

# Exploratory Data Analysis

May 21, 2021

## 1 Exploratory Data Analysis

```
[1]: import pandas as pd
```

```
[2]: df = pd.read_csv('iris_csv.csv')
df.head()
```

```
[2]:
```

	sepalength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

### 1.1 Data Preprocessing

```
[3]: from sklearn.preprocessing import LabelEncoder
```

```
[4]: le = LabelEncoder()
df['class'] = le.fit_transform(df['class'])
```

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

```
[5]:
```

	sepalength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   sepalength      150 non-null   float64
```

```

1  sepalwidth  150 non-null  float64
2  petallength 150 non-null  float64
3  petalwidth  150 non-null  float64
4  class       150 non-null  int32
dtypes: float64(4), int32(1)
memory usage: 5.4 KB

```

```
[7]: df.describe()
```

```

[7]:      sepallength  sepalwidth  petallength  petalwidth  class
count    150.000000   150.000000   150.000000   150.000000  150.000000
mean         5.843333     3.054000     3.758667     1.198667    1.000000
std          0.828066     0.433594     1.764420     0.763161    0.819232
min          4.300000     2.000000     1.000000     0.100000    0.000000
25%          5.100000     2.800000     1.600000     0.300000    0.000000
50%          5.800000     3.000000     4.350000     1.300000    1.000000
75%          6.400000     3.300000     5.100000     1.800000    2.000000
max          7.900000     4.400000     6.900000     2.500000    2.000000

```

```
[8]: df['class'].unique()
```

```
[8]: array([0, 1, 2])
```

```
[9]: df['class'].value_counts()
```

```

[9]: 2    50
     1    50
     0    50
     Name: class, dtype: int64

```

```
[10]: df.corr()
```

```

[10]:      sepallength  sepalwidth  petallength  petalwidth  class
sepallength    1.000000   -0.109369     0.871754     0.817954   0.782561
sepalwidth     -0.109369    1.000000    -0.420516    -0.356544  -0.419446
petallength     0.871754   -0.420516    1.000000     0.962757   0.949043
petalwidth      0.817954   -0.356544     0.962757    1.000000   0.956464
class           0.782561   -0.419446     0.949043     0.956464   1.000000

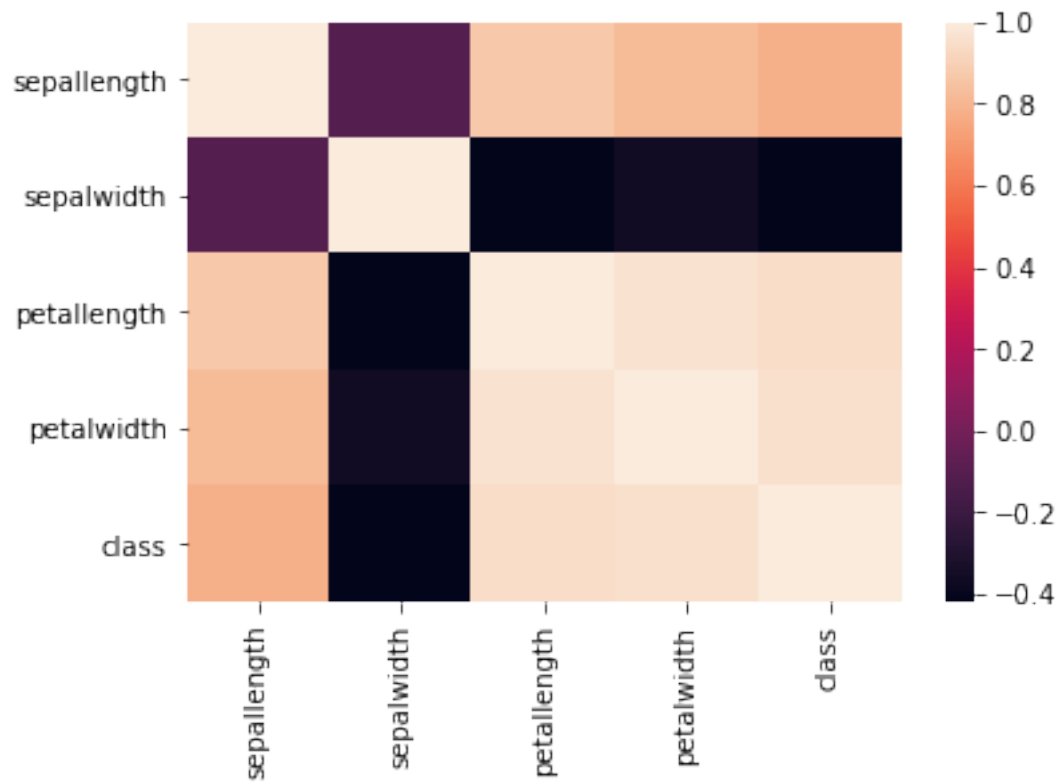
```

```
[11]: import seaborn as sns
```

```
[12]: co = df.corr()
```

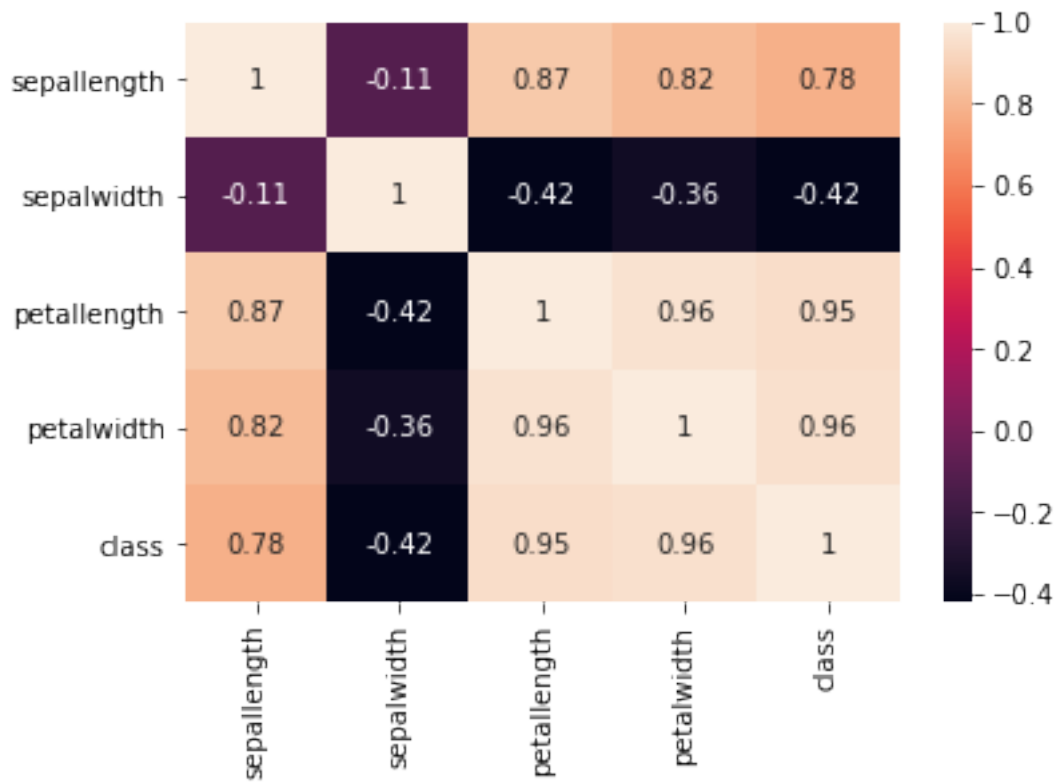
```
[13]: sns.heatmap(co)
```

```
[13]: <matplotlib.axes._subplots.AxesSubplot at 0x25a4cf42580>
```



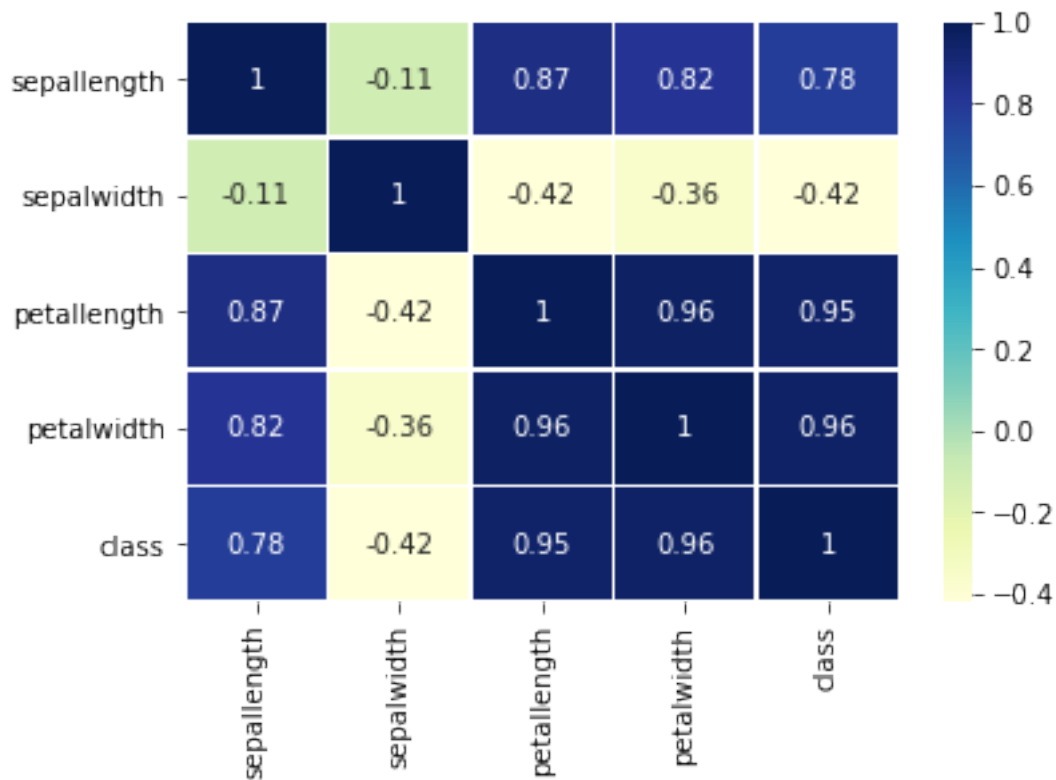
```
[14]: sns.heatmap(co, annot=True)
```

```
[14]: <matplotlib.axes._subplots.AxesSubplot at 0x25a4d6e0c70>
```



```
[15]: sns.heatmap(co, annot=True, cmap="YlGnBu", linewidth=.5)
```

```
[15]: <matplotlib.axes._subplots.AxesSubplot at 0x25a4d7d7ca0>
```



```
[16]: sns.heatmap(co, annot=True, linewidth=.5, cbar_kws={"orientation": "horizontal",
↪ "horizontal"})
```

```
[16]: <matplotlib.axes._subplots.AxesSubplot at 0x25a4d7d7310>
```

