jupyter Lab1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Технологии машинного обучения

Отчет по лабораторной работе № 1

Разведочный анализ данных. Исследование и визуализация данных.

Студент:
группы ИУ5-64Б
Веласко Е.А.

2021 г.
Москва

1)Текстовое описание набора данных

В качестве набора данных мы будем использовать набор данных по распознаванию качества красных вин

https://www.kaggle.com/uciml/273/downloads

Датасет состоит из одного файла

winequality_red.csv

Файл содержит следующие колонки:

1. fixed acidity – фиксированная кислотность
2. volatile acidity – летучая кислотность
3. citric acid – лимонная кислота
4. residual sugar – остаточный сахар
5. chlorides – хлориды
6. free sulfur dioxide – свободный диоксид серы
7. total sulfur dioxide – общий диоксид серы
8. density – плотность
9. pH – кислотный индекс
10. sulphates – сульфаты
11. alcohol – алкоголь
12. quality – оценка качества (выходной параметр)

Импорт библиотек

[ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Загрузка данных

[ ]:
```
data = pd.read_csv('data/winequality_red.csv', sep=",")
```

2) Основные характеристики датасета

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

Out[ ]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

[ ]:
```
# Размер датасета - 2000 строк, 13 колонок
data.shape
```

Out[ ]:
```
(1599, 12)
```

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

Out[ ]:
```
Всего строк: 1599
```

[ ]:
```
# Список колонок
data.columns.values
```

Out[ ]:
```
array(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide',
       'density', 'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='<U20')
```

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

Out[ ]:
```
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                   int64
dtype: object
```

[ ]:
```
# Проверка наличия пустых значений
# Цикл по всем колонкам
for col in data.columns:
    # Количество пустых значений - все значения исключая заполненные
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
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
```

[ ]:
```
# Основные статистические характеристики набора данных
data.describe()
```

Out[ ]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	q
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.0
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.6
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.8
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.0
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.0

8 rows × 12 columns

[ ]:
```

```

3) Визуальное исследование датасета

Диаграмма рассеивания

Позволяет построить распределение двух числовых данных и визуально обнаружить наличие зависимости: так представляется, есть значения отсортированы наиболее, по-нашему...

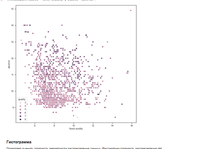
[ ]:
```
fig, ax = plt.subplots(figsize=(15,10))
sns.scatterplot(data=data, x='fixed acidity', y='alcohol', ax=ax)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1ca1a90>
```

Посмотрим на зависит качество алкоголя от его кислоты и фиксированной кислотности

[ ]:
```
fig, ax = plt.subplots(figsize=(15,10))
sns.scatterplot(data=data, x='fixed acidity', y='alcohol', hue='quality')
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1c9ca10>
```



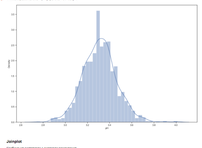
Гистограмма

Позволяет оценить плотность вероятности распределения данных. Визуально похожа на распределение pH

[ ]:
```
fig, ax = plt.subplots(figsize=(15,10))
sns.distplot(data['pH'])
```

```
C:\Users\Evelina\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1c9ca10>
```

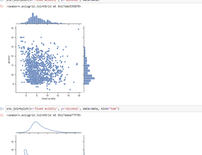


Jointplot

Комбинация гистограмм и диаграмм рассеивания

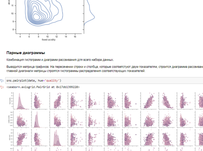
[ ]:
```
sns.jointplot(x='fixed acidity', y='alcohol', data=data)
```

Out[ ]:
```
<seaborn.axisgrid.JointGrid at 0x7f8ed1f5bf90>
```



[ ]:
```
sns.jointplot(x='fixed acidity', y='alcohol', data=data, kind="hex")
```

Out[ ]:
```
<seaborn.axisgrid.JointGrid at 0x7f8ed1f7f7d0>
```



Парные диаграммы

Комбинация гистограмм и диаграмм рассеивания для всего набора данных.

Позволяет оценить наличие или отсутствие связи между парами признаков, а также характеристики двух показателей: строится диаграмма рассеивания. В главной диагонали наносятся гистограммы распределения соответствующих признаков.

[ ]:
```
sns.pairplot(data, hue='quality')
```

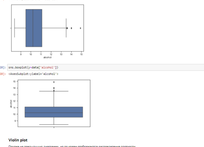
Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7f8ed1f5b1d0>
```

Ящик с усами

Отображает одномерное распределение вероятности

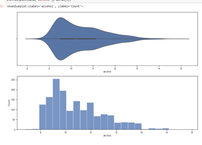
[ ]:
```
sns.boxplot(y='alcohol', data=data)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1f5b1d0>
```



[ ]:
```
sns.boxplot(x='alcohol', data=data)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1f5b1d0>
```



Violin plot

Похожа на предыдущую диаграмму, но по сути отображает распределение плотности

[ ]:
```
fig, ax = plt.subplots(1, 1, figsize=(15,10))
sns.violinplot(data=data, x='alcohol', ax=ax)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1f5b1d0>
```

Информация о корректных признаков

Позволяет корректно признаки позволяет решить две задачи:

- Понять какие признаки и сколько данных необходимо для обучения моделей с целевым признаком (в нашем примере это колонка "quality"). Именно это признаки будут наиболее информативны для модели машинного обучения. Признаки, которые стоит игнорировать с целевым признаком. Например, если мы хотим предсказывать стоимость квартиры, то признак "Количество этажей в доме" скорее всего будет коррелировать с ценой квартиры, а признак "Индекс почтового отделения" скорее всего нет (хотя при желании для почтовых индексов...)
- Понять какие недостающие признаки можно заполнить между собой. Пример заполнения признаков: не бывает очень плохого качества квартир. Поэтому если мы знаем количество этажей дома, то для построения модели их можно заполнить между собой (например средним).

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

Out[ ]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006	-0.061668	0.124052
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987	-0.202288	-0.390558
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770	0.109903	0.226373
...	...	...	...	...	...	...	...	...	...	...	...	...
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.205633	0.093595	1.000000	0.476166
quality	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-0.057731	0.251397	0.476166	1.000000

12 rows × 12 columns

Целевой признак наиболее сильно коррелирует с предыдущим алкоголя (0.47) и «вторым» (0.25). Эти признаки обязательно следует оставить в модели.

Heatmap

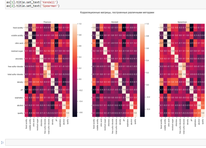
Тепловой отличны корректных различными цветами.

[ ]:
```
sns.heatmap(data.corr())
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1f5b1d0>
```

[ ]:
```
sns.heatmap(data.corr(), cmap="YlGnBu", annot=True, fmt='.2f')
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ed1f5b1d0>
```



[ ]:
```
fig, ax = plt.subplots(1, 1, figsize=(15,10))
sns.heatmap(data.corr(), annot=True, fmt='.2f', ax=ax)
sns.heatmap(data.corr(method='pearson'), annot=True, fmt='.2f')
sns.heatmap(data.corr(method='kendall'), annot=True, fmt='.2f')
sns.heatmap(data.corr(method='spearman'), annot=True, fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
fig.tight_layout()
fig.subplots_adjust(top=0.88)
ax.set_title('Pearson')
ax.set_title('Kendall')
ax.set_title('Spearman')
```

Out[ ]:
```
Text(0.5, 1.0, 'Spearman')
```


[ ]:
```

```