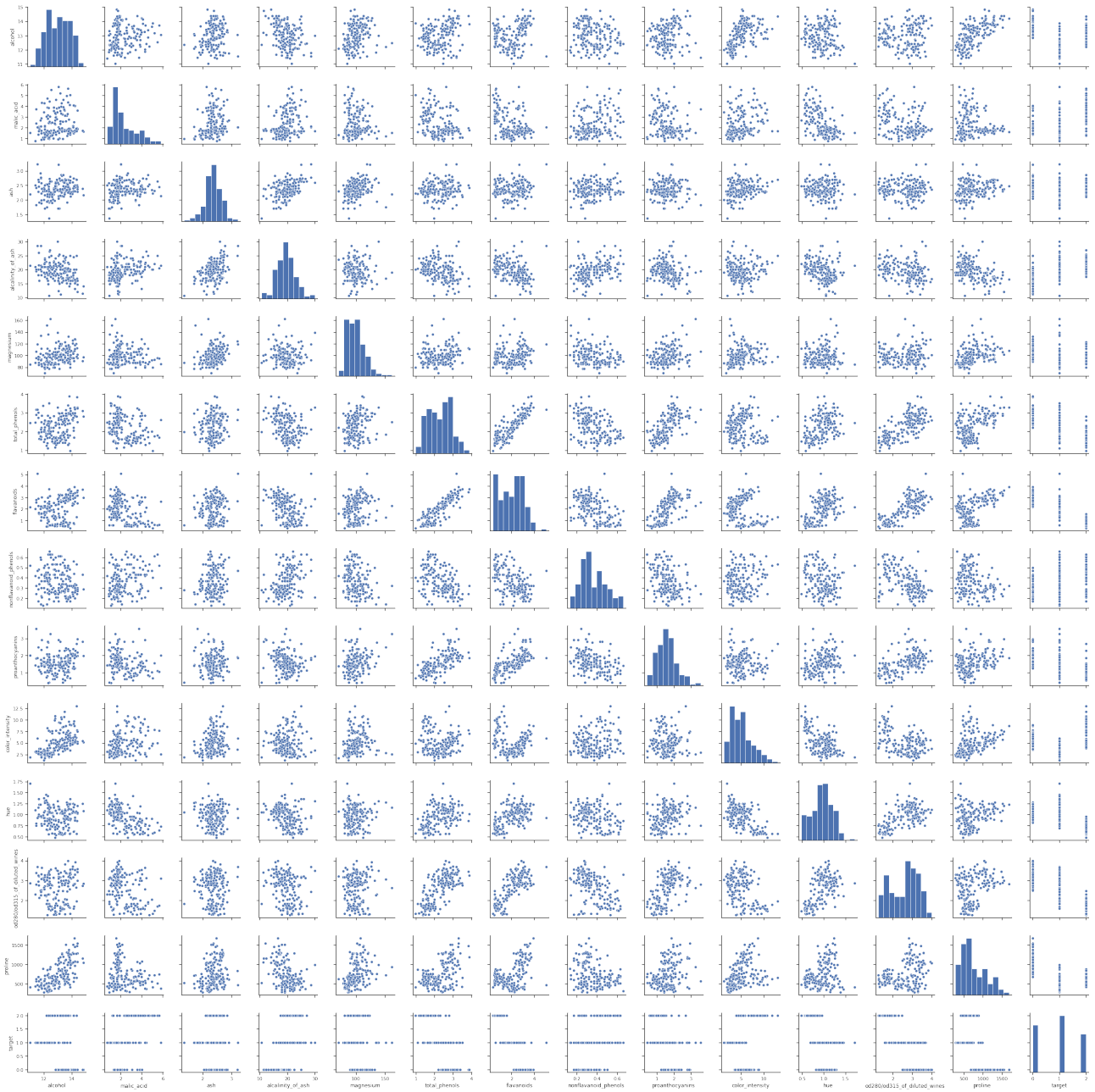
4.4. Парные диаграммы

Комбинация гистограмм и диаграмм рассеивания для всего набора данных.

Выводится матрица графиков. На пересечении строки и столбца, которые соответстуют двум показателям, строится диаграмма рассеивания. В главной диагонали матрицы строятся гистограммы распределения соответствующих показателей.

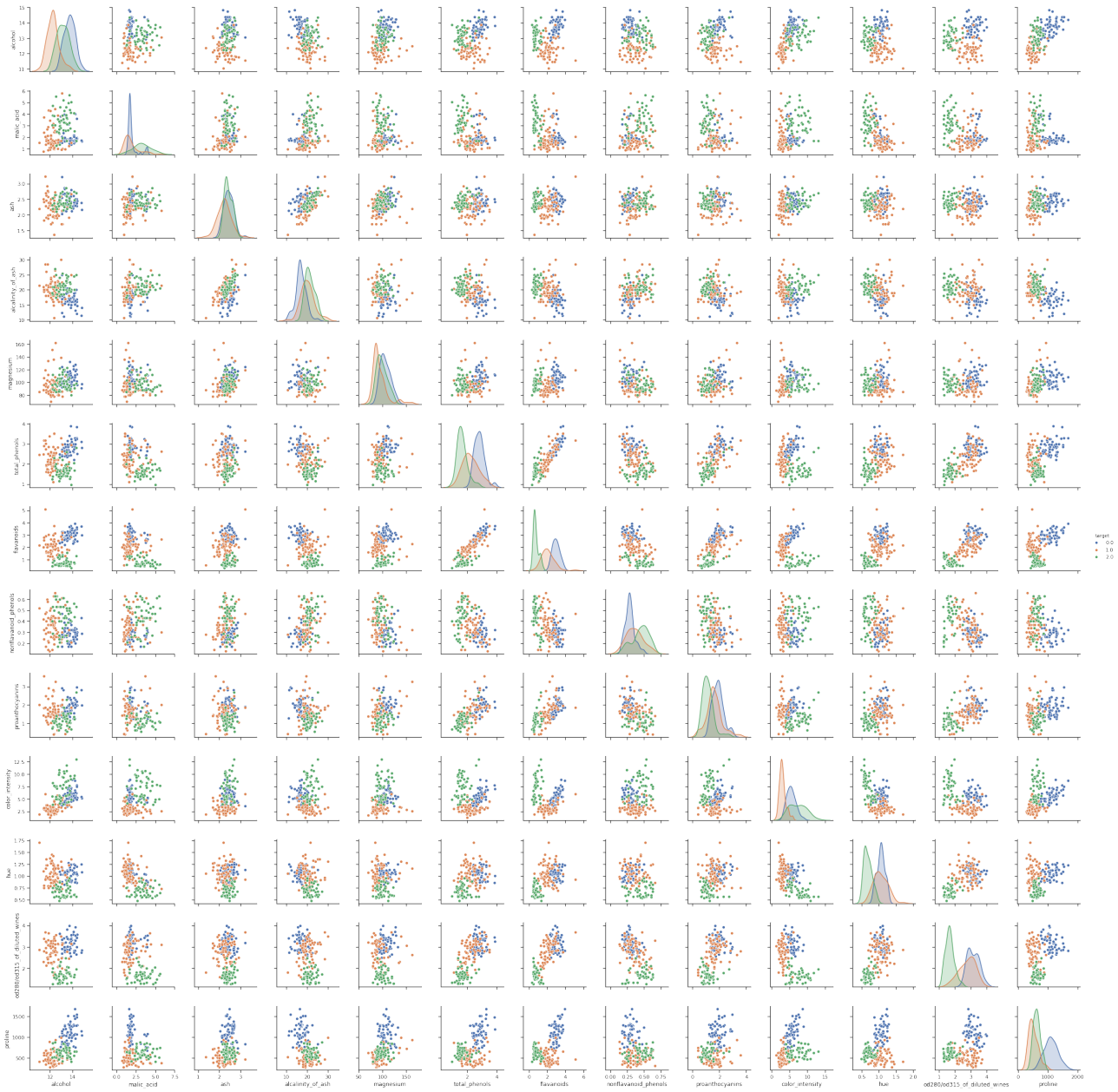
[ ]:
```
sns.pairplot(data1)
```

[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7f921fe29940>
```



[ ]:
```
sns.pairplot(data1, hue="target")
```

[ ]: <seaborn.axisgrid.PairGrid at 0x7f921bf3fbe0>

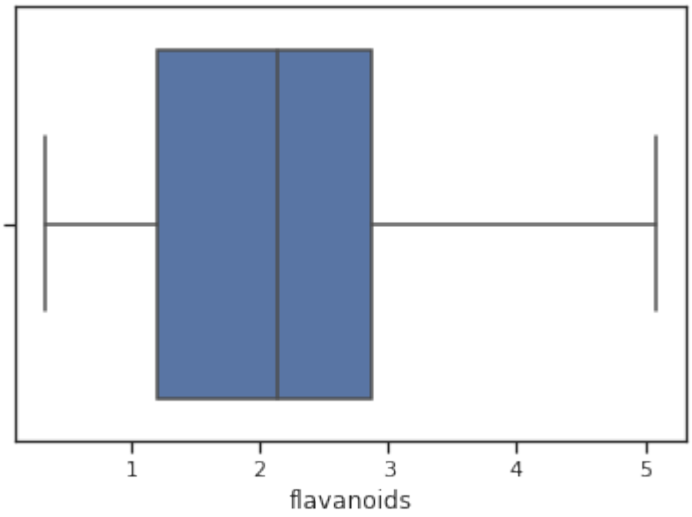


4.5. Ящик с усами

Отображает одномерное распределение вероятности.

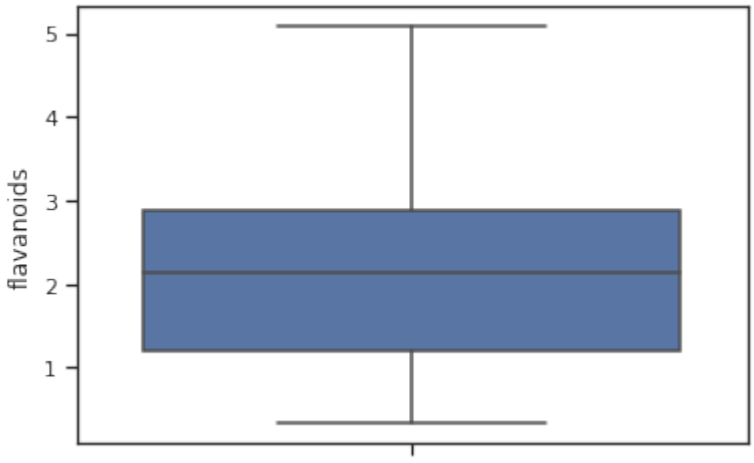
[ ]:
```
sns.boxplot(x=data1['flavanoids'])
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9217984cc0>
```



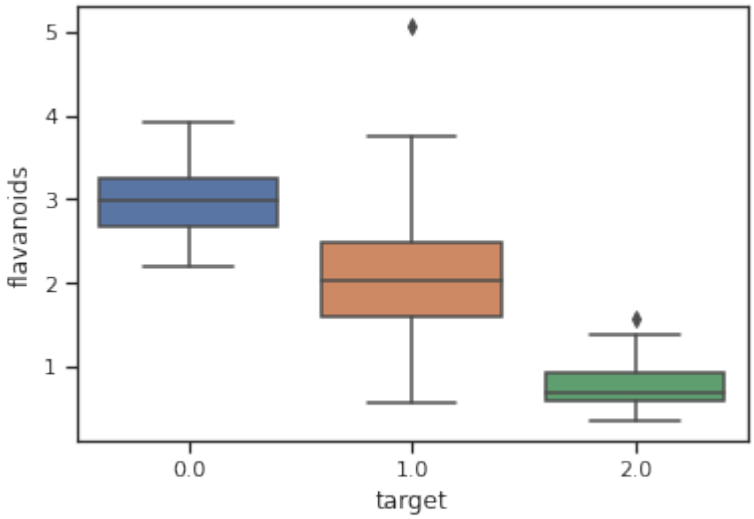
[ ]:
```
# По вертикали
sns.boxplot(y=data1['flavanoids'])
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9216bed358>
```



[ ]:
```
# Распределение параметра Flavanoids сгруппированные по Class.
sns.boxplot(x='target', y='flavanoids', data=data1)
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9216fa4e10>
```

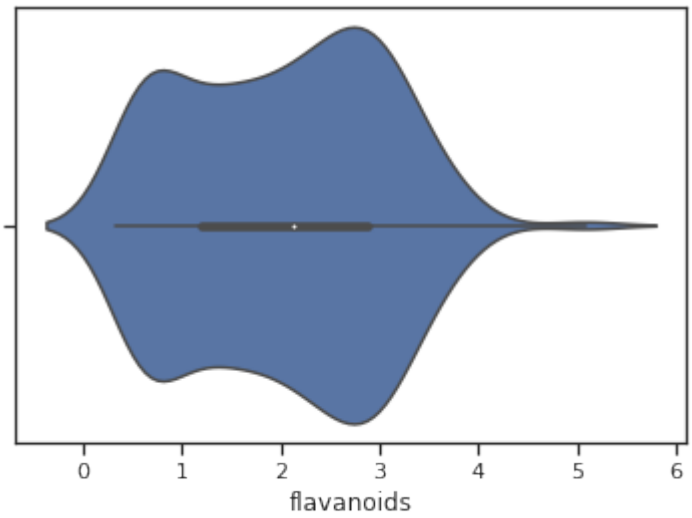


4.6. Диаграмма в виде скрипки

Похожа на предыдущую диаграмму, но по краям отображаются распределения плотности.

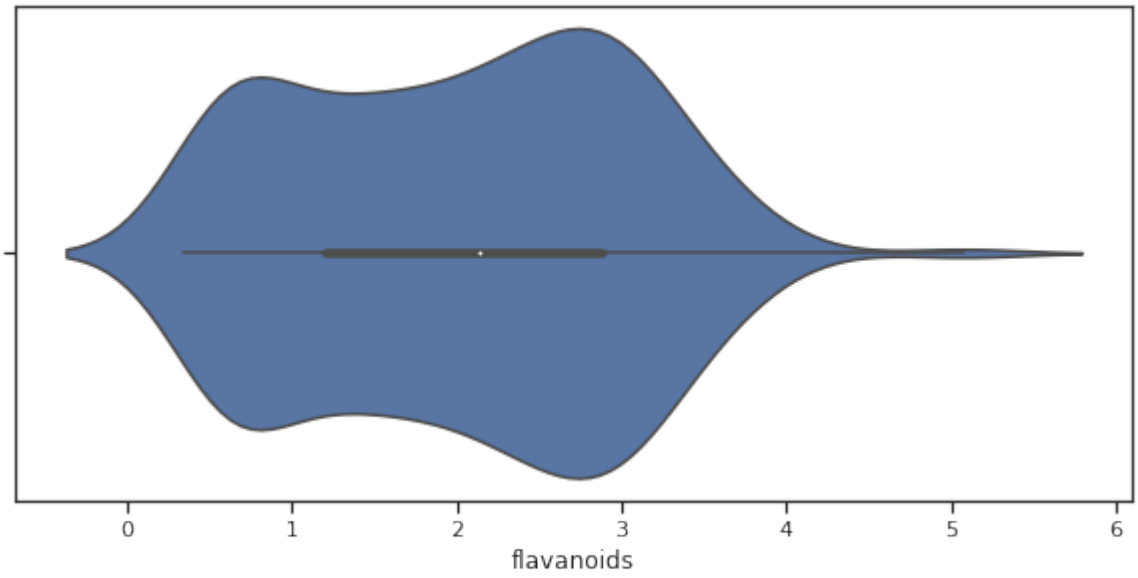
[ ]:
```
sns.violinplot(x=data1['flavanoids'])
```

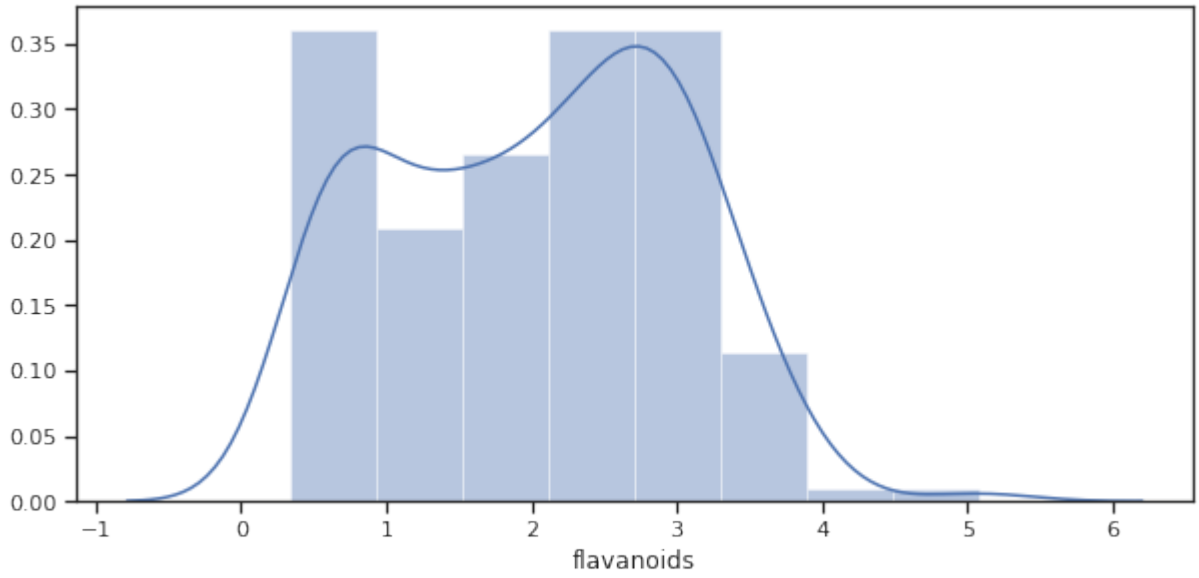
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9216bb8390>
```



[ ]:
```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data1['flavanoids'])
sns.distplot(data1['flavanoids'], ax=ax[1])
```

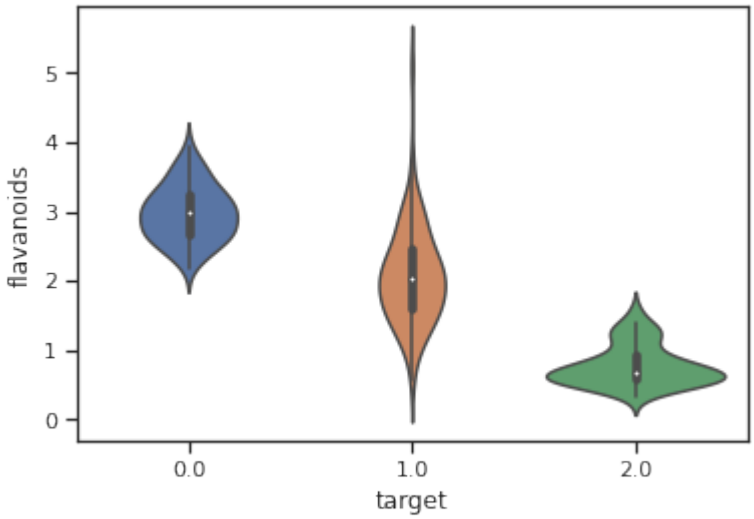
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9216b4d588>
```





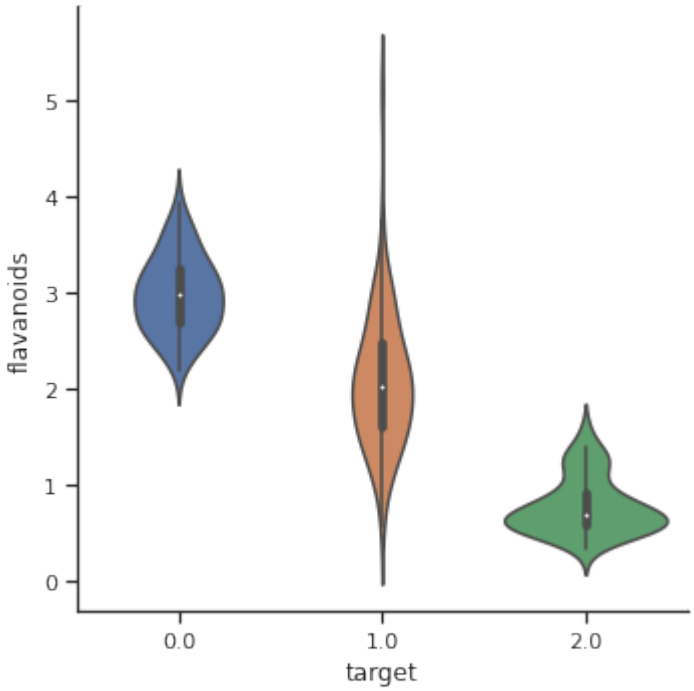
[ ]:
```
# Распределение параметра Flavanoids сгруппированные по Class.
sns.violinplot(x='target', y='flavanoids', data=data1)
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9216a7dd30>
```



[ ]:
```
sns.catplot(y='flavanoids', x='target', data=data1, kind="violin",
↪split=True)
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7f9216a6e390>
```



5. 4) Информация о корреляции признаков

Проверка корреляции признаков позволяет решить две задачи:

Понять какие признаки (колонки датасета) наиболее сильно коррелируют с целевым признаком (в нашем примере это колонка “Occupancy”). Именно эти признаки будут наиболее

информативными для моделей машинного обучения. Признаки, которые слабо коррелируют с целевым признаком, можно попробовать исключить из построения модели, иногда это повышает качество модели. Нужно отметить, что некоторые алгоритмы машинного обучения автоматически определяют ценность того или иного признака для построения модели. Понять какие нецелевые признаки линейно зависимы между собой. Линейно зависимые признаки, как правило, очень плохо влияют на качество моделей. Поэтому если несколько признаков линейно зависимы, то для построения модели из них выбирают какой-то один признак.

[ ]:
```
data1.corr()
```

[ ]:
```
                               alcohol  malic_acid  …   proline   ?
  ↪target
alcohol                       1.000000    0.094397  …  0.643720 -0.
  ↪328222
malic_acid                    0.094397    1.000000  … -0.192011  0.
  ↪437776
ash                           0.211545    0.164045  …  0.223626 -0.
  ↪049643
alcalinity_of_ash            -0.310235    0.288500  … -0.440597  0.
  ↪517859
magnesium                     0.270798   -0.054575  …  0.393351 -0.
  ↪209179
total_phenols                 0.289101   -0.335167  …  0.498115 -0.
  ↪719163
flavanoids                    0.236815   -0.411007  …  0.494193 -0.
  ↪847498
nonflavanoid_phenols         -0.155929    0.292977  … -0.311385  0.
  ↪489109
proanthocyanins               0.136698   -0.220746  …  0.330417 -0.
  ↪499130
color_intensity               0.546364    0.248985  …  0.316100  0.
  ↪265668
hue                          -0.071747   -0.561296  …  0.236183 -0.
  ↪617369
od280/od315_of_diluted_wines  0.072343   -0.368710  …  0.312761 -0.
  ↪788230
proline                       0.643720   -0.192011  …  1.000000 -0.
  ↪633717
target                       -0.328222    0.437776  … -0.633717  1.
  ↪000000

[14 rows x 14 columns]
```

Корреляционная матрица содержит коэффициенты корреляции между всеми парами признаков.

Корреляционная матрица симметрична относительно главной диагонали. На главной диагонали расположены единицы (корреляция признака самого с собой).

На основе корреляционной матрицы можно сделать следующие выводы:

Целевой признак наиболее сильно коррелирует с флаваноидами (0.84), разбавленностью (0.78) и общим содержанием фенолов (0.71). Эти признаки обязательно следует оставить в мо-

дели. Целевой признак отчасти коррелирует с оттенком (0.61) и сортом (0.63), алкоголем (0.32), яблочной кислотой(0.43), содержанием щёлочи(0.51), нефлаваноидными фенолами (0.48), проантоцианидинами (0.49). Эти признаки стоит также оставить в модели. Целевой признак слабо коррелирует с щёлочью (0.04) и магнием (0.2), интенсивностью цвета (0.26). Скорее всего эти признаки стоит исключить из модели, возможно они только ухудшат качество модели. Флаваноиды и Общее содержание фенолов достаточно сильно коррелируют между собой (0.86). Это неудивительно, ведь Флаваноиды - подвид фенолов. Поэтому из этих признаков в модели можно оставлять только один. Также можно сделать вывод, что выбирая из признаков флаваноиды и фенолы лучше выбрать флаваноиды, потому что он сильнее коррелирован с целевым признаком. Если линейно зависимые признаки сильно коррелированы с целевым, то оставляют именно тот признак, который коррелирован с целевым сильнее.

[ ]:
```
data1.corr(method='pearson')
```

[ ]:
```
                               alcohol  malic_acid  …   proline   ?
  ↪target
alcohol                       1.000000    0.094397  …  0.643720 -0.
  ↪328222
malic_acid                    0.094397    1.000000  … -0.192011  0.
  ↪437776
ash                           0.211545    0.164045  …  0.223626 -0.
  ↪049643
alcalinity_of_ash            -0.310235    0.288500  … -0.440597  0.
  ↪517859
magnesium                     0.270798   -0.054575  …  0.393351 -0.
  ↪209179
total_phenols                 0.289101   -0.335167  …  0.498115 -0.
  ↪719163
flavanoids                    0.236815   -0.411007  …  0.494193 -0.
  ↪847498
nonflavanoid_phenols         -0.155929    0.292977  … -0.311385  0.
  ↪489109
proanthocyanins               0.136698   -0.220746  …  0.330417 -0.
  ↪499130
color_intensity               0.546364    0.248985  …  0.316100  0.
  ↪265668
hue                          -0.071747   -0.561296  …  0.236183 -0.
  ↪617369
od280/od315_of_diluted_wines  0.072343   -0.368710  …  0.312761 -0.
  ↪788230
proline                       0.643720   -0.192011  …  1.000000 -0.
  ↪633717
target                       -0.328222    0.437776  … -0.633717  1.
  ↪000000

[14 rows x 14 columns]
```

[ ]:
```
data1.corr(method='kendall')
```

[ ]:

```
                              alcohol  malic_acid  …   proline
↪target
alcohol                      1.000000    0.093844  …  0.449387 -0.
↪238984
malic_acid                   0.093844    1.000000  … -0.044660  0.
↪247494
ash                          0.170154    0.158178  …  0.171574 -0.
↪038085
alcalinity_of_ash           -0.212978    0.210119  … -0.313218  0.
↪449402
magnesium                    0.250506    0.050869  …  0.343016 -0.
↪184992
total_phenols                0.209099   -0.174929  …  0.280203 -0.
↪590404
flavanoids                   0.191087   -0.211918  …  0.263661 -0.
↪725255
nonflavanoid_phenols        -0.109554    0.175129  … -0.174108  0.
↪379234
proanthocyanins              0.133526   -0.168714  …  0.204172 -0.
↪450225
color_intensity              0.434353    0.195607  …  0.316632  0.
↪065124
hue                         -0.021717   -0.388707  …  0.143508 -0.
↪479229
od280/od315_of_diluted_wines 0.061513   -0.162909  …  0.151559 -0.
↪607572
proline                      0.449387   -0.044660  …  1.000000 -0.
↪406260
target                      -0.238984    0.247494  … -0.406260  1.
↪000000

[14 rows x 14 columns]
```

[ ]:

```
data1.corr(method='spearman')
```

[ ]:

```
                              alcohol  malic_acid  …   proline
↪target
alcohol                      1.000000    0.140430  …  0.633580 -0.
↪354167
malic_acid                   0.140430    1.000000  … -0.057466  0.
↪346913
ash                          0.243722    0.230674  …  0.253163 -0.
↪053988
alcalinity_of_ash           -0.306598    0.304069  … -0.456090  0.
↪569792
magnesium                    0.365503    0.080188  …  0.507575 -0.
↪250498
total_phenols                0.310920   -0.280225  …  0.419470 -0.
↪726544
```

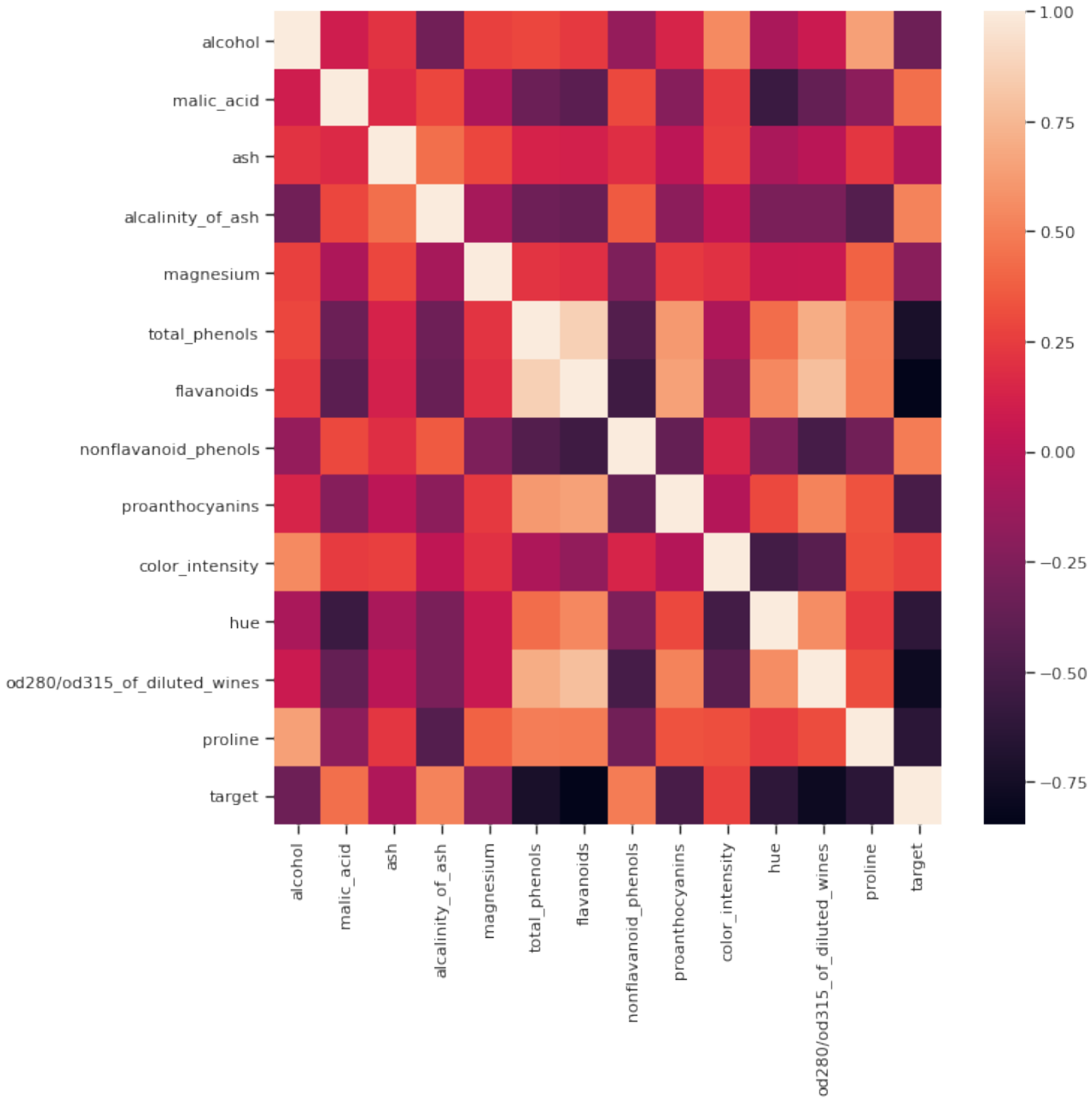
```
flavanoids                     0.294740   -0.325202  …   0.429904 -0.
 ↪854908
nonflavanoid_phenols          -0.162207    0.255236  … -0.270112  0.
 ↪474205
proanthocyanins                0.192734   -0.244825  …   0.308249 -0.
 ↪570648
color_intensity                0.635425    0.290307  …   0.457096  0.
 ↪131170
hue                           -0.024203   -0.560265  …   0.207740 -0.
 ↪616570
od280/od315_of_diluted_wines   0.103050   -0.255185  …   0.253266 -0.
 ↪743787
proline                        0.633580   -0.057466  …   1.000000 -0.
 ↪576383
target                        -0.354167    0.346913  … -0.576383  1.
 ↪000000

[14 rows x 14 columns]
```

5.1. Тепловая карта

Обеспечивает более лёгкий визуальный анализ матрицы.

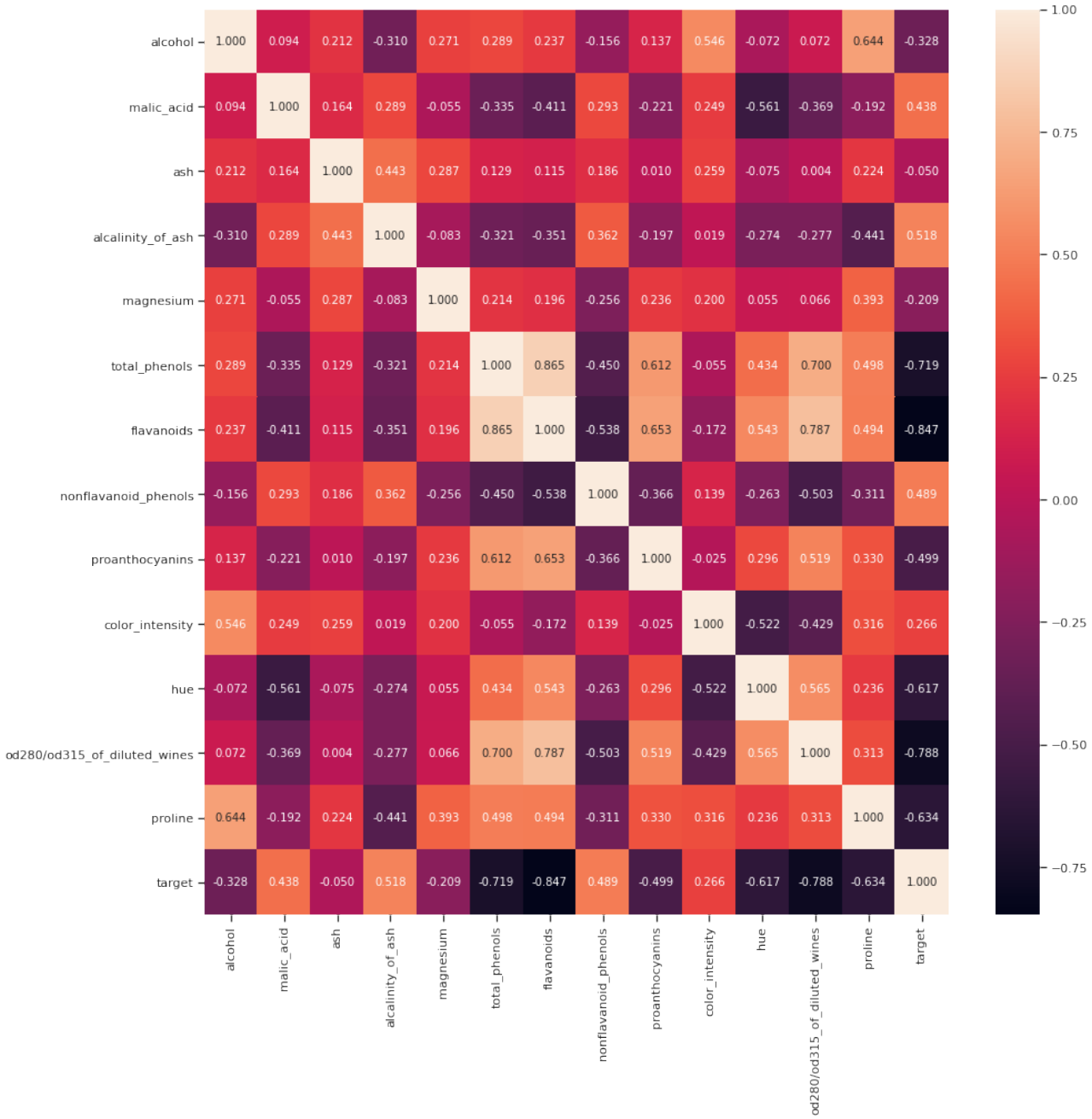
[ ]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(data1.corr())
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f92169cc518>
```



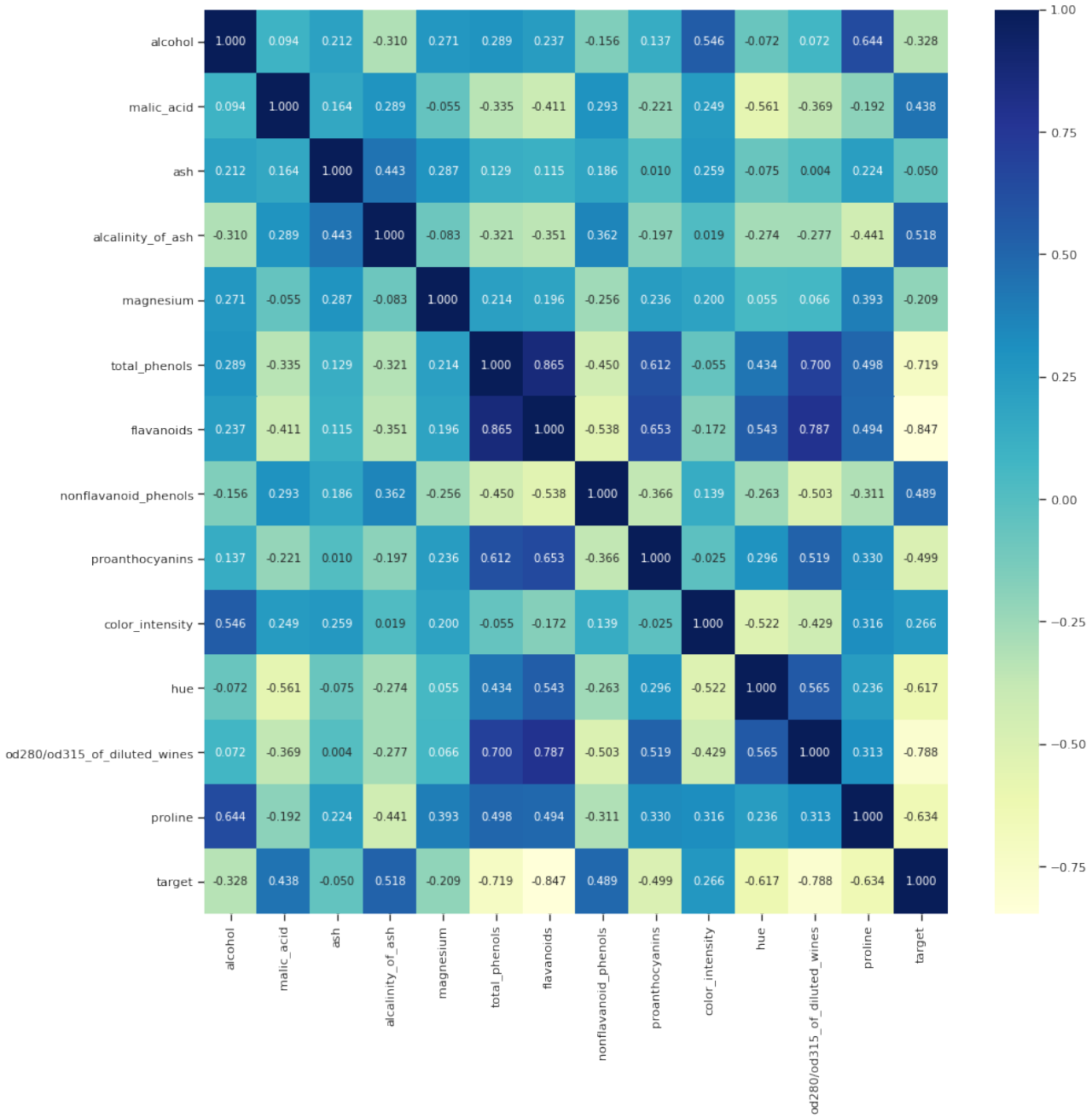
[ ]:
```
# Вывод значений в ячейках
fig, ax = plt.subplots(figsize=(15,15))
sns.heatmap(data1.corr(), annot=True, fmt='.3f')
```

[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f92156181d0>



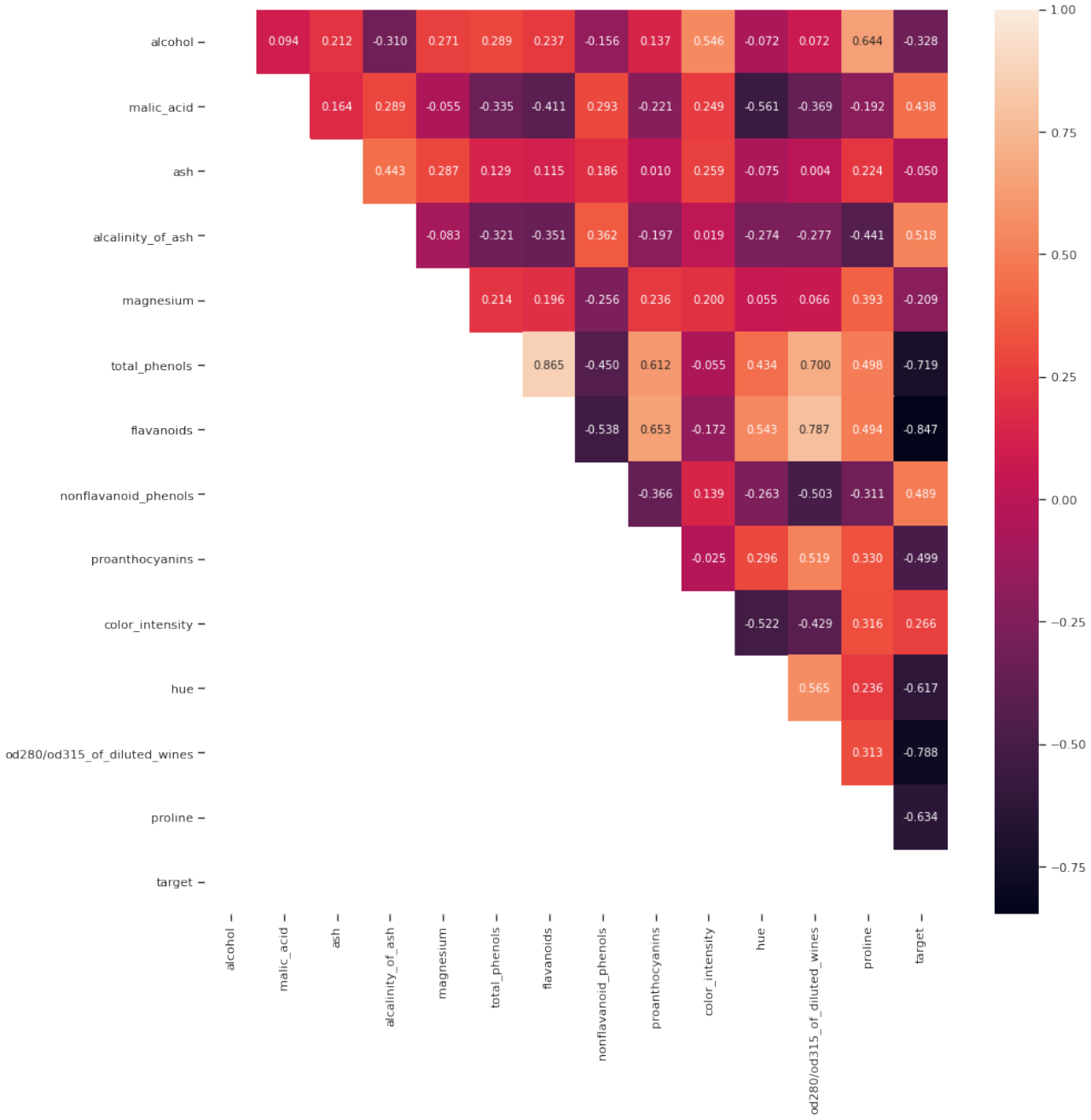
[ ]:
```
# Изменение цветовой гаммы
fig, ax = plt.subplots(figsize=(15,15))
sns.heatmap(data1.corr(), cmap='YlGnBu', annot=True, fmt='.3f')
```

[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9215639550>

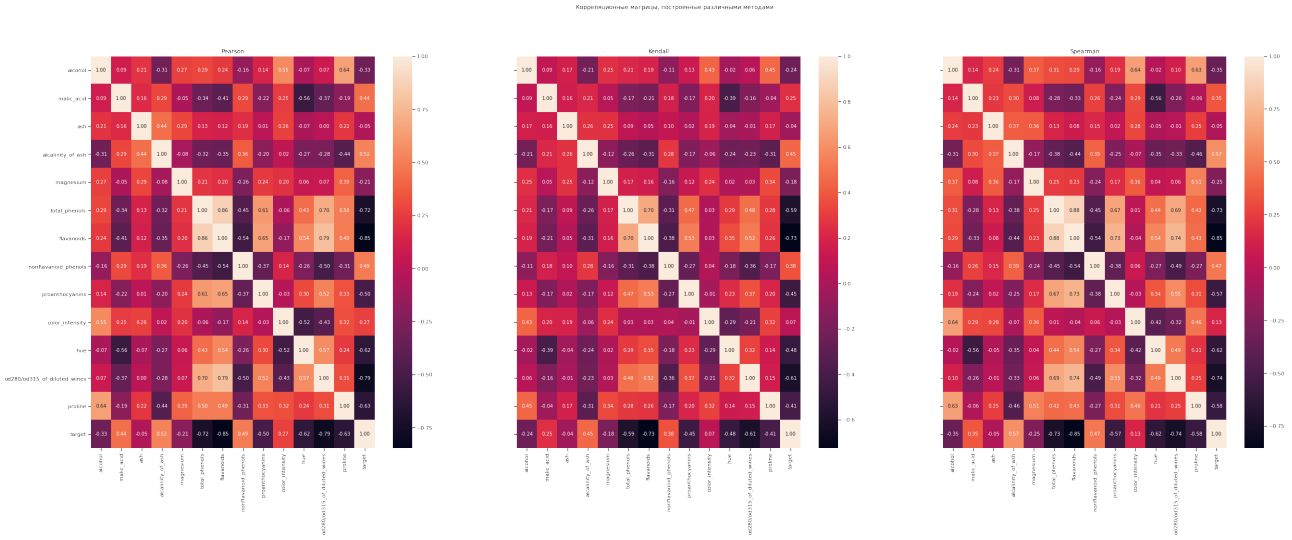


[ ]:
```
# Треугольный вариант матрицы
mask = np.zeros_like(data1.corr(), dtype=np.bool)
# чтобы оставить нижнюю часть матрицы
# mask[np.triu_indices_from(mask)] = True
# чтобы оставить верхнюю часть матрицы
mask[np.tril_indices_from(mask)] = True
fig, ax = plt.subplots(figsize=(15,15))
sns.heatmap(data1.corr(), mask=mask, annot=True, fmt='.3f')
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f92151f4c88>
```



[ ]:
```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row',
↪figsize=(45,15))
sns.heatmap(data1.corr(method='pearson'), ax=ax[0], annot=True, fmt='.
↪2f')
sns.heatmap(data1.corr(method='kendall'), ax=ax[1], annot=True, fmt='.
↪2f')
sns.heatmap(data1.corr(method='spearman'), ax=ax[2], annot=True, fmt='.
↪2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```



6. Список литературы:

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование и визуализация данных» [Электронный ресурс] // GitHub. —- 2020. —- Режим доступа: https://github.com/ugapanyuk/ml_course_2020/wiki/LAB_MMO__EDA_VISUALIZATION (дата обращения: 21.02.2020).

[2] Waskom M. seaborn 0.1.0 documentation [Electronic resource] // PyData. — 2020. — Access mode: https://seaborn.pydata.org/ (online; accessed: 21.02.2020).

[3] pandas 1.0.1 documentation [Electronic resource] // PyData. — 2020. — Access mode: http://pandas.pydata.org/pandas-docs/stable/ (online; accessed: 21.02.2020).