Assignment 9

September 21, 2021

1 ASSIGNMENT 9

2 SIRSS2276

3 SOUMYA GITE

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib as mpl
%matplotlib inline
```

[3]:
```
df1 = pd.read_csv('train.csv')
df2 = pd.read_csv('test.csv')
```

[4]:
```
df1.head()
```

[4]:
```
   id                species   margin1   margin2   margin3   margin4  \
0   1            Acer_Opalus  0.007812  0.023438  0.023438  0.003906
1   2  Pterocarya_Stenoptera  0.005859  0.000000  0.031250  0.015625
2   3   Quercus_Hartwissiana  0.005859  0.009766  0.019531  0.007812
3   5        Tilia_Tomentosa  0.000000  0.003906  0.023438  0.005859
4   6     Quercus_Variabilis  0.005859  0.003906  0.048828  0.009766

    margin5   margin6   margin7  margin8  …  texture55  texture56  \
0  0.011719  0.009766  0.027344      0.0  …   0.007812   0.000000
1  0.025391  0.001953  0.019531      0.0  …   0.000977   0.000000
2  0.003906  0.005859  0.068359      0.0  …   0.154300   0.000000
3  0.021484  0.019531  0.023438      0.0  …   0.000000   0.000977
4  0.013672  0.015625  0.005859      0.0  …   0.096680   0.000000

   texture57  texture58  texture59  texture60  texture61  texture62  \
0   0.002930   0.002930   0.035156        0.0        0.0   0.004883
1   0.000000   0.000977   0.023438        0.0        0.0   0.000977
2   0.005859   0.000977   0.007812        0.0        0.0   0.000000
3   0.000000   0.000000   0.020508        0.0        0.0   0.017578
```

```
4   0.021484   0.000000   0.000000        0.0        0.0   0.000000

   texture63  texture64
0   0.000000   0.025391
1   0.039062   0.022461
2   0.020508   0.002930
3   0.000000   0.047852
4   0.000000   0.031250

[5 rows x 194 columns]
```

[5]:
```
df2.head()
```

[5]:
```
   id   margin1   margin2   margin3   margin4   margin5   margin6   margin7  \
0   4  0.019531  0.009766  0.078125  0.011719  0.003906  0.015625  0.005859
1   7  0.007812  0.005859  0.064453  0.009766  0.003906  0.013672  0.007812
2   9  0.000000  0.000000  0.001953  0.021484  0.041016  0.000000  0.023438
3  12  0.000000  0.000000  0.009766  0.011719  0.017578  0.000000  0.003906
4  13  0.001953  0.000000  0.015625  0.009766  0.039062  0.000000  0.009766

   margin8   margin9  …  texture55  texture56  texture57  texture58  \
0      0.0  0.005859  …   0.006836   0.000000   0.015625   0.000977
1      0.0  0.033203  …   0.000000   0.000000   0.006836   0.001953
2      0.0  0.011719  …   0.128910   0.000000   0.000977   0.000000
3      0.0  0.003906  …   0.012695   0.015625   0.002930   0.036133
4      0.0  0.005859  …   0.000000   0.042969   0.016602   0.010742

   texture59  texture60  texture61  texture62  texture63  texture64
0   0.015625        0.0        0.0   0.000000   0.003906   0.053711
1   0.013672        0.0        0.0   0.000977   0.037109   0.044922
2   0.000000        0.0        0.0   0.015625   0.000000   0.000000
3   0.013672        0.0        0.0   0.089844   0.000000   0.008789
4   0.041016        0.0        0.0   0.007812   0.009766   0.007812

[5 rows x 193 columns]
```

[6]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 990 entries, 0 to 989
Columns: 194 entries, id to texture64
dtypes: float64(192), int64(1), object(1)
memory usage: 1.5+ MB
```

[7]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594 entries, 0 to 593
```

```
Columns: 193 entries, id to texture64
dtypes: float64(192), int64(1)
memory usage: 895.8 KB
```

[8]:
```
print(df1.shape)
print(df2.shape)
```

```
(990, 194)
(594, 193)
```

[9]:
```
df1.duplicated().sum()
```

[9]:
```
0
```

[10]:
```
df1.describe()
```

[10]:
```
                id     margin1     margin2     margin3     margin4  \
count   990.000000  990.000000  990.000000  990.000000  990.000000
mean    799.595960    0.017412    0.028539    0.031988    0.023280
std     452.477568    0.019739    0.038855    0.025847    0.028411
min       1.000000    0.000000    0.000000    0.000000    0.000000
25%     415.250000    0.001953    0.001953    0.013672    0.005859
50%     802.500000    0.009766    0.011719    0.025391    0.013672
75%    1195.500000    0.025391    0.041016    0.044922    0.029297
max    1584.000000    0.087891    0.205080    0.156250    0.169920

          margin5     margin6     margin7     margin8     margin9  …  \
count  990.000000  990.000000  990.000000  990.000000  990.000000  …
mean     0.014264    0.038579    0.019202    0.001083    0.007167  …
std      0.018390    0.052030    0.017511    0.002743    0.008933  …
min      0.000000    0.000000    0.000000    0.000000    0.000000  …
25%      0.001953    0.000000    0.005859    0.000000    0.001953  …
50%      0.007812    0.015625    0.015625    0.000000    0.005859  …
75%      0.017578    0.056153    0.029297    0.000000    0.007812  …
max      0.111330    0.310550    0.091797    0.031250    0.076172  …

        texture55   texture56   texture57   texture58   texture59   texture60  \
count  990.000000  990.000000  990.000000  990.000000  990.000000  990.000000
mean     0.036501    0.005024    0.015944    0.011586    0.016108    0.014017
std      0.063403    0.019321    0.023214    0.025040    0.015335    0.060151
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000    0.000000    0.000977    0.000000    0.004883    0.000000
50%      0.004883    0.000000    0.005859    0.000977    0.012695    0.000000
75%      0.043701    0.000000    0.022217    0.009766    0.021484    0.000000
max      0.429690    0.202150    0.172850    0.200200    0.106450    0.578130

        texture61   texture62   texture63   texture64
count  990.000000  990.000000  990.000000  990.000000
```

```
mean      0.002688    0.020291    0.008989    0.019420
std       0.011415    0.039040    0.013791    0.022768
min       0.000000    0.000000    0.000000    0.000000
25%       0.000000    0.000000    0.000000    0.000977
50%       0.000000    0.003906    0.002930    0.011719
75%       0.000000    0.023438    0.012695    0.029297
max       0.151370    0.375980    0.086914    0.141600

[8 rows x 193 columns]
```

[11]:
```
df2.describe()
```

[11]:
```
                id     margin1     margin2     margin3     margin4  \
count   594.000000  594.000000  594.000000  594.000000  594.000000
mean    780.673401    0.017562    0.028425    0.031858    0.022556
std     465.646977    0.019585    0.038351    0.025719    0.028797
min       4.000000    0.000000    0.000000    0.000000    0.000000
25%     368.500000    0.001953    0.001953    0.013672    0.005859
50%     774.000000    0.009766    0.010743    0.023438    0.013672
75%    1184.500000    0.028809    0.041016    0.042969    0.027344
max    1583.000000    0.085938    0.189450    0.167970    0.164060

          margin5     margin6     margin7     margin8     margin9  …  \
count  594.000000  594.000000  594.000000  594.000000  594.000000  …
mean     0.014527    0.037497    0.019222    0.001085    0.007092  …
std      0.018029    0.051372    0.017122    0.002697    0.009515  …
min      0.000000    0.000000    0.000000    0.000000    0.000000  …
25%      0.001953    0.000000    0.005859    0.000000    0.001953  …
50%      0.007812    0.013672    0.015625    0.000000    0.005859  …
75%      0.019531    0.056641    0.029297    0.000000    0.007812  …
max      0.093750    0.271480    0.087891    0.021484    0.083984  …

        texture55   texture56   texture57   texture58   texture59   texture60  \
count  594.000000  594.000000  594.000000  594.000000  594.000000  594.000000
mean     0.035291    0.005923    0.015033    0.011762    0.015881    0.011217
std      0.064482    0.026934    0.022318    0.024771    0.014898    0.052530
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000    0.000000    0.000977    0.000000    0.004883    0.000000
50%      0.003906    0.000000    0.005859    0.001953    0.012695    0.000000
75%      0.038086    0.000000    0.019531    0.010498    0.022461    0.000000
max      0.353520    0.441410    0.153320    0.177730    0.083984    0.606450

        texture61   texture62   texture63   texture64
count  594.000000  594.000000  594.000000  594.000000
mean     0.002617    0.019975    0.009389    0.020970
std      0.011204    0.034704    0.013457    0.023407
min      0.000000    0.000000    0.000000    0.000000
```

```
25%      0.000000    0.000000    0.000000    0.000977
50%      0.000000    0.003418    0.002930    0.013184
75%      0.000000    0.022461    0.014648    0.032227
max      0.123050    0.247070    0.086914    0.149410

[8 rows x 193 columns]
```

[12]:
```
df1.isnull().any().sum()
```

[12]:
```
0
```

[13]:
```
df1.isnull().any().sum()
```

[13]:
```
0
```

[14]:
```
df1.columns
```

[14]:
```
Index(['id', 'species', 'margin1', 'margin2', 'margin3', 'margin4', 'margin5',
       'margin6', 'margin7', 'margin8',
       …
       'texture55', 'texture56', 'texture57', 'texture58', 'texture59',
       'texture60', 'texture61', 'texture62', 'texture63', 'texture64'],
      dtype='object', length=194)
```

[15]:
```
df1['species'].nunique()
```

[15]:
```
99
```

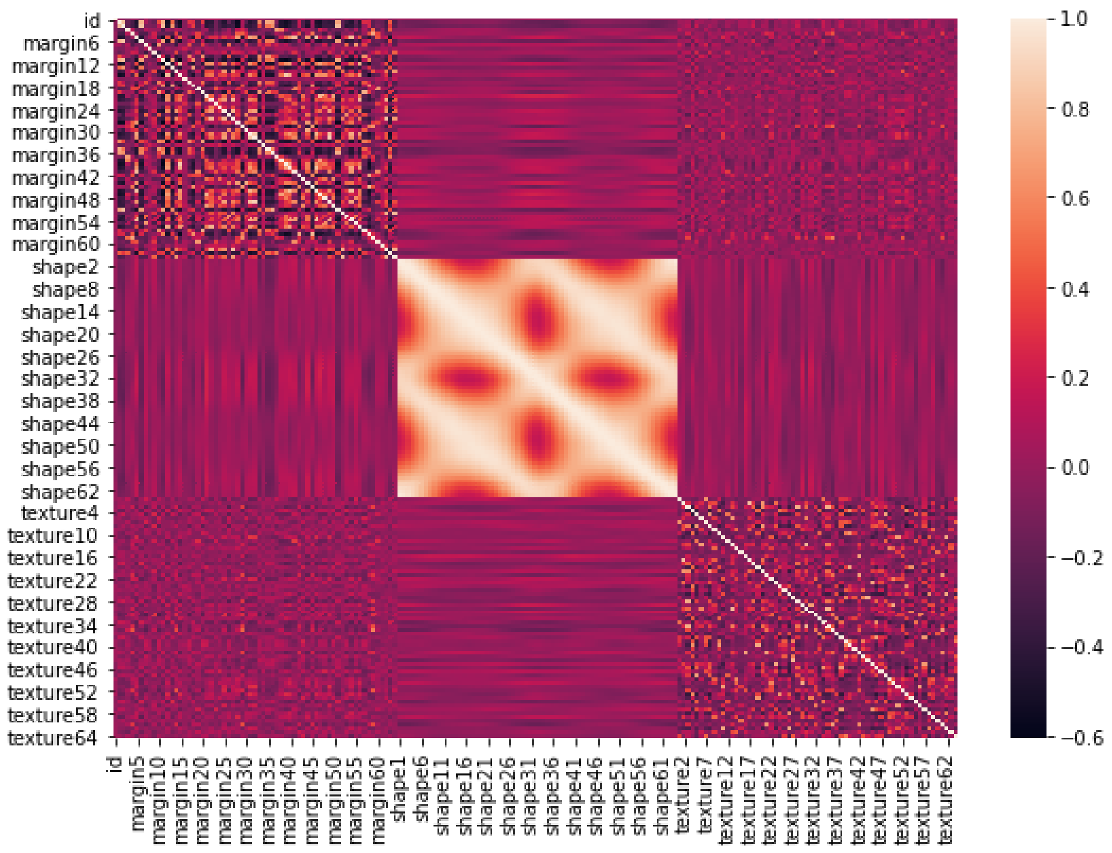
[16]:
```
df1.corr()
```

[16]:
```
                 id   margin1   margin2   margin3   margin4   margin5  \
id         1.000000 -0.011673 -0.027565 -0.059533  0.001639 -0.002419
margin1   -0.011673  1.000000  0.806390 -0.182829 -0.297807 -0.475874
margin2   -0.027565  0.806390  1.000000 -0.204640 -0.315953 -0.444312
margin3   -0.059533 -0.182829 -0.204640  1.000000  0.120042 -0.185007
margin4    0.001639 -0.297807 -0.315953  0.120042  1.000000  0.029480
…               …         …         …         …         …         …
texture60 -0.000823  0.035072  0.081069 -0.019850 -0.052317  0.006542
texture61  0.026319 -0.007581 -0.007057  0.084957  0.320644 -0.109229
texture62  0.032873 -0.033159 -0.037405 -0.081999 -0.073886  0.151675
texture63  0.024299 -0.075171 -0.098957 -0.148193  0.050970  0.022299
texture64  0.035396  0.030414 -0.029532  0.061780  0.014343 -0.148834

            margin6   margin7   margin8   margin9  …  texture55  texture56  \
id        -0.051818  0.061214 -0.039509 -0.070954  …  -0.040292  -0.005132
margin1    0.767718  0.066273 -0.094137 -0.181496  …   0.137158  -0.047771
margin2    0.825762 -0.083273 -0.086428 -0.120276  …   0.154407  -0.021096
margin3   -0.163976  0.095449  0.024350 -0.000042  …   0.047347  -0.027618
```

```
margin4   -0.261437 -0.268271 -0.047693  0.227543  …  -0.071974  -0.009537
…               …        …        …        …  …         …          …
texture60  0.066262 -0.034094  0.048647 -0.028292  …  -0.129365   0.004412
texture61 -0.050498 -0.163375 -0.079283  0.088517  …  -0.002235   0.053707
texture62 -0.031555  0.015391 -0.048843 -0.031954  …  -0.217239   0.171577
texture63 -0.132087 -0.001364  0.027758 -0.119494  …  -0.207887   0.002057
texture64 -0.003164  0.068512 -0.003191 -0.097760  …  -0.095205  -0.095913

           texture57  texture58  texture59  texture60  texture61  texture62  \
id         -0.043101   0.063337  -0.007915  -0.000823   0.026319   0.032873
margin1     0.126227  -0.024139  -0.168201   0.035072  -0.007581  -0.033159
margin2     0.123834  -0.063654  -0.157842   0.081069  -0.007057  -0.037405
margin3     0.007261  -0.021390   0.033505  -0.019850   0.084957  -0.081999
margin4    -0.050529  -0.044318   0.088857  -0.052317   0.320644  -0.073886
…                …         …         …         …         …         …
texture60  -0.155187   0.240704  -0.183369   1.000000  -0.051838   0.265879
texture61  -0.072814  -0.084638  -0.023539  -0.051838   1.000000  -0.063582
texture62  -0.283316   0.563088  -0.128010   0.265879  -0.063582   1.000000
texture63  -0.064724  -0.059866   0.156568  -0.089679  -0.068065  -0.058189
texture64   0.224686  -0.269157  -0.015374  -0.190194   0.036374  -0.245527

           texture63  texture64
id          0.024299   0.035396
margin1    -0.075171   0.030414
margin2    -0.098957  -0.029532
margin3    -0.148193   0.061780
margin4     0.050970   0.014343
…                …         …
texture60  -0.089679  -0.190194
texture61  -0.068065   0.036374
texture62  -0.058189  -0.245527
texture63   1.000000   0.029305
texture64   0.029305   1.000000

[193 rows x 193 columns]
```

[17]:
```
plt.figure(figsize=(10,7))
sns.heatmap(df1.corr())
```

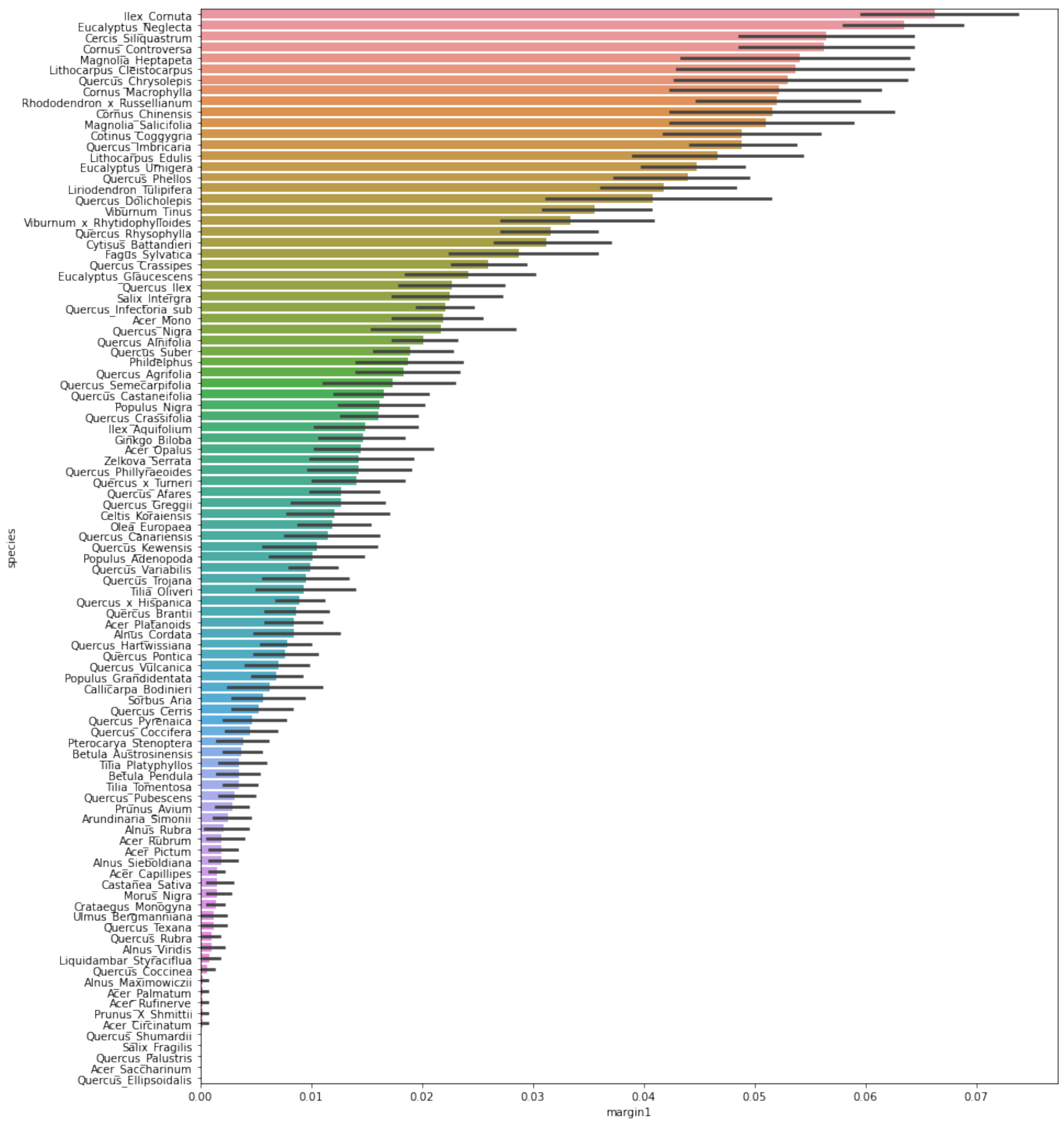
[17]:
```
<AxesSubplot:>
```



```
[18]: plt.figure(figsize = (14,18))
      order = df1.groupby(['species']).mean().sort_values('margin1', ascending =␣
       ↪False).index
      sns.barplot(df1['margin1'],df1['species'], order = order)
```

```
C:\Users\SUDHAKAR\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

```
[18]: <AxesSubplot:xlabel='margin1', ylabel='species'>
```



[19]:
```
test_ids = df2.pop('id')
```

[20]:
```
x = df1.drop(['species','id'],axis=1).values
```

[21]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder().fit(df1['species'])
y = encoder.transform(df1['species'])
y
```

[21]:
```
array([ 3, 49, 65, 94, 84, 40, 54, 78, 53, 89, 98, 16, 74, 50, 58, 31, 43,
        4, 75, 44, 83, 84, 13, 66, 15,  6, 73, 22, 73, 31, 36, 27, 94, 88,
12, 28, 21, 25, 20, 60, 84, 65, 69, 58, 23, 76, 18, 52, 54,  9, 48,
47, 64, 81, 83, 36, 58, 21, 81, 20, 62, 88, 34, 92, 79, 82, 20, 32,
 4, 84, 36, 35, 72, 60, 71, 72, 52, 50, 54, 11, 51, 18, 47,  5,  8,
37, 97, 20, 33,  1, 59,  1, 56,  1,  9, 57, 20, 79, 29, 16, 32, 54,
93, 10, 46, 59, 84, 76, 15, 10, 15,  0, 69,  4, 51, 51, 94, 36, 39,
62,  2, 24, 26, 35, 25, 87,  0, 55, 34, 38,  1, 45,  7, 93, 56, 38,
21, 51, 75, 81, 74, 33, 20, 37,  9, 40, 60, 31, 83, 50, 71, 67, 30,
66,  1, 43, 61, 23, 65, 84, 87, 46, 57, 16,  2, 28, 12, 96, 44, 76,
29, 75, 41, 87, 67, 61, 30,  5, 12, 62,  3, 83, 81,  6, 85,  4, 37,
57, 84, 39, 71, 61,  6, 76, 14, 31, 98, 40, 17, 51, 16, 42, 63, 86,
37, 69, 86, 71, 80, 78, 14, 35, 25,  5, 39,  8,  9, 26, 44, 60, 13,
14, 77, 13, 80, 87, 18, 60, 78, 92, 51, 45, 78, 41, 51, 30, 14, 35,
46, 21,  8,  6, 92, 38, 40, 15, 32, 17, 93, 71, 92, 27, 78, 15, 19,
60, 21, 38, 36, 49, 74, 67, 95, 31, 82, 45, 16, 83, 63, 80, 42, 22,
74, 53, 15, 44, 47, 57, 94, 76, 17, 32, 24, 15, 93, 24, 80, 59, 46,
12, 51, 77, 79, 70, 69, 16,  2, 63, 83, 55, 12, 53,  1, 67,  0,  2,
36, 42, 10,  9, 52, 59,  6, 22, 86, 31, 51, 37, 43, 75, 90, 24, 86,
96, 45, 32, 98, 36, 66, 48, 73, 73, 79, 56, 41, 21, 25, 27, 97, 18,
44, 45, 40, 80, 63, 20, 35,  0,  8, 27, 25, 35, 59, 61, 21, 37, 29,
 6, 19, 78, 50, 54, 37, 93, 33, 46, 79, 59, 29, 43,  0, 23, 17, 38,
66, 38, 89, 17, 25, 31, 65, 10, 26, 86, 58, 42, 46, 24, 95, 93,  8,
53, 32, 14, 10, 94,  8,  8, 64, 44, 74, 30, 97, 22, 11, 68, 56, 90,
96, 16, 43, 57, 91, 24, 28, 82, 90, 64, 61, 92, 28, 84, 70, 45, 85,
34,  7, 88, 89, 61, 26, 88, 41, 46,  8, 91, 41, 14, 98, 28, 26, 36,
70, 74,  7, 52, 70, 42, 66, 22, 13, 44, 91, 53, 22, 16, 40, 40, 28,
70,  6, 60, 95, 23, 16, 50, 29, 49,  9, 18, 55, 63, 60, 19, 28, 30,
31, 85, 66, 88, 63, 83, 64, 96, 13, 34, 27, 95, 36, 72, 29, 91, 22,
65, 71, 66, 11, 32,  2, 75, 39,  5, 37, 67, 81, 55, 61, 57, 81, 82,
63, 55, 54, 35, 86, 25, 24, 96, 10, 58, 59, 28, 89, 54, 52, 85, 68,
69,  8, 39, 95, 39, 82, 48, 74, 52, 74, 55,  9, 47, 84, 91, 12, 96,
82, 64,  7, 40, 73, 77, 11, 36, 68, 23, 28, 46, 75, 43,  2, 11, 47,
53, 56, 62, 62, 80, 56, 30,  3, 88, 37, 33, 73, 76, 21,  5, 76, 87,
68, 83, 62, 57, 47, 19, 88, 96, 42, 23, 44, 87, 82, 49, 63, 24, 94,
69, 54,  5, 79, 43, 12, 50,  5, 52, 92,  4, 84,  1, 33, 49, 26, 18,
44, 13, 24, 73, 89, 78, 67, 41, 11, 46, 47, 69,  0, 18, 98, 44, 85,
29, 53,  1, 45,  3,  9, 13,  2, 66, 59, 79,  6, 17, 43, 83, 26,  1,
12, 49, 71, 89, 58, 93, 39, 42, 15, 38, 55, 15, 93,  4, 90, 88, 55,
40, 55, 17, 34, 94, 57, 92, 81, 26, 60, 89, 49, 89, 30, 65, 58,  4,
19,  4, 76, 74, 71, 21, 54, 13, 16, 72, 68, 62, 61, 25, 72,  7, 12,
18, 77, 90, 62, 14,  3, 78, 65, 37, 27, 50, 95, 98, 60, 72, 58, 38,
87, 93, 19,  7, 83, 50,  3, 91, 77,  7, 64, 61, 69, 23, 76, 65, 48,
41, 92, 20, 91, 18, 70,  9,  9, 29, 85, 67,  0, 35, 98, 91, 90, 31,
53, 39, 24, 85, 96, 17,  7, 11, 96, 39, 56, 90, 79, 45, 64, 97, 41,
19, 74, 11, 10, 62, 95, 28, 96, 10,  7, 68,  7, 93, 34, 42, 68, 41,
14, 22, 58, 12, 71, 27, 98, 72, 91,  3, 43, 19, 61, 75, 20, 81, 63,
67, 56, 26, 47, 11, 31, 57, 62, 66, 19, 75, 97, 94, 13, 75, 95, 32,
50, 97, 52, 87, 32,  3, 47, 77, 48, 33, 73, 64, 49, 68, 43, 94, 77,
68, 47, 82,  2, 30, 23, 33, 34, 66, 33, 35, 88, 68, 27, 87, 54, 79,
34, 67, 65, 18,  4, 26, 30, 52, 86,  0, 29, 80, 67, 95, 39, 25, 70,
58, 35, 27, 17, 38, 91, 13, 23, 77, 79, 77, 22, 49, 98, 48, 46, 48,
 5, 63, 97, 80, 53, 20, 25, 78, 10, 65, 33, 41, 85, 90, 98, 97, 71,
95, 52,  3, 29, 69, 51, 70, 27, 22, 34,  6, 48, 72, 21, 89, 17, 97,
72, 80, 10, 57, 64, 92, 38, 15, 73, 87, 73, 48, 42, 82, 33, 56,  3,
42,  1, 53, 55, 90, 19,  6, 30, 86, 64, 49,  2,  8, 45, 76, 92,  0,
23, 69, 59, 80, 90, 32,  5, 59, 85, 89, 94, 45, 48, 86, 81, 14,  4,
77, 56, 82,  2, 85, 70, 88,  0, 75, 14, 86, 81, 97, 70, 72, 34, 40,
 5, 11, 78, 50])
```

[22]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler().fit(x)
x = sc.transform(x)
```

[23]:
```
from sklearn.model_selection import  train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2,␣
 ↪random_state =1)
```

[24]:
```
from sklearn.ensemble import RandomForestClassifier
rf_classifier = RandomForestClassifier(n_estimators = 40,criterion =␣
 ↪'entropy',max_depth = 20,random_state = 5)
rf_classifier.fit(x_train, y_train)
```

[24]:
```
RandomForestClassifier(criterion='entropy', max_depth=20, n_estimators=40,
                       random_state=5)
```

[25]:
```
pred_train = rf_classifier.predict(x_train)
pred_test = rf_classifier.predict(x_test)
```

[26]:
```
from sklearn.metrics import accuracy_score,confusion_matrix
print(confusion_matrix(y_test,pred_test))
print('Training Accuracy: ', accuracy_score(y_train, pred_train))
print('Testing Accuracy: ', accuracy_score(y_test, pred_test))
```

```
[[4 0 0 … 0 0 0]
 [0 1 0 … 0 0 0]
 [0 0 2 … 0 0 0]
 …
 [0 0 0 … 1 0 0]
 [0 0 0 … 0 2 0]
 [0 0 0 … 0 0 2]]
Training Accuracy:  1.0
Testing Accuracy:  0.9292929292929293
```

[27]:
```
x_test = df2.values
```

[28]:
```
from sklearn.preprocessing import StandardScaler
x_test = sc.transform(x_test)
y_test = rf_classifier.predict_proba(x_test)
```

[29]:
```
submission = pd.DataFrame(y_test, index=test_ids, columns=encoder.classes_)
```

[30]:
```
submission.head(10)
```

[30]:
```
    Acer_Capillipes  Acer_Circinatum  Acer_Mono  Acer_Opalus  Acer_Palmatum  \
id
4             0.000            0.000        0.0        0.000          0.000
7             0.000            0.025        0.0        0.050          0.000
9             0.000            0.650        0.0        0.000          0.025
12            0.025            0.000        0.0        0.025          0.000
13            0.050            0.025        0.0        0.000          0.025
16            0.000            0.000        0.0        0.250          0.000
19            0.000            0.000        0.0        0.325          0.000
23            0.000            0.000        0.0        0.000          0.000
24            0.000            0.025        0.0        0.000          0.000
28            0.025            0.000        0.0        0.000          0.000

    Acer_Pictum  Acer_Platanoids  Acer_Rubrum  Acer_Rufinerve  \
id
4           0.0              0.0        0.000           0.000
7           0.0              0.0        0.000           0.000
9           0.0              0.0        0.000           0.025
12          0.0              0.0        0.000           0.050
13          0.0              0.0        0.000           0.150
16          0.0              0.0        0.000           0.000
19          0.0              0.0        0.025           0.000
23          0.0              0.0        0.000           0.000
24          0.0              0.0        0.000           0.000
28          0.0              0.0        0.000           0.725

    Acer_Saccharinum  …  Salix_Fragilis  Salix_Intergra  Sorbus_Aria  \
id                    …
4              0.000  …           0.000            0.00        0.000
7              0.000  …           0.000            0.05        0.000
9              0.025  …           0.000            0.00        0.025
12             0.000  …           0.075            0.00        0.000
13             0.000  …           0.000            0.00        0.000
16             0.000  …           0.000            0.00        0.000
19             0.000  …           0.000            0.00        0.000
23             0.000  …           0.000            0.00        0.000
24             0.000  …           0.000            0.00        0.000
28             0.000  …           0.000            0.00        0.000
```

```
    Tilia_Oliveri  Tilia_Platyphyllos  Tilia_Tomentosa  Ulmus_Bergmanniana  \
id
4           0.000                 0.0            0.000               0.000
7           0.000                 0.0            0.025               0.000
9           0.000                 0.0            0.000               0.000
12          0.000                 0.0            0.100               0.125
13          0.000                 0.0            0.000               0.000
16          0.025                 0.0            0.125               0.025
19          0.025                 0.0            0.100               0.050
23          0.000                 0.0            0.000               0.000
24          0.000                 0.0            0.000               0.000
28          0.000                 0.0            0.000               0.000

    Viburnum_Tinus  Viburnum_x_Rhytidophylloides  Zelkova_Serrata
id
4             0.00                           0.0            0.000
7             0.00                           0.0            0.000
9             0.00                           0.0            0.000
12            0.00                           0.0            0.025
13            0.00                           0.0            0.000
16            0.05                           0.0            0.025
19            0.00                           0.0            0.050
23            0.00                           0.0            0.000
24            0.00                           0.0            0.000
28            0.00                           0.0            0.000

[10 rows x 99 columns]
```

[31]:
```
submission.to_csv('submission_leaf_classification.csv')
```

[ ]:
```

```