Лабораторная работа №8. Разработка и оптимизация модели машинного обучения.

Используемый набор данных: [seeds (https://archive.ics.uci.edu/ml/datasets/seeds)](https://archive.ics.uci.edu/ml/datasets/seeds).

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
from sklearn.preprocessing import label_binarize
from sklearn.model_selection import train_test_split
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report, roc_curve, roc_auc_score
from itertools import cycle
import os
import requests

%matplotlib inline

pd.options.display.max_columns = None
```

In [2]:

```
def downloadFile(url, filePath):
    if not os.path.exists(filePath):
        req = requests.get(url)
        f = open(filePath, "wb")
        f.write(req.content)
        f.close

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00236"
downloadFile(url + "/seeds_dataset.txt", "dataset/seeds_dataset.txt")
```

In [3]:

```
headers = ["Area", "Perimeter", "Compactness", "Length of kernel", "Width of kernel",
"Asymmetry coefficient",
           "Length of kernel groove", "Class"]
data = pd.read_csv("dataset/seeds_dataset.txt", names=headers, sep="\t")
data = data.astype({"Class": "category"})
data.sample(40)
```

Out[3]:

	Area	Perimeter	Compactness	Length of kernel	Width of kernel	Asymmetry coefficient	Length of kernel groove	Class
70	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
151	12.01	13.52	0.8249	5.405	2.776	6.992	5.270	3
147	12.49	13.46	0.8658	5.267	2.967	4.421	5.002	3
143	12.22	13.32	0.8652	5.224	2.967	5.469	5.221	3
169	11.24	13.00	0.8359	5.090	2.715	3.521	5.088	3
148	12.70	13.71	0.8491	5.386	2.911	3.260	5.316	3
172	11.27	12.97	0.8419	5.088	2.763	4.309	5.000	3
125	18.75	16.18	0.8999	6.111	3.869	4.188	5.992	2
122	16.17	15.38	0.8588	5.762	3.387	4.286	5.703	2
12	13.89	14.02	0.8880	5.439	3.199	3.986	4.738	1
166	12.44	13.59	0.8462	5.319	2.897	4.924	5.270	3
144	11.82	13.40	0.8274	5.314	2.777	4.471	5.178	3
26	13.02	13.76	0.8641	5.395	3.026	3.373	4.825	1
142	13.34	13.95	0.8620	5.389	3.074	5.995	5.307	3
201	12.67	13.32	0.8977	4.984	3.135	2.300	4.745	3
100	16.41	15.25	0.8866	5.718	3.525	4.217	5.618	2
124	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2
85	18.27	16.09	0.8870	6.173	3.651	2.443	6.197	2
14	13.74	14.05	0.8744	5.482	3.114	2.932	4.825	1
126	18.65	16.41	0.8698	6.285	3.594	4.391	6.102	2
62	12.36	13.19	0.8923	5.076	3.042	3.220	4.605	1
77	20.71	17.23	0.8763	6.579	3.814	4.451	6.451	2
182	12.19	13.36	0.8579	5.240	2.909	4.857	5.158	3
10	15.26	14.85	0.8696	5.714	3.242	4.543	5.314	1
110	18.45	16.12	0.8921	6.107	3.769	2.235	5.794	2
193	10.82	12.83	0.8256	5.180	2.630	4.853	5.089	3
65	12.88	13.50	0.8879	5.139	3.119	2.352	4.607	1
186	11.81	13.45	0.8198	5.413	2.716	4.898	5.352	3
89	20.88	17.05	0.9031	6.450	4.032	5.016	6.321	2
23	12.08	13.23	0.8664	5.099	2.936	1.415	4.961	1
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1
42	13.16	13.55	0.9009	5.138	3.201	2.461	4.783	1
157	12.13	13.73	0.8081	5.394	2.745	4.825	5.220	3
145	11.21	13.13	0.8167	5.279	2.687	6.169	5.275	3
175	10.80	12.57	0.8590	4.981	2.821	4.773	5.063	3
162	12.05	13.41	0.8416	5.267	2.847	4.988	5.046	3

	Area	Perimeter	Compactness	Length of kernel	Width of kernel	Asymmetry coefficient	Length of kernel groove	Class
54	14.52	14.60	0.8557	5.741	3.113	1.481	5.487	1
33	13.94	14.17	0.8728	5.585	3.150	2.124	5.012	1
150	11.83	13.23	0.8496	5.263	2.840	5.195	5.307	3
36	16.20	15.27	0.8734	5.826	3.464	2.823	5.527	1

In [4]:

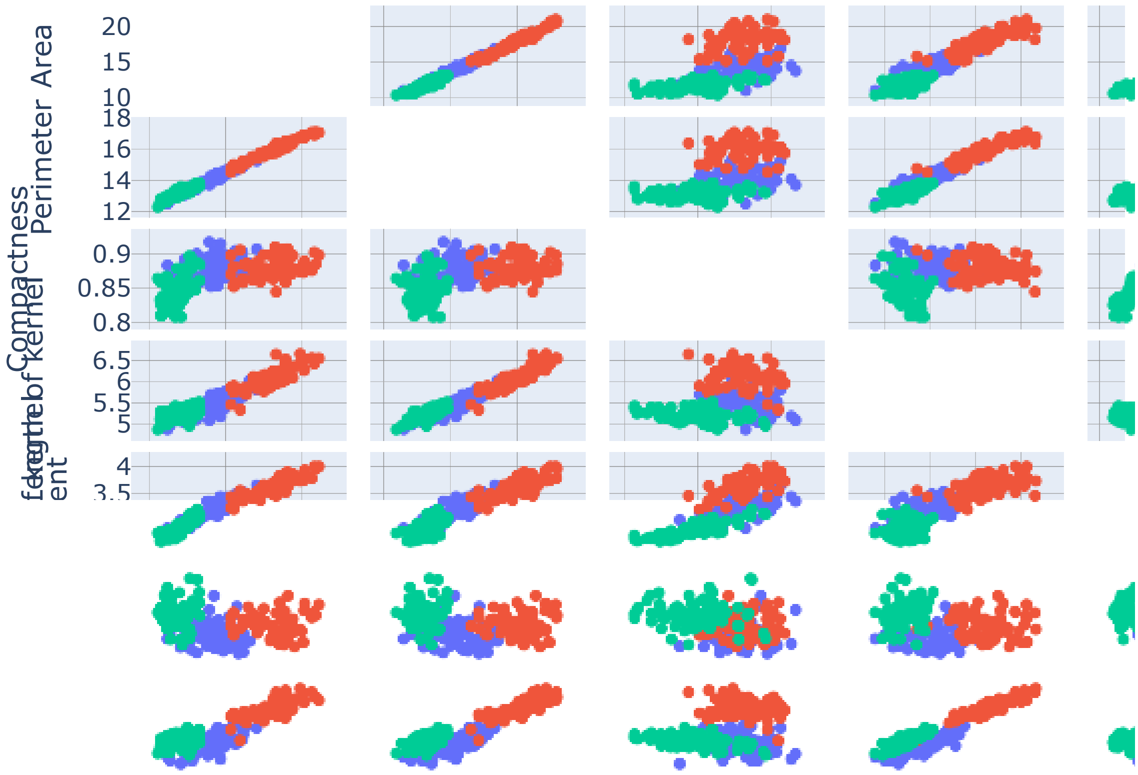
```
display(data.isna().sum())
display(data.describe())
```

```
Area                     0
Perimeter                0
Compactness              0
Length of kernel         0
Width of kernel          0
Asymmetry coefficient    0
Length of kernel groove  0
Class                    0
dtype: int64
```

	Area	Perimeter	Compactness	Length of kernel	Width of kernel	Asymmetry coefficient	Length c kerne groov
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.00000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.40807
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.49148
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.51900
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.04500
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.22300
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.87700
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.55000

In [5]:

```
plots = px.scatter_matrix(data, dimensions=headers[:len(headers) - 1], color="Class")
plots.update_traces(diagonal_visible=False)
plots.show()
```



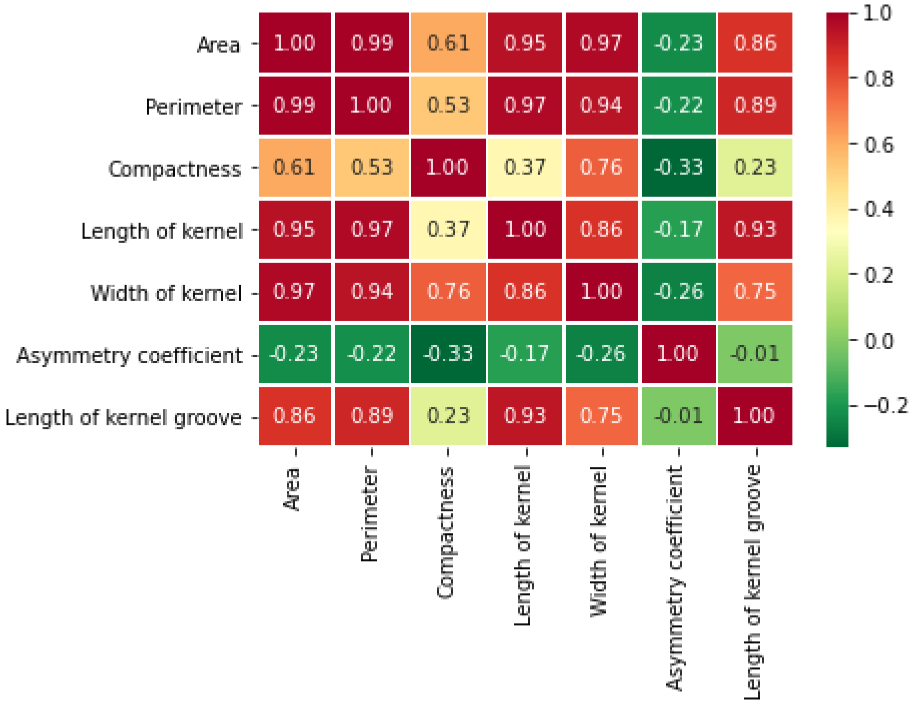
Построим [тепловую карту (https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D0%B2%D0%B0%D1%8F_%D](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D0%B2%D0%B0%D1%8F_%D) для визуализации корреляции между атрибутами.

In [6]:

```
sns.heatmap(data.corr(), annot=True, cmap='RdYlGn_r', linewidths=0.5, fmt= '.2f')
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c1a3430>
```



Выполним классификацию с полным набором признаков.

In [7]:

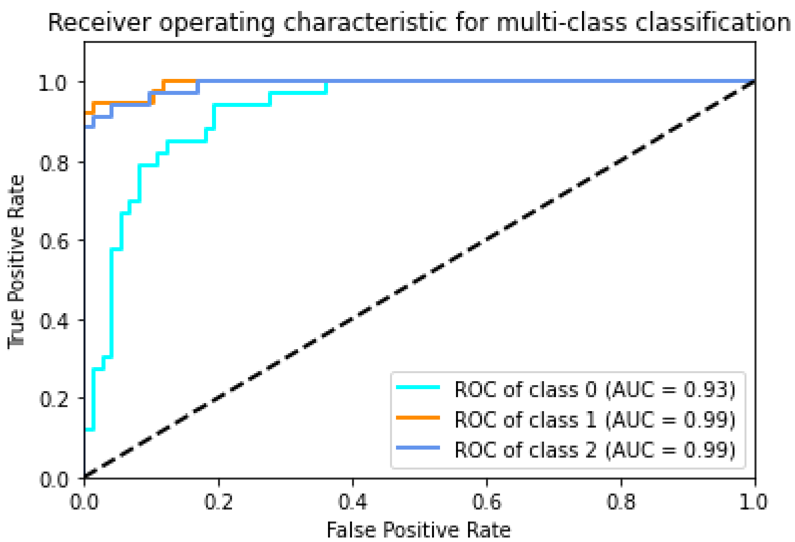
```
def make_clf(drop_cols):
    classes = data["Class"].unique()
    n_classes = len(classes)
    y = label_binarize(data["Class"], classes=classes)
    X = data.drop(columns=["Class"] + drop_cols).copy()
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.5, random_stat
e=25)
    clf = OneVsRestClassifier(SVC(kernel='linear', probability=True, random_state=159))
    y_score = clf.fit(X_train, y_train).decision_function(X_test)

    fpr, tpr, auc = dict(), dict(), dict()
    for i in range(n_classes):
        y_test_cl = y_test[:,i]
        y_score_cl = y_score[:,i]
        fpr[i], tpr[i], _ = roc_curve(y_test_cl, y_score_cl)
        auc[i] = roc_auc_score(y_test_cl, y_score_cl)

    lw = 2
    colors = cycle(['aqua', 'darkorange', 'cornflowerblue'])
    for i, color in zip(range(n_classes), colors):
        plt.plot(fpr[i], tpr[i], color=color, lw=lw, label='ROC of class {0} (AUC = {1:
0.2f})'.format(i, auc[i]))
    plt.plot([0, 1], [0, 1], 'k--', lw=lw)
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.1])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver operating characteristic for multi-class classification')
    plt.legend(loc="lower right")
    plt.show()
```

In [8]:

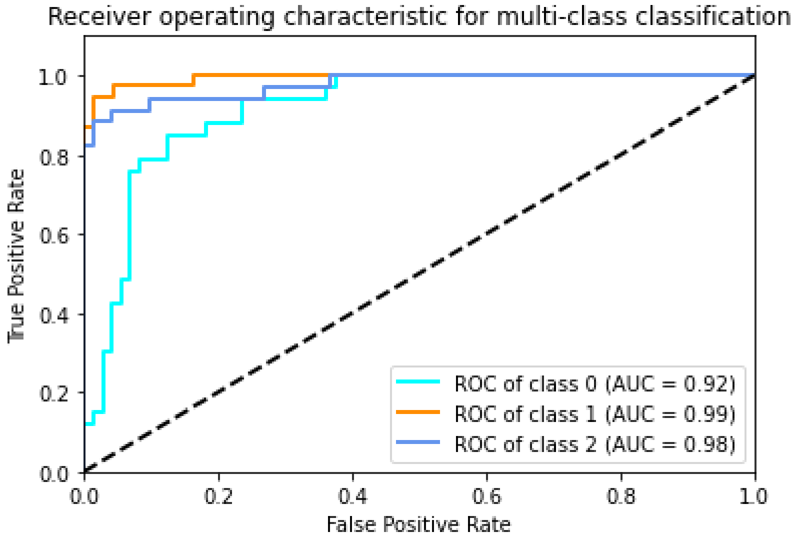
```
make_clf([])
```



По диаграмме рассеяния и тепловой карте видно, что наибольшую корреляцию имеют следующие пары атриутов: *Area* и *Perimeter*, *Area* и *Length of kernel*, *Area* и *Width of kernel*, *Perimeter* и *Length of kernel*, *Perimeter* и *Width of kernel*. Исключим из набора данных атрибуты *Area*, *Perimeter* и *Length of kernel* и выполним классификацию.

In [9]:

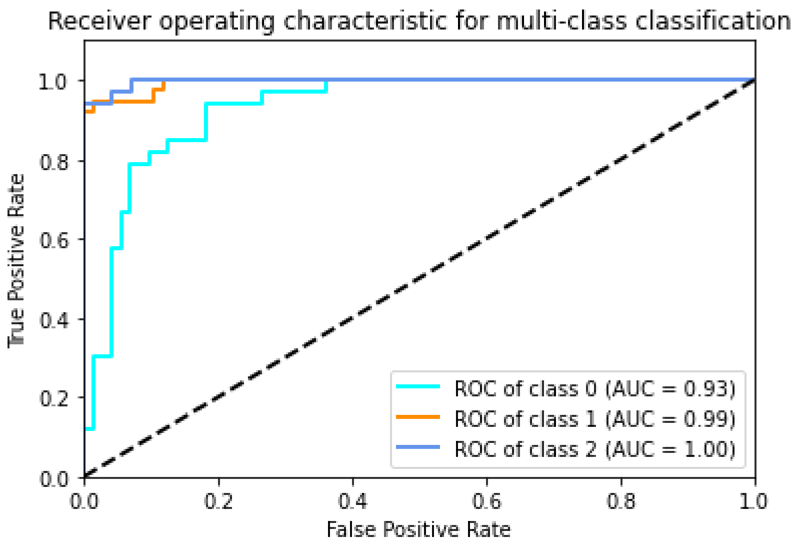
```
make_clf(["Area", "Perimeter", "Length of kernel"])
```



Величины AUC для ROC-крвых уменьшились, что говорит об ухудшении качества классицикации. Вернем в набор атрибут *Perimeter*.

In [10]:

```
make_clf(["Area", "Length of kernel"])
```



Качество классицикации повысилось.