svm_homework

April 11, 2025

1 Worked with:

1.1 Trevor Mathisen

1.2 Viet Nguyen

[1]:
```
from numpy import set_printoptions, logspace, mean, std
import matplotlib.pyplot as plt
import pandas as pd
from pandas import set_option
from pandas import read_csv
from pandas.plotting import scatter_matrix

from sklearn.preprocessing import StandardScaler, Normalizer, LabelEncoder
from sklearn.model_selection import KFold, GridSearchCV, cross_val_score

import seaborn as sns
```

1.3 3. Use the Wine dataset (check under ML dataset Module on Canvas)

[2]:
```
filename = 'wine.csv'
# Column names added to csv including 'Class' as the first column/output␣
↪variable
data = read_csv(filename)
```

1.4 Exploratory Data Analysis

[3]:
```
set_printoptions(precision=3)
print(data.head(5))
print(data.isnull().sum())
print(f'{data.shape[0]} rows and {data.shape[1]} columns')
for column in data.columns:
    print(column)
    print(data[column].unique())
```

```
   Class  Alcohol  Malic acid   Ash  Alcalinity of ash  Magnesium  \
0      1    14.23        1.71  2.43               15.6        127
1      1    13.20        1.78  2.14               11.2        100
```

```
2      1    13.16        2.36  2.67              18.6        101
3      1    14.37        1.95  2.50              16.8        113
4      1    13.24        2.59  2.87              21.0        118

   Total phenols  Flavanoids  Nonflavanoid phenols  Proanthocyanins  \
0           2.80        3.06                  0.28             2.29
1           2.65        2.76                  0.26             1.28
2           2.80        3.24                  0.30             2.81
3           3.85        3.49                  0.24             2.18
4           2.80        2.69                  0.39             1.82

   Color intensity   Hue  OD280/OD315 of diluted wines  Proline
0             5.64  1.04                          3.92     1065
1             4.38  1.05                          3.40     1050
2             5.68  1.03                          3.17     1185
3             7.80  0.86                          3.45     1480
4             4.32  1.04                          2.93      735
Class                           0
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
dtype: int64
178 rows and 14 columns
Class
[1 2 3]
Alcohol
[14.23 13.2  13.16 14.37 13.24 14.2  14.39 14.06 14.83 13.86 14.1  14.12
 13.75 14.75 14.38 13.63 14.3  13.83 14.19 13.64 12.93 13.71 12.85 13.5
 13.05 13.39 13.3  13.87 14.02 13.73 13.58 13.68 13.76 13.51 13.48 13.28
 13.07 14.22 13.56 13.41 13.88 14.21 13.9  13.94 13.82 13.77 13.74 13.29
 13.72 12.37 12.33 12.64 13.67 12.17 13.11 13.34 12.21 12.29 13.49 12.99
 11.96 11.66 13.03 11.84 12.7  12.   12.72 12.08 12.67 12.16 11.65 11.64
 12.69 11.62 12.47 11.81 12.6  12.34 11.82 12.51 12.42 12.25 12.22 11.61
 11.46 12.52 11.76 11.41 11.03 12.77 11.45 11.56 11.87 12.07 12.43 11.79
 12.04 12.86 12.88 12.81 12.53 12.84 13.36 13.52 13.62 12.87 13.32 13.08
 12.79 13.23 12.58 13.17 13.84 12.45 14.34 12.36 13.69 12.96 13.78 13.45
 12.82 13.4  12.2  14.16 13.27 14.13]
Malic acid
```

```
[1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 2.16 1.48 1.73 1.81
 1.92 1.57 1.59 3.1  1.63 3.8  1.86 1.6  2.05 1.77 1.72 1.9  1.68 1.5
 1.66 1.83 1.53 1.8  1.65 3.99 3.84 1.89 3.98 4.04 3.59 2.02 1.75 1.67
 1.7  1.97 1.43 0.94 1.1  1.36 1.25 1.13 1.45 1.21 1.01 1.17 1.19 1.61
 1.51 1.09 1.88 0.9  2.89 0.99 3.87 0.92 3.86 0.89 0.98 2.06 1.33 2.83
 1.99 1.52 2.12 1.41 1.07 3.17 2.08 1.34 2.45 2.55 1.29 3.74 2.43 2.68
 0.74 1.39 1.47 3.43 2.4  4.43 5.8  4.31 2.13 4.3  2.99 2.31 3.55 1.24
 2.46 4.72 5.51 2.96 2.81 2.56 4.95 3.88 3.57 5.04 4.61 3.24 3.9  3.12
 2.67 3.3  5.19 4.12 3.03 3.83 3.26 3.27 3.45 2.76 4.36 3.7  3.37 2.58
 4.6  2.39 2.51 5.65 3.91 4.28 4.1 ]
Ash
[2.43 2.14 2.67 2.5  2.87 2.45 2.61 2.17 2.27 2.3  2.32 2.41 2.39 2.38
 2.7  2.72 2.62 2.48 2.56 2.28 2.65 2.36 2.52 3.22 2.8  2.21 2.84 2.55
 2.1  2.51 2.31 2.12 2.59 2.29 2.44 2.4  2.04 2.6  2.42 2.68 2.25 2.46
 1.36 2.02 1.92 2.16 2.53 1.7  1.75 2.24 1.71 2.23 1.95 2.   2.2  2.58
 2.26 2.22 2.74 1.98 1.9  1.88 1.94 1.82 2.92 1.99 2.19 3.23 2.73 2.13
 2.78 2.54 2.64 2.35 2.15 2.75 2.69 2.86 2.37]
Alcalinity of ash
[15.6 11.2 18.6 16.8 21.  15.2 14.6 17.6 14.  16.  18.  11.4 12.  17.2
 20.  16.5 16.6 17.8 25.  16.1 17.  19.4 22.5 19.1 19.5 19.  20.5 15.5
 13.2 16.2 18.8 15.  17.5 18.9 17.4 12.4 17.1 16.4 16.3 16.7 10.6 18.1
 19.6 20.4 24.  30.  14.8 23.  22.8 26.  21.6 23.6 18.5 22.  20.7 21.5
 20.8 28.5 26.5 24.5 23.5 25.5 27. ]
Magnesium
[127 100 101 113 118 112  96 121  97  98 105  95  89  91 102 120 115 108
 116 126 124  93  94 107 106 104 132 110 128 117  90 103 111  92  88  87
  78 151  86 139 136  85  99  84  70  81  80 162 134 119  82 122 123]
Total phenols
[2.8  2.65 3.85 3.27 2.5  2.6  2.98 2.95 2.2  3.1  3.3  2.85 2.7  3.
 2.41 2.61 2.48 2.53 2.63 2.4  2.86 2.42 2.35 2.45 3.15 3.25 2.64 2.75
 2.88 2.72 3.88 2.96 3.2  3.4  1.98 2.05 2.02 2.1  3.5  1.89 2.11 1.85
 1.1  1.88 3.38 1.61 1.95 1.72 1.9  2.83 2.   1.65 1.78 1.92 1.6  1.45
 1.38 3.02 2.55 3.52 2.23 2.56 1.68 2.36 2.74 3.18 1.75 2.46 1.63 2.9
 2.62 2.13 2.22 1.51 1.3  1.15 1.7  1.62 1.79 2.32 1.54 1.4  1.55 1.5
 0.98 1.93 1.41 1.48 1.8  1.74 2.3  1.83 1.39 1.35 1.28 1.25 1.59]
Flavanoids
[3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 3.32 2.43 3.69 3.64
 2.91 3.14 3.4  3.93 3.03 3.17 2.41 2.88 2.37 2.61 2.68 2.94 2.19 2.97
 2.33 3.25 3.19 2.74 2.53 2.64 3.04 3.29 3.56 2.63 3.   2.65 2.92 3.54
 3.27 2.99 3.74 2.79 2.9  2.78 3.23 3.67 0.57 1.09 1.41 1.79 3.1  1.75
 3.18 2.   1.3  1.28 1.02 2.86 1.84 2.89 2.14 1.57 2.03 1.32 1.85 2.55
 2.26 1.58 1.59 2.21 1.94 1.69 1.61 1.5  1.25 1.46 2.25 2.27 0.99 2.5
 3.75 2.17 1.36 2.11 1.64 1.92 1.76 2.04 2.58 2.01 2.29 1.6  2.09 5.08
 2.13 2.24 2.45 1.22 1.2  0.58 0.66 0.47 0.6  0.48 0.5  0.52 0.8  0.78
 0.55 0.34 0.65 0.76 1.39 0.83 0.63 1.31 1.1  0.92 0.56 0.7  0.68 0.84
 0.96 0.49 0.51 0.61 0.75 0.69]
Nonflavanoid phenols
[0.28 0.26 0.3  0.24 0.39 0.34 0.31 0.29 0.22 0.43 0.33 0.4  0.32 0.17
```

```
 0.25 0.27 0.47 0.37 0.42 0.5  0.2  0.21 0.19 0.63 0.53 0.45 0.55 0.14
 0.13 0.35 0.61 0.48 0.52 0.58 0.66 0.6  0.41 0.44 0.56]
Proanthocyanins
[2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.85 2.38 1.57 1.81 2.96 1.46
 1.72 1.86 1.66 2.1  1.69 1.92 1.45 1.35 1.76 1.95 1.54 1.36 1.44 1.37
 2.08 2.34 1.48 1.7  2.03 2.19 2.14 2.91 1.87 1.68 1.62 2.45 2.04 0.42
 0.41 0.62 0.73 1.03 2.28 1.04 2.5  1.96 1.65 1.15 0.95 2.76 1.43 1.77
 1.4  2.35 1.56 1.34 1.38 1.64 1.63 1.99 3.28 1.31 1.42 2.49 3.58 1.22
 1.05 2.01 1.53 1.61 0.83 1.83 1.71 1.9  0.94 0.84 0.8  1.1  0.88 0.81
 0.75 0.64 0.55 1.02 1.14 1.3  0.68 0.86 1.26 1.55 2.7  0.96 0.97 1.11
 1.24 1.06 1.41]
Color intensity
[ 5.64  4.38  5.68  7.8   4.32  6.75  5.25  5.05  5.2   7.22  5.75  5.
  5.6   5.4   7.5   7.3   6.2   6.6   8.7   5.1   5.65  4.5   3.8   3.93
  3.52  3.58  4.8   3.95  4.7   5.7   6.9   3.84  4.2   4.6   4.25  3.7
  6.13  4.28  5.43  4.36  5.04  5.24  4.9   6.1   8.9   7.2   7.05  6.3
  5.85  6.25  6.38  6.    6.8   1.95  3.27  4.45  2.95  5.3   4.68  3.17
  2.85  3.05  3.38  3.74  3.35  3.21  2.65  3.4   2.57  2.5   3.9   2.2
  2.62  2.45  2.6   2.8   1.74  2.4   3.6   2.15  3.25  2.9   2.3   3.3
  2.06  2.94  2.7   2.    3.08  1.9   1.28  2.08  2.76  3.94  3.    2.12
  4.1   5.45  7.1   3.85  4.92  4.35  4.4   8.21  4.    7.65  8.42  9.4
  8.6  10.8  10.52  7.6   7.9   9.01 13.   11.75  5.88  5.58  5.28  9.58
  6.62 10.68 10.26  8.66  8.5   5.5   9.9   9.7   7.7  10.2   9.3   9.2 ]
Hue
[1.04  1.05  1.03  0.86  1.02  1.06  1.08  1.01  1.25  1.17  1.15  1.2
 1.28  1.07  1.13  1.23  0.96  1.09  1.11  1.12  0.92  1.19  1.1   1.18
 0.89  0.95  0.91  0.88  0.82  0.87  1.24  0.98  0.94  1.22  1.45  0.906
 1.36  1.31  0.99  1.38  1.16  0.84  0.79  1.33  1.    1.42  1.27  0.8
 0.75  0.9   0.93  1.71  0.7   0.73  0.69  0.97  0.76  0.74  0.66  0.78
 0.81  0.77  0.65  0.6   0.58  0.54  0.55  0.57  0.59  0.48  0.61  0.56
 0.67  0.68  0.85  0.72  0.62  0.64 ]
OD280/OD315 of diluted wines
[3.92 3.4  3.17 3.45 2.93 2.85 3.58 3.55 2.82 2.9  2.73 3.   2.88 2.65
 2.57 3.36 3.71 3.52 4.   3.63 3.82 3.2  3.22 2.77 3.59 2.71 2.87 3.47
 2.78 2.51 2.69 3.53 3.38 3.56 3.35 3.33 3.44 2.75 3.1  2.91 3.37 3.26
 3.03 3.31 2.84 1.82 1.67 1.59 2.46 2.23 2.3  3.18 3.48 1.93 3.07 3.16
 3.5  3.13 2.14 2.48 2.52 2.31 3.12 3.14 2.72 2.01 3.08 2.26 3.21 2.27
 2.06 3.3  2.96 2.63 2.74 2.83 2.44 3.57 2.42 3.02 2.81 2.5  3.19 2.12
 3.05 3.39 3.69 3.64 3.28 1.29 1.42 1.36 1.51 1.58 1.27 1.69 2.15 2.47
 2.05 2.   1.68 1.33 1.86 1.62 1.3  1.47 1.55 1.48 1.64 1.73 1.96 1.78
 2.11 1.75 1.56 1.8  1.92 1.83 1.63 1.71 1.74 1.6 ]
Proline
[1065 1050 1185 1480  735 1450 1290 1295 1045 1510 1280 1320 1150 1547
 1310 1130 1680  845  780  770 1035 1015  830 1195 1285  915 1515  990
 1235 1095  920  880 1105 1020  760  795  680  885 1080  985 1060 1260
 1265 1190 1375 1120  970 1270  520  450  630  420  355  678  502  510
  750  718  870  410  472  886  428  392  500  463  278  714  515  495
  562  625  480  290  345  937  660  406  710  438  415  672  315  488
```

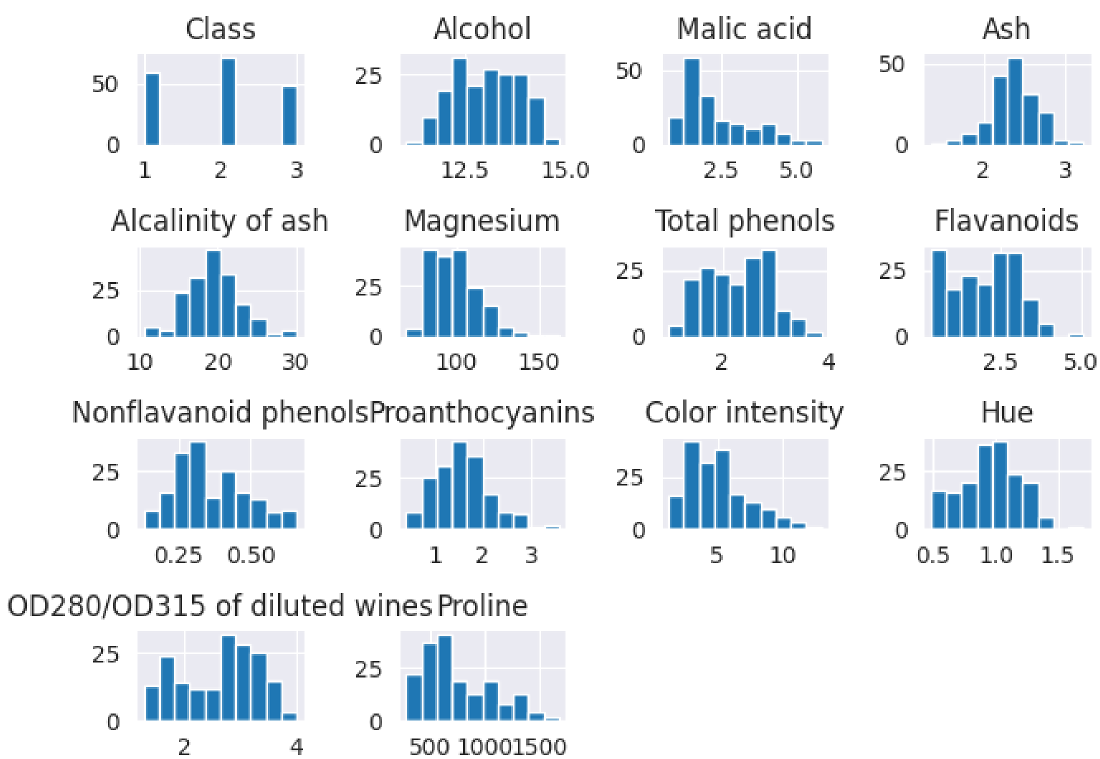
```
 312  325  607  434  385  407  372  564  465  365  380  378  352  466
 342  580  530  560  600  650  695  720  590  550  855  425  675  640
 725  620  570  615  685  470  740  835  840]
```

[4]:
```
def data_info(_data):
    print(_data.describe())
    _data.hist()
    plt.tight_layout()
    plt.show()
    plt.figure() # new plot
    plt.tight_layout()
    corMat = _data.corr(method='pearson')
    print(corMat)
    ## plot correlation matrix as a heat map
    sns.heatmap(corMat, square=True)
    plt.yticks(rotation=0)
    plt.xticks(rotation=90)
    plt.title(f"CORRELATION MATRIX USING HEAT MAP")
    plt.show()

    ## scatter plot of all _data
    plt.figure()
    # # The output overlaps itself, resize it to display better (w padding)
    scatter_matrix(_data)
    plt.tight_layout(pad=0.1)
    plt.show()

data_info(data)
```

```
            Class     Alcohol  Malic acid         Ash  Alcalinity of ash  \
count  178.000000  178.000000  178.000000  178.000000         178.000000
mean     1.938202   13.000618    2.336348    2.366517          19.494944
std      0.775035    0.811827    1.117146    0.274344           3.339564
min      1.000000   11.030000    0.740000    1.360000          10.600000
25%      1.000000   12.362500    1.602500    2.210000          17.200000
50%      2.000000   13.050000    1.865000    2.360000          19.500000
75%      3.000000   13.677500    3.082500    2.557500          21.500000
max      3.000000   14.830000    5.800000    3.230000          30.000000

        Magnesium  Total phenols  Flavanoids  Nonflavanoid phenols  \
count  178.000000     178.000000  178.000000            178.000000
mean    99.741573       2.295112    2.029270              0.361854
std     14.282484       0.625851    0.998859              0.124453
min     70.000000       0.980000    0.340000              0.130000
25%     88.000000       1.742500    1.205000              0.270000
50%     98.000000       2.355000    2.135000              0.340000
75%    107.000000       2.800000    2.875000              0.437500
max    162.000000       3.880000    5.080000              0.660000
```

	Proanthocyanins	Color intensity	Hue	\
count	178.000000	178.000000	178.000000	
mean	1.590899	5.058090	0.957449	
std	0.572359	2.318286	0.228572	
min	0.410000	1.280000	0.480000	
25%	1.250000	3.220000	0.782500	
50%	1.555000	4.690000	0.965000	
75%	1.950000	6.200000	1.120000	
max	3.580000	13.000000	1.710000	

	OD280/OD315 of diluted wines	Proline
count	178.000000	178.000000
mean	2.611685	746.893258
std	0.709990	314.907474
min	1.270000	278.000000
25%	1.937500	500.500000
50%	2.780000	673.500000
75%	3.170000	985.000000
max	4.000000	1680.000000



	Class	Alcohol	Malic acid	Ash	\
Class	1.000000	-0.328222	0.437776	-0.049643	

Alcohol	-0.328222	1.000000	0.094397	0.211545
Malic acid	0.437776	0.094397	1.000000	0.164045
Ash	-0.049643	0.211545	0.164045	1.000000
Alcalinity of ash	0.517859	-0.310235	0.288500	0.443367
Magnesium	-0.209179	0.270798	-0.054575	0.286587
Total phenols	-0.719163	0.289101	-0.335167	0.128980
Flavanoids	-0.847498	0.236815	-0.411007	0.115077
Nonflavanoid phenols	0.489109	-0.155929	0.292977	0.186230
Proanthocyanins	-0.499130	0.136698	-0.220746	0.009652
Color intensity	0.265668	0.546364	0.248985	0.258887
Hue	-0.617369	-0.071747	-0.561296	-0.074667
OD280/OD315 of diluted wines	-0.788230	0.072343	-0.368710	0.003911
Proline	-0.633717	0.643720	-0.192011	0.223626

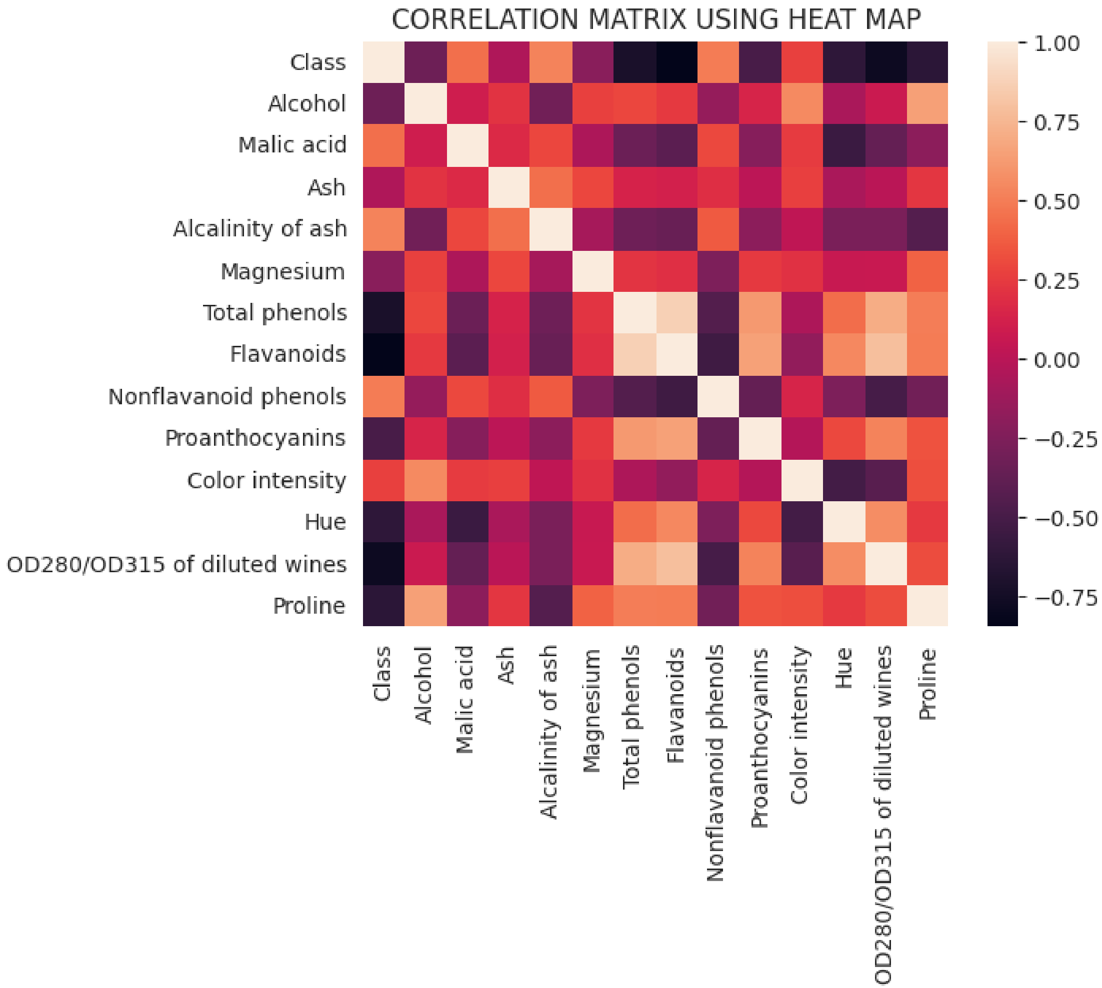
	Alcalinity of ash	Magnesium	Total phenols \
Class	0.517859	-0.209179	-0.719163
Alcohol	-0.310235	0.270798	0.289101
Malic acid	0.288500	-0.054575	-0.335167
Ash	0.443367	0.286587	0.128980
Alcalinity of ash	1.000000	-0.083333	-0.321113
Magnesium	-0.083333	1.000000	0.214401
Total phenols	-0.321113	0.214401	1.000000
Flavanoids	-0.351370	0.195784	0.864564
Nonflavanoid phenols	0.361922	-0.256294	-0.449935
Proanthocyanins	-0.197327	0.236441	0.612413
Color intensity	0.018732	0.199950	-0.055136
Hue	-0.273955	0.055398	0.433681
OD280/OD315 of diluted wines	-0.276769	0.066004	0.699949
Proline	-0.440597	0.393351	0.498115

	Flavanoids	Nonflavanoid phenols \
Class	-0.847498	0.489109
Alcohol	0.236815	-0.155929
Malic acid	-0.411007	0.292977
Ash	0.115077	0.186230
Alcalinity of ash	-0.351370	0.361922
Magnesium	0.195784	-0.256294
Total phenols	0.864564	-0.449935
Flavanoids	1.000000	-0.537900
Nonflavanoid phenols	-0.537900	1.000000
Proanthocyanins	0.652692	-0.365845
Color intensity	-0.172379	0.139057
Hue	0.543479	-0.262640
OD280/OD315 of diluted wines	0.787194	-0.503270
Proline	0.494193	-0.311385

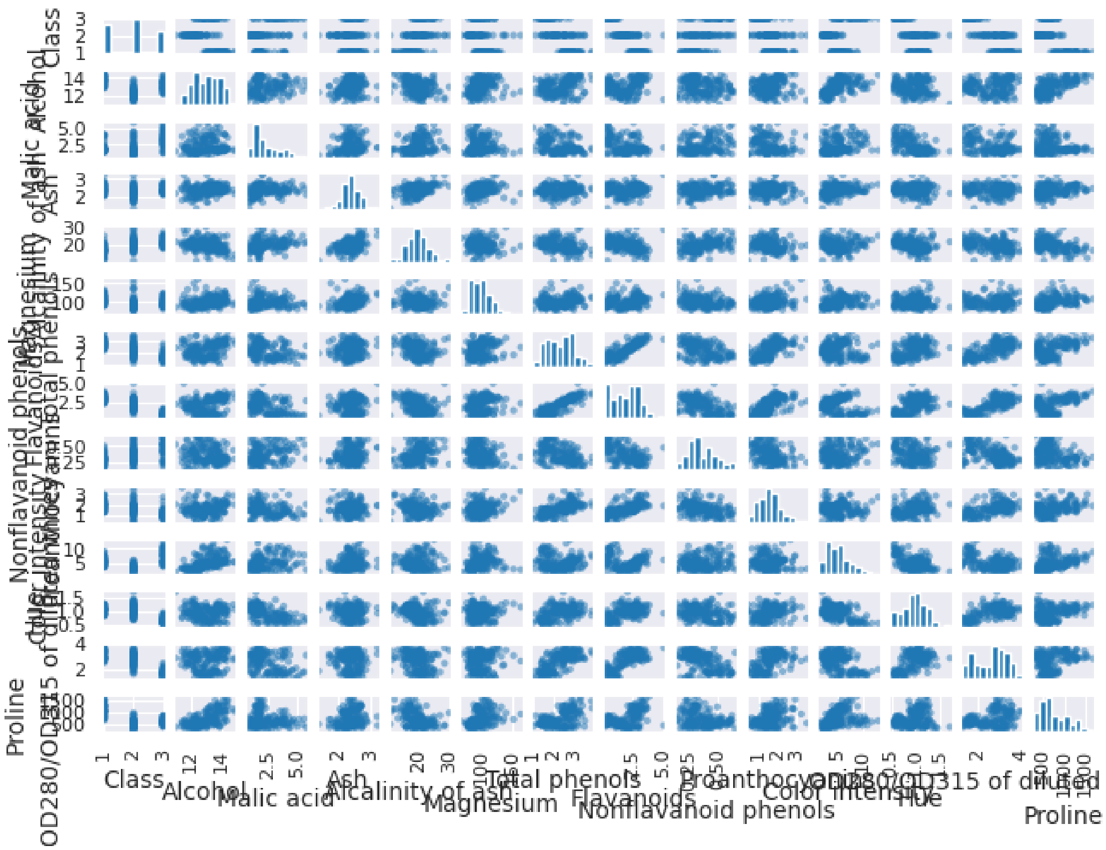
	Proanthocyanins	Color intensity	Hue \
Class	-0.499130	0.265668	-0.617369

Alcohol	0.136698	0.546364	-0.071747
Malic acid	-0.220746	0.248985	-0.561296
Ash	0.009652	0.258887	-0.074667
Alcalinity of ash	-0.197327	0.018732	-0.273955
Magnesium	0.236441	0.199950	0.055398
Total phenols	0.612413	-0.055136	0.433681
Flavanoids	0.652692	-0.172379	0.543479
Nonflavanoid phenols	-0.365845	0.139057	-0.262640
Proanthocyanins	1.000000	-0.025250	0.295544
Color intensity	-0.025250	1.000000	-0.521813
Hue	0.295544	-0.521813	1.000000
OD280/OD315 of diluted wines	0.519067	-0.428815	0.565468
Proline	0.330417	0.316100	0.236183

	OD280/OD315 of diluted wines	Proline
Class	-0.788230	-0.633717
Alcohol	0.072343	0.643720
Malic acid	-0.368710	-0.192011
Ash	0.003911	0.223626
Alcalinity of ash	-0.276769	-0.440597
Magnesium	0.066004	0.393351
Total phenols	0.699949	0.498115
Flavanoids	0.787194	0.494193
Nonflavanoid phenols	-0.503270	-0.311385
Proanthocyanins	0.519067	0.330417
Color intensity	-0.428815	0.316100
Hue	0.565468	0.236183
OD280/OD315 of diluted wines	1.000000	0.312761
Proline	0.312761	1.000000




```
<Figure size 640x480 with 0 Axes>
```



1.5 Standardize the data as all are continuous variables and appear mostly guassian

[5]:
```
Y1 = data['Class']
data_stand = data.copy()
X1_stand = data.drop(columns=['Class'])
X1_stand = StandardScaler().fit(X1_stand).transform(X1_stand)
```

[6]:
```
from matplotlib import pyplot

def evaluate_each_model_in_turn(models, X, Y):
    results = []
    names = []
    scoring = 'accuracy'
    for name, model in models:
        kfold = KFold(n_splits=10, random_state=7, shuffle=True)
        cv_results = cross_val_score(model, X, Y, cv=kfold, scoring=scoring)
        results.append(cv_results)
        names.append(name)
        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
```

```
        print(msg)
    ## boxplot algorithm comparison
    fig = pyplot.figure()
    fig.suptitle('Algorithm Comparison')
    ax = fig.add_subplot(111)
    pyplot.boxplot(results)
    ax.set_xticklabels(names)
    pyplot.show()
```

1.6 3.

1.7 add an SV Classifier(SVC: use the default settings given in the sample code, use RFB kernel with C = 1.0)

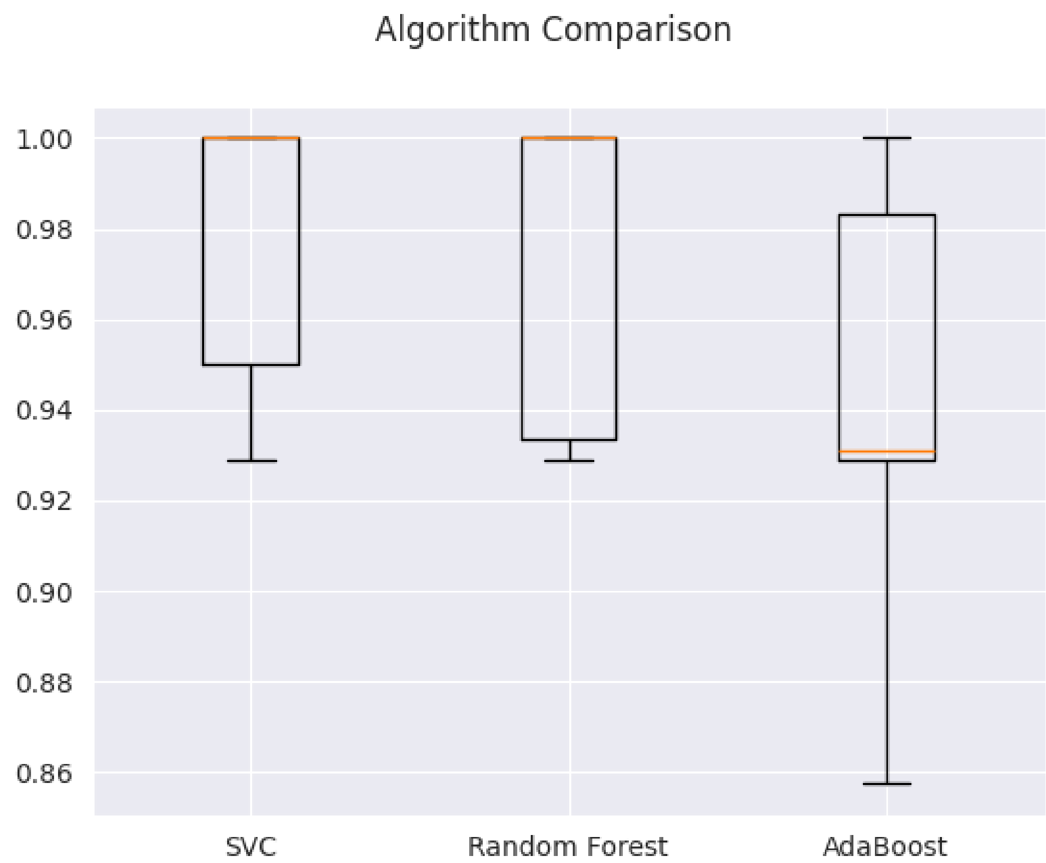
1.8 a random forest classifier with a depth of 2

1.9 and an Adaboost classifier

1.10 and compare them using kfold cross validation with k=10.

[7]:
```
# add an SV Classifier(SVC), a random forest classifier with a depth of 2 and␣
↪an Adaboost
# classifier and compare them using kfold cross validation with k=10. For the␣
↪SVC, use
# the default settings given in the sample code, use RFB kernel with C = 1.0
models = []
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.svm import SVC
svc_model = ('SVC', SVC(kernel='rbf', C=1.0, random_state=1, probability=True))
models.append(svc_model)
models.append(('Random Forest', RandomForestClassifier(max_depth=2,␣
↪random_state=1)))
models.append(('AdaBoost', AdaBoostClassifier(n_estimators=100,␣
↪random_state=1)))

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X1_stand, Y1, test_size=0.2)

evaluate_each_model_in_turn(models, X_train, Y_train)
```

```
SVC: 0.979048 (0.032029)
Random Forest: 0.972381 (0.033860)
AdaBoost: 0.936667 (0.049900)
```



1.11 4. Plot all the accuracy results vs. each model (model type on the x-axis and accuracy on the y-axis)

1.11.1 Results:

1.11.2 1. The SVC with RBF kernel has the highest accuracy.

1.11.3 2. The AdaBoost classifier has the lowest accuracy.

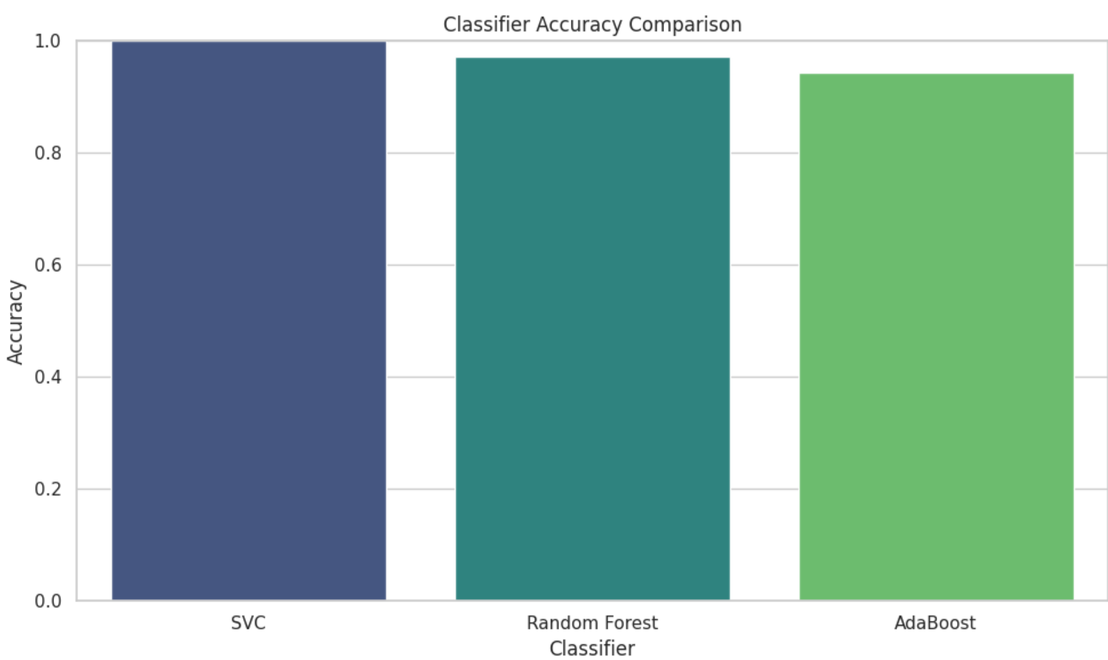
1.11.4 3. SVC also has the lowest standard deviation, indicating more stable performance.

1.11.5 4. In all runs, SVC continuously outperformed the other classifiers.

[8]:
```
accuracy_dict = {}
for name, model in models:
    model.fit(X_train, Y_train)
    y_pred = model.predict(X_test)
    accuracy = model.score(X_test, Y_test)
    accuracy_dict[name] = accuracy
    print(f"{name} Accuracy: {accuracy:.4f}")
```

```
# Plotting the accuracies
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Set the style
sns.set_theme(style="whitegrid")
methods = list(accuracy_dict.keys())
accuracies = list(accuracy_dict.values())

plt.figure(figsize=(10, 6))
sns.barplot(x=methods, y=accuracies, hue=methods, palette="viridis",␣
 ↪legend=False)
plt.title('Classifier Accuracy Comparison')
plt.xlabel('Classifier')
plt.ylabel('Accuracy')
plt.ylim(0, 1) # Accuracy domain
plt.tight_layout()
plt.show()
```

```
SVC Accuracy: 1.0000
Random Forest Accuracy: 0.9722
AdaBoost Accuracy: 0.9444
```



1.12 5. Try a polynomial kernel by setting kernel = 'poly' and change the kernel degree from 2 – 5.

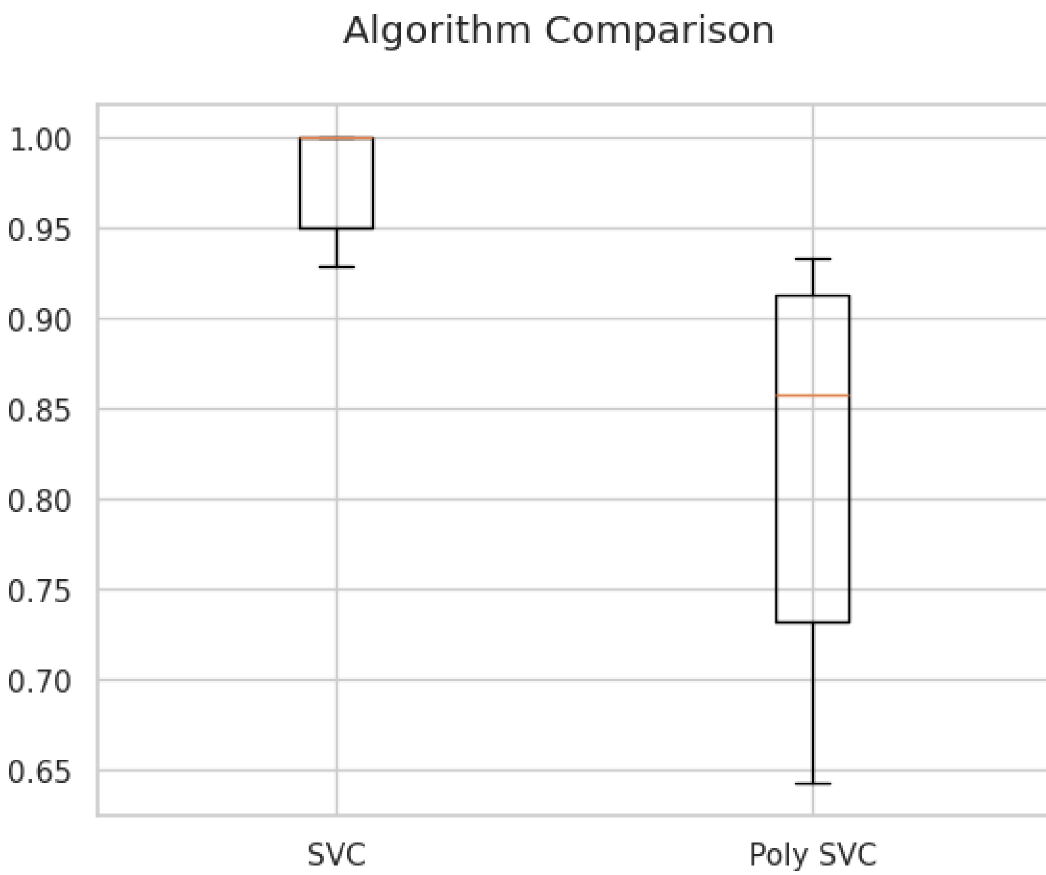
1.13 6. Compare the results with the RBF kernel and the same value of C=1.0

1.13.1 Results:

1.13.2 The RBF kernel significantly outperforms the polynomial kernel in accuracy. The polynomial kernel has both lower accuracy and higher variance in cross-validation scores, indicating less robust performance across different data subsets.

[9]:
```
poly_svc_model = SVC(kernel='poly', degree=5, C=1.0, random_state=1,␣
 ↪probability=True)
# fit and append
poly_svc_model.fit(X_train, Y_train)
y_pred_poly = poly_svc_model.predict(X_test)
accuracy_poly = poly_svc_model.score(X_test, Y_test)
poly_svc = ('Poly SVC', poly_svc_model)
accuracy_dict[poly_svc[0]] = accuracy_poly
models.append(poly_svc)


compare_svc_models = [svc_model, poly_svc]
evaluate_each_model_in_turn(compare_svc_models, X_train, Y_train)
```

```
SVC: 0.979048 (0.032029)
Poly SVC: 0.815714 (0.107835)
```



Algorithm Comparison
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
SVC
Poly SVC

1.14 7. Write down your observation on the comparison results.

1.14.1 Results:

1.14.2 1. The RBF kernel significantly outperforms the polynomial kernel in accuracy.

1.14.3 2. The polynomial kernel has both lower accuracy and higher variance in cross-validation scores, indicating less robust performance across different data subsets.

1.14.4 3. Despite the accuracy differences, the polynomial kernel still achieves excellent ROC AUC, suggesting good ranking capabilities even when making some classification errors.

1.15 8. Plot the multi-class ROC curve and use the roc_auc_score function to calculate ROC score.

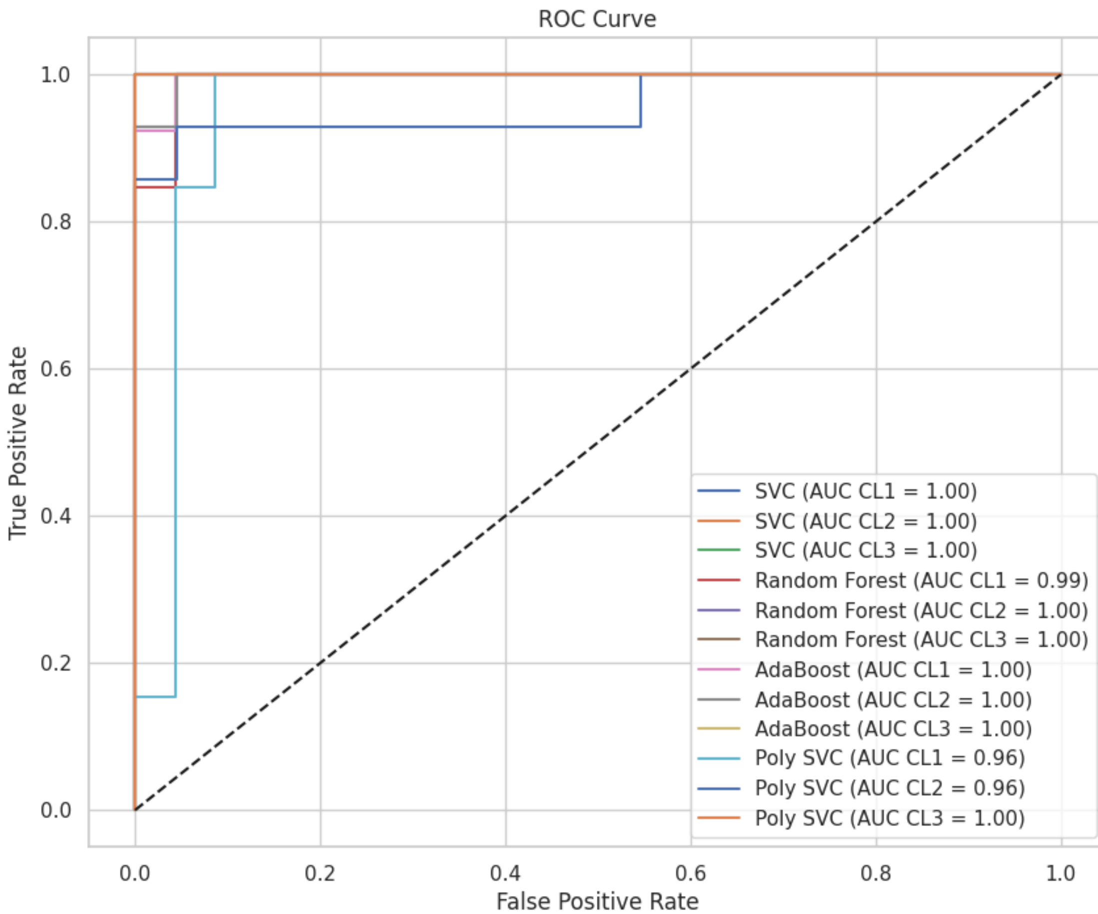
1.15.1 Results:

1.15.2 1. Both SVC and Random Forest achieve perfect ROC AUC scores despite Random Forest having lower accuracy. This suggests Random Forest still ranks predictions correctly even when it makes some misclassifications.

1.15.3 2. AdaBoost performs slightly worse but still shows excellent performance

[12]:
```
# Plot the multi-class ROC curve and use the roc_auc_score function to
↪calculate ROC
# AUC for each class.
from sklearn.metrics import roc_curve, roc_auc_score

# Now you can calculate and plot ROC curves
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
import numpy as np

# For multi-class problems, you need to binarize the output
y_test_bin = label_binarize(Y_test, classes=np.unique(Y_test))
n_classes = y_test_bin.shape[1]

# Plot ROC curves
plt.figure(figsize=(10, 8))

for name, model in models:
    # Getting probabilities
    y_score = model.predict_proba(X_test)

    # Compute ROC curve and ROC area for each class
    fpr = dict()
    tpr = dict()
    roc_auc = dict()
```

```
    for i in range(n_classes):
        fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_score[:, i])
        roc_auc[i] = auc(fpr[i], tpr[i])

    # Plot ROC for the first class as an example
    # plt.plot(fpr[0], tpr[0], label=f'{name} (AUC = {roc_auc[0]:.2f})')
    # Plot ROC for all classes
    plt.plot(fpr[0], tpr[0], label=f'{name} (AUC CL1 = {roc_auc[0]:.2f})')
    plt.plot(fpr[1], tpr[1], label=f'{name} (AUC CL2 = {roc_auc[1]:.2f})')
    plt.plot(fpr[2], tpr[2], label=f'{name} (AUC CL3 = {roc_auc[2]:.2f})')

plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc='lower right')
plt.show()

# Calculate micro-average ROC curve and ROC area (will average performance␣
 ↪across all classes)
for name, model in models:
    y_score = model.predict_proba(X_test)
    y_pred = model.predict(X_test)

    # Calculate and print micro-average ROC AUC score
    print(f"{name} - Micro-average ROC AUC: {roc_auc_score(Y_test, y_score,␣
 ↪multi_class='ovo'):.4f}")
```




```
SVC - Micro-average ROC AUC: 1.0000
Random Forest - Micro-average ROC AUC: 0.9973
AdaBoost - Micro-average ROC AUC: 0.9982
Poly SVC - Micro-average ROC AUC: 0.9746
```