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

Лабораторная работа №3
по курсу «Методы машинного обучения»

«Обработка признаков (часть 2)»

ИСПОЛНИТЕЛЬ:

Ерохин И.А.
Группа ИУ5-24М

_____

"__"_____2022 г.

Москва 2022

Цель работы:

Изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

Задание:

Выбрать один или несколько наборов данных (датасетов) для решения следующих задач. Каждая задача может быть решена на отдельном датасете, или несколько задач могут быть решены на одном датасете. Просьба не использовать датасет, на котором данная задача решалась в лекции. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:

1) масштабирование признаков (не менее чем тремя способами);
2) обработку выбросов для числовых признаков (по одному способу для удаления выбросов и для замены выбросов);
3) обработку по крайней мере одного нестандартного признака (который не является числовым или категориальным);
4) отбор признаков:
   a) один метод из группы методов фильтрации (filter methods);
   b) один метод из группы методов обертывания (wrapper methods);
   c) один метод из группы методов вложений (embedded methods).

Выполнение:

Масштабирование признаков

In [4]:
```
data = pd.read_csv('WineQT.csv', sep = ',')
data.head()
```

Out[4]:

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

In [40]:
```
data.describe()
```

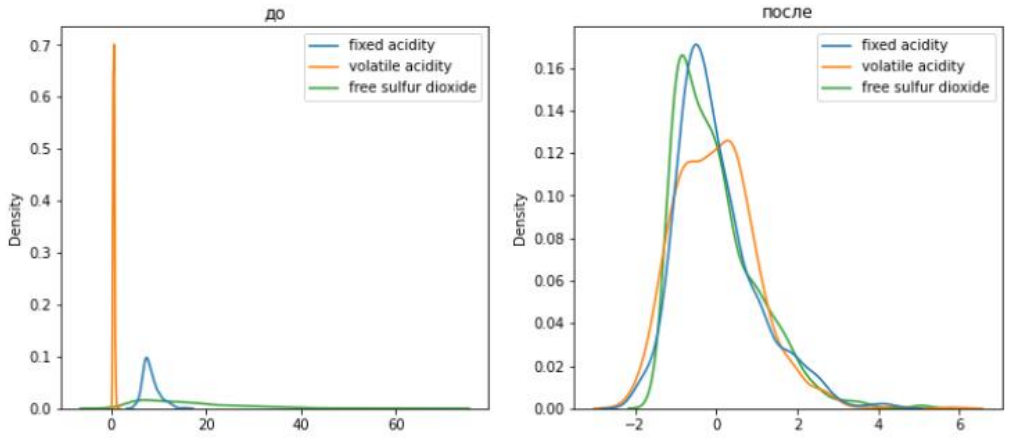
Out[40]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density
count	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000
mean	8.311111	0.531339	0.268364	2.532152	0.086933	15.615486	45.914698	0.996730
std	1.747595	0.179633	0.196686	1.355917	0.047267	10.250486	32.782130	0.001925
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070
25%	7.100000	0.392500	0.090000	1.900000	0.070000	7.000000	21.000000	0.995570
50%	7.900000	0.520000	0.250000	2.200000	0.079000	13.000000	37.000000	0.996680
75%	9.100000	0.640000	0.420000	2.600000	0.090000	21.000000	61.000000	0.997845
max	15.900000	1.580000	1.000000	15.500000	0.611000	68.000000	289.000000	1.003690

In [69]:
```
# DataFrame не содержащий целевой признак
X_ALL = data.drop('citric acid', axis=1)
```

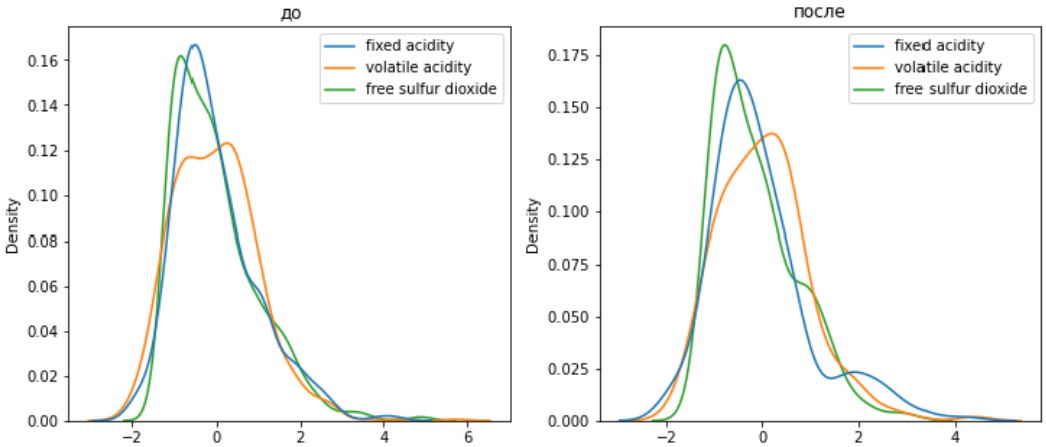
In [76]:
```
# масштабирование на z оценке

data_arr = StandardScaler().fit_transform(X_ALL)

data_z = pd.DataFrame(data_arr, columns = X_ALL.columns)
```

```
draw_kde(['fixed acidity', 'volatile acidity', 'free sulfur dioxide'],
         data, data_z, 'до', 'после')
```

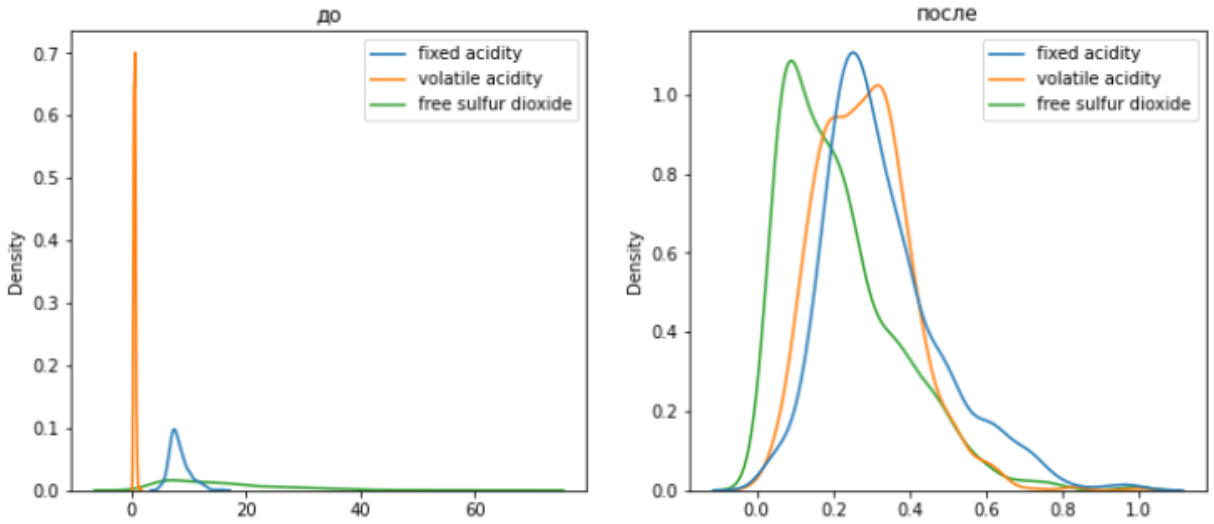



```
train_x, test_x, = train_test_split(X_ALL, test_size = 0.2, random_state = 1)

ss = StandardScaler()
ss.fit(train_x)
data_arr1 = ss.transform(train_x)
data_arr2 = ss.transform(test_x)

data_z_train = pd.DataFrame(data_arr1, columns = X_ALL.columns)
data_z_test = pd.DataFrame(data_arr2, columns = X_ALL.columns)
```

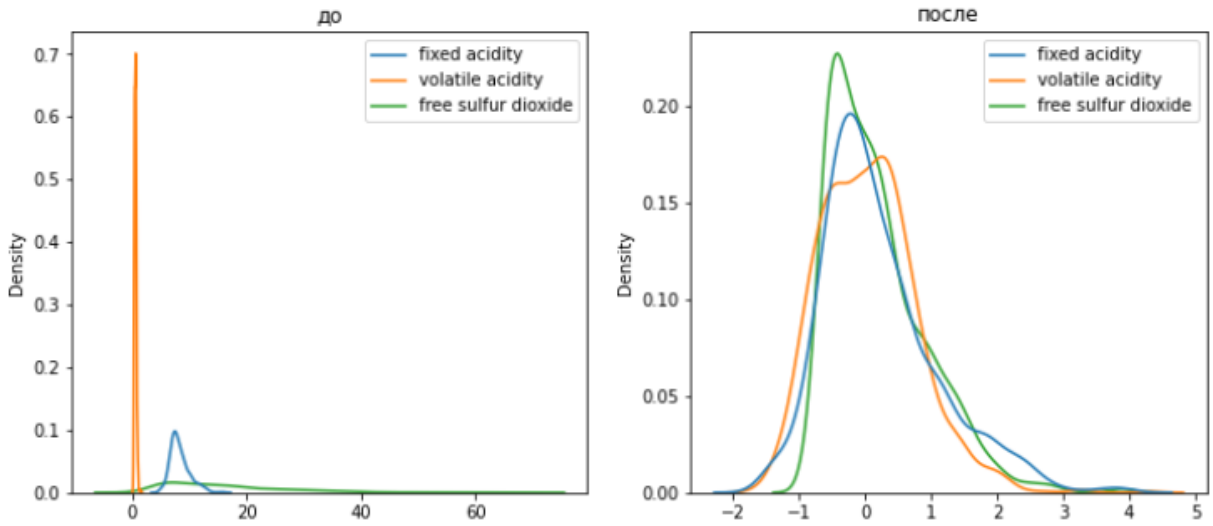
```
draw_kde(['fixed acidity', 'volatile acidity', 'free sulfur dioxide'],
         data_z_train, data_z_test, 'до', 'после')
```



```
# min max

data_mm_arr = MinMaxScaler().fit_transform(X_ALL)
data_mm = pd.DataFrame(data_mm_arr, columns = X_ALL.columns)

draw_kde(['fixed acidity', 'volatile acidity', 'free sulfur dioxide'],
         data, data_mm, 'до', 'после')
```

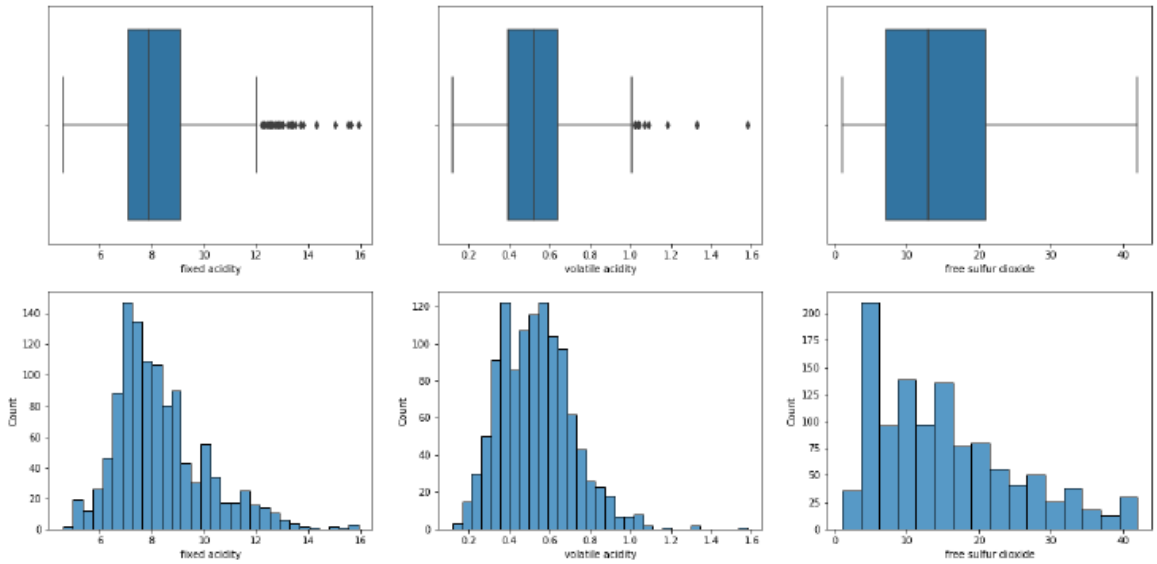



```
# по медиане

data_rob_arr = RobustScaler().fit_transform(X_ALL)
data_rob = pd.DataFrame(data_rob_arr, columns = X_ALL.columns)

draw_kde(['fixed acidity', 'volatile acidity', 'free sulfur dioxide'],
         data, data_rob, 'до', 'после')
```



Обработка выбросов для числовых признаков

```
show_data(data,'fixed acidity', 'volatile acidity', 'free sulfur dioxide')
```




```
def del_fuc(some_data, col_name, low, up):
    buf = np.where(some_data[col_name] > up, True,
                   np.where(some_data[col_name] < low, True, False))
    res = some_data.loc[~(buf)]
    return res
```

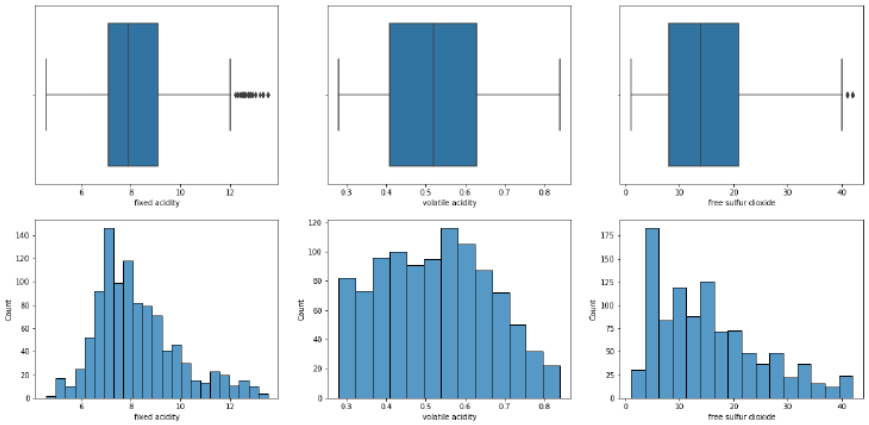
```
def rep_fuc(some_data, col_name, low, up):
    res = some_data.copy()
    res[col_name] = np.where(some_data[col_name]>up, up,
              np.where(some_data[col_name]<low, low, some_data[col_name]))
    return res
```

```
# правило 3 сигм

low = data['fixed acidity'].mean() - 3 * data['fixed acidity'].std()
up = data['fixed acidity'].mean() + 3 * data['fixed acidity'].std()
del_res = del_fuc(data, 'fixed acidity', low, up)
rep_res = rep_fuc(data, 'fixed acidity', low, up)
```

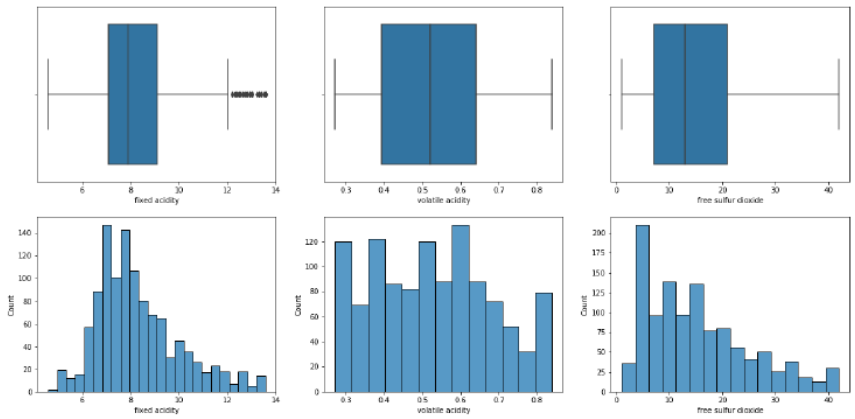
```
# 5, 95 квантилей

low = data['volatile acidity'].quantile(0.05)
up = data['volatile acidity'].quantile(0.95)
#show_data(data,'fixed acidity', 'volatile acidity', 'free sulfur dioxide')
del_res2 = del_fuc(del_res, 'volatile acidity', low, up)
rep_res2 = rep_fuc(rep_res, 'volatile acidity', low, up)
```

```
# межквартильный размах

iqr = data['free sulfur dioxide'].quantile(0.75) - data['free sulfur dioxide'].quantil
low = data['free sulfur dioxide'].quantile(0.25) - 1.5*iqr
up = data['free sulfur dioxide'].quantile(0.75) + 1.5*iqr

del_res3 = del_fuc(del_res2, 'free sulfur dioxide', low, up)
rep_res3 = rep_fuc(rep_res2, 'free sulfur dioxide', low, up)
```

```
show_data(del_res3,'fixed acidity', 'volatile acidity', 'free sulfur dioxide')
```




```
show_data(rep_res3,'fixed acidity', 'volatile acidity', 'free sulfur dioxide')
```



Обработка нестандартного признака

In [131…
```
# обработка нестандартного признака
data2 = pd.read_csv('top_100_movies_by_genres.csv', sep=",")
```

In [132…
```
data2.head()
```

Out[132…

	Genre	Rank	RatingTomatometer	Title	No. of Reviews
0	Action & Adventure	1.0	96%	Black Panther (2018)	525
1	Action & Adventure	2.0	94%	Avengers: Endgame (2019)	547
2	Action & Adventure	3.0	97%	Mission: Impossible - Fallout (2018)	437
3	Action & Adventure	4.0	97%	Mad Max: Fury Road (2015)	434
4	Action & Adventure	5.0	97%	Spider-Man: Into the Spider-Verse (2018)	393

In [133…
```
data2.dtypes
```

Out[133…
```
Genre                object
Rank                float64
RatingTomatometer    object
Title                object
No. of Reviews        int64
dtype: object
```

In [147…
```
year = []
for tit in data2['Title']:
    res = re.search(r'\d\d\d\d', tit)
    year.append(res.group(0))
```

```
n [148…
data2['Year'] = year
```

```
n [150…
data2.head()
```

ut[150…

	Genre	Rank	RatingTomatometer	Title	No. of Reviews	Year
0	Action & Adventure	1.0	96%	Black Panther (2018)	525	2018
1	Action & Adventure	2.0	94%	Avengers: Endgame (2019)	547	2019
2	Action & Adventure	3.0	97%	Mission: Impossible - Fallout (2018)	437	2018
3	Action & Adventure	4.0	97%	Mad Max: Fury Road (2015)	434	2015
4	Action & Adventure	5.0	97%	Spider-Man: Into the Spider-Verse (2018)	393	2018

Отбор признаков (filter methods)

```
# метод фильтрации (корреляция)

def make_corr_df(some_data):
    cr = some_data.corr()
    cr = cr.abs().unstack()
    cr = cr.sort_values(ascending=False)
    cr = cr[cr >= 0.5]
    cr = cr[cr < 1]
    cr = pd.DataFrame(cr).reset_index()
    cr.columns = ['f1', 'f2', 'corr']
    return cr
```

```
make_corr_df(data)
```

	f1	f2	corr
0	pH	fixed acidity	0.685163
1	fixed acidity	pH	0.685163
2	density	fixed acidity	0.681501
3	fixed acidity	density	0.681501
4	fixed acidity	citric acid	0.673157
5	citric acid	fixed acidity	0.673157
6	total sulfur dioxide	free sulfur dioxide	0.671150
7	free sulfur dioxide	total sulfur dioxide	0.671150
8	pH	citric acid	0.546339
9	citric acid	pH	0.546339
10	volatile acidity	citric acid	0.544187
11	citric acid	volatile acidity	0.544187

```
# Обнаружение групп коррелирующих признаков
def corr_groups(cr):
    grouped_feature_list = []
    correlated_groups = []
    for feature in cr['f1'].unique():
        if feature not in grouped_feature_list:
            # находим коррелирующие признаки
            correlated_block = cr[cr['f1'] == feature]
            cur_dups = list(correlated_block['f2'].unique()) + [feature]
            grouped_feature_list = grouped_feature_list + cur_dups
            correlated_groups.append(cur_dups)
    return correlated_groups
```

```
# Группы коррелирующих признаков
corr_groups(make_corr_df(data))
```

```
[['fixed acidity', 'citric acid', 'pH'],
 ['fixed acidity', 'density'],
 ['free sulfur dioxide', 'total sulfur dioxide'],
 ['citric acid', 'volatile acidity']]
```

Отбор признаков (wrapper methods)

```
# оберточный (полный перебор)

from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=3)
```

```
data_x = data.drop('quality', axis = 1)
data_y = data['quality']
```

```
efs1 = EFS(knn,
           min_features=2,
           max_features=8,
           scoring='accuracy',
           print_progress=True,
           cv=5)

efs1 = efs1.fit(data_x, data_y)

print('Best accuracy score: %.2f' % efs1.best_score_)
print('Best subset (indices):', efs1.best_idx_)
print('Best subset (corresponding names):', efs1.best_feature_names_)
```

```
Features: 3784/3784
Best accuracy score: 0.52
Best subset (indices): (2, 7, 9, 10)
Best subset (corresponding names): ('citric acid', 'density', 'sulphates', 'alcohol')
```

Отбор признаков (embedded methods)

```
# линейная регрессия
som_d = Lasso(fit_intercept = False, random_state = 2, max_iter = 2000)
som_d.fit(new_data_x, new_data_y)
```

```
Lasso(fit_intercept=False, max_iter=2000, random_state=2)
```

```
som_d.coef_
```

```
array([0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 4.58721205, 0.11990509, 0.        ,
       0.        , 2.29838969])
```

```
som_d_res = SelectFromModel(som_d)
som_d_res.fit(new_data_x, new_data_y)
list(zip(data.columns, som_d_res.get_support()))
```

```
[('fixed acidity', False),
 ('volatile acidity', False),
 ('citric acid', False),
 ('residual sugar', False),
 ('chlorides', False),
 ('free sulfur dioxide', False),
 ('total sulfur dioxide', False),
 ('density', True),
 ('pH', True),
 ('sulphates', False),
 ('alcohol', False),
 ('quality', True)]
```

Вывод:

В результате работы были изучены способы предварительной обработки данных для дальнейшего формирования моделей.