In [7]:
```
import sklearn
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import scipy
import statistics
from sklearn import model_selection
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import label_binarize
import os
```

In [16]:
```
path = os.getcwd()
iris_df = pd.read_csv(path+'\\Learn Dataset\\iris_dataset_missing.csv')
iris_df_nona = iris_df.dropna()
iris_df_nona["Class"] = list(iris_df_nona.loc[:,"species"].values)
iris_df_nona["Class"]=iris_df_nona["Class"].replace("Iris-versicolor",0).replace("Iris-setosa",1).replace("Iris-virginica",2)
heart_df = pd.read_csv(path+'\\Learn Dataset\\heart_disease_missing.csv')
heart_df_nona = heart_df.dropna()
heart_df_nona["cp"] = heart_df_nona.loc[:,"cp"].replace("Asympt.",0).replace("Atypical",1).replace("Non",2).replace("Typical",3)
heart_df_nona["restecg"] = heart_df_nona.loc[:,"restecg"].replace("Normal",0).replace("ST-T wave",1).replace("LV hyper", 2)
heart_df_nona["slope"] = heart_df_nona.loc[:,"slope"].replace("down",0).replace("flat",1).replace("up",2)
heart_df_nona["thal"] = heart_df_nona.loc[:,"thal"].replace("Revers.",0).replace("Normal",1).replace("Fixed",2)

features = ["exang","thal","slope","cp","oldpeak"]
heart_df_sub = heart_df_nona.copy()
for i in heart_df_nona.columns:
    if i not in features and i not in ["target"]:
        heart_df_sub.drop(columns = [i], inplace=True)
```

# CM2

## Correlation Coefficients

### Iris dataset

In [17]:
```
iris_df_nona.corr()
```

Out[17]:

| | sepal_length | sepal_width | petal_length | petal_width | Class |
|---|---|---|---|---|---|
| **sepal_length** | 1.000000 | -0.014750 | 0.879809 | 0.813983 | 0.351707 |
| **sepal_width** | -0.014750 | 1.000000 | -0.285793 | -0.252136 | 0.261473 |
| **petal_length** | 0.879809 | -0.285793 | 1.000000 | 0.958429 | 0.319066 |
| **petal_width** | 0.813983 | -0.252136 | 0.958429 | 1.000000 | 0.382987 |
| **Class** | 0.351707 | 0.261473 | 0.319066 | 0.382987 | 1.000000 |

We can see high amount of correlation between 'petal_length' & 'sepal_length' and 'petal_length' and 'petal_width'. From the pairplots, we saw how petal_length and petal_width individually had good class separation in its distribution. High correlation between features can open doors for us to use one feature or the other when we have too many features and want to reduce features.

### Heart Disease Dataset

In [18]:
```
heart_df_nona.corr()
```

Out[18]:

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **age** | 1.000000 | -0.162805 | -0.094568 | 0.311194 | 0.184968 | 0.058412 | -0.104239 | -0.418467 | 0.117830 | 0.132613 | -0.151080 | 0.260586 | 0.031341 | -0.177700 |
| **sex** | -0.162805 | 1.000000 | -0.058450 | -0.080464 | -0.198022 | 0.047452 | -0.072782 | -0.051201 | 0.090487 | 0.103859 | -0.098354 | 0.146411 | 0.235369 | -0.249585 |
| **cp** | -0.094568 | -0.058450 | 1.000000 | -0.008539 | -0.078022 | 0.053333 | -0.030776 | 0.288617 | -0.396391 | -0.177998 | 0.184471 | -0.207397 | -0.157975 | 0.531465 |
| **trestbps** | 0.311194 | -0.080464 | -0.008539 | 1.000000 | 0.158249 | 0.166439 | -0.112712 | -0.109834 | 0.064888 | 0.160363 | -0.175388 | 0.072259 | 0.022846 | -0.117806 |
| **chol** | 0.184968 | -0.198022 | -0.078022 | 0.158249 | 1.000000 | -0.033144 | -0.094489 | -0.073405 | 0.086526 | 0.073519 | 0.010260 | 0.038942 | 0.004097 | -0.114501 |
| **fbs** | 0.058412 | 0.047452 | 0.053333 | 0.166439 | -0.033144 | 1.000000 | -0.139330 | 0.021496 | 0.075085 | -0.081840 | -0.039972 | 0.109652 | 0.021199 | 0.010362 |
| **restecg** | -0.104239 | -0.072782 | -0.030776 | -0.112712 | -0.094489 | -0.139330 | 1.000000 | -0.017183 | -0.001014 | -0.039762 | 0.052756 | -0.081353 | -0.006021 | 0.067840 |
| **thalach** | -0.418467 | -0.051201 | 0.288617 | -0.109834 | -0.073405 | 0.021496 | -0.017183 | 1.000000 | -0.387369 | -0.363022 | 0.462101 | -0.192047 | -0.116345 | 0.438963 |
| **exang** | 0.117830 | 0.090487 | -0.396391 | 0.064888 | 0.086526 | 0.075085 | -0.001014 | -0.387369 | 1.000000 | 0.262387 | -0.327414 | 0.079687 | 0.167880 | -0.449802 |
| **oldpeak** | 0.132613 | 0.103859 | -0.177998 | 0.160363 | 0.073519 | -0.081840 | -0.039762 | -0.363022 | 0.262387 | 1.000000 | -0.674435 | 0.156793 | 0.185610 | -0.456554 |
| **slope** | -0.151080 | -0.098354 | 0.184471 | -0.175388 | 0.010260 | -0.039972 | 0.052756 | 0.462101 | -0.327414 | -0.674435 | 1.000000 | -0.080760 | -0.160397 | 0.427994 |
| **ca** | 0.260586 | 0.146411 | -0.207397 | 0.072259 | 0.038942 | 0.109652 | -0.081353 | -0.192047 | 0.079687 | 0.156793 | -0.080760 | 1.000000 | 0.112082 | -0.307917 |
| **thal** | 0.031341 | 0.235369 | -0.157975 | 0.022846 | 0.004097 | 0.021199 | -0.006021 | -0.116345 | 0.167880 | 0.185610 | -0.160397 | 0.112082 | 1.000000 | -0.352234 |
| **target** | -0.177700 | -0.249585 | 0.531465 | -0.117806 | -0.114501 | 0.010362 | 0.067840 | 0.438963 | -0.449802 | -0.456554 | 0.427994 | -0.307917 | -0.352234 | 1.000000 |

From the entire pairwise plot, we can notice a decent amount of negative correlation between thalach and age. Another interesting correlation is the negative correlation between oldpeak and slope. Looking at the correlation of features with target, we can see good positive correlation of target with features {cp, thalach, slope} and negative correlation with features {exang, oldpeak, thal}. Correlation comparison with targets can help us choose features. The features that have good correlation(either positive or negative) with target are quite significant for our model.

In [19]:

```
heart_df_sub.corr()
```

Out[19]:

| | cp | exang | oldpeak | slope | thal | target |
|---|---|---|---|---|---|---|
| **cp** | 1.000000 | -0.396391 | -0.177998 | 0.184471 | -0.157975 | 0.531465 |
| **exang** | -0.396391 | 1.000000 | 0.262387 | -0.327414 | 0.167880 | -0.449802 |
| **oldpeak** | -0.177998 | 0.262387 | 1.000000 | -0.674435 | 0.185610 | -0.456554 |
| **slope** | 0.184471 | -0.327414 | -0.674435 | 1.000000 | -0.160397 | 0.427994 |
| **thal** | -0.157975 | 0.167880 | 0.185610 | -0.160397 | 1.000000 | -0.352234 |
| **target** | 0.531465 | -0.449802 | -0.456554 | 0.427994 | -0.352234 | 1.000000 |

The above table shows the correlation matrix for the features we selected. The same observations as above applies here as well since this is just a reduced form of the above matrix.

# Mean, Variance, Skew and Kurtosis

## Iris Dataset

In [20]:

```
cols = iris_df_nona.columns
for i in cols:
    if i not in ["Class"]:
        try:
            print("Skew of ",i, scipy.stats.skew(iris_df_nona.loc[:,i]))
            print("Kurtosis of ",i, scipy.stats.kurtosis(iris_df_nona.loc[:,i]))
            print("Mean of ",i, statistics.mean(iris_df_nona.loc[:,i]))
            print("Variance of ",i, statistics.variance(iris_df_nona.loc[:,i]))
            print("")
```

```
        except:
            pass
```

```
Skew of  sepal_length 0.41780073656176503
Kurtosis of  sepal_length -0.7056907198851405
Mean of  sepal_length 5.8678935378329244
Variance of  sepal_length 0.7961469847275912

Skew of  sepal_width 0.1837539677513432
Kurtosis of  sepal_width 0.24582896985113
Mean of  sepal_width 3.0549349521835065
Variance of  sepal_width 0.1931281503408631

Skew of  petal_length -0.23438961958410212
Kurtosis of  petal_length -1.4011477488138373
Mean of  petal_length 3.8081183998737864
Variance of  petal_length 3.2811668273075663

Skew of  petal_width -0.09949385864536328
Kurtosis of  petal_width -1.313533572417677
Mean of  petal_width 1.2098255632819341
Variance of  petal_width 0.6298904416646675
```

A low skew of all features in this dataset says how these features are close to being symmetric and are not inclined to any side. Features related to 'sepal' have lesser kurtosis indicating a distribution closer to normal distribution. Petal features on the other hand have higher negative kurtosis indicating a longer tail with a sharper peak in the distribution. The mean here signifies the expected or the average value about which the values are distributed. Variance of petal length is high

## Heart Disease Dataset

In [21]:

```
cols = heart_df_sub.columns
for i in cols:
    if i not in ["target"]:
        try:
            print("Skew of ",i, scipy.stats.skew(heart_df_sub.loc[:,i]))
            print("Kurtosis of ",i, scipy.stats.kurtosis(heart_df_sub.loc[:,i]))
            print("Mean of ",i, statistics.mean(heart_df_sub.loc[:,i]))
            print("Variance of ",i, statistics.variance(heart_df_sub.loc[:,i]))
            print("")
        except:
            pass
```

```
Skew of  cp 0.4685078835742903
Kurtosis of  cp -1.2222401604461226
Mean of  cp 0.9482758620689655
Variance of  cp 1.0319912298186167

Skew of  exang 0.5482409888162154
Kurtosis of  exang -1.6994318181818184
Mean of  exang 0.367816091954023
Variance of  exang 0.2338715035545811

Skew of  oldpeak 1.2845026147862473
Kurtosis of  oldpeak 1.4068902594982005
Mean of  oldpeak 1.0964225740435576
Variance of  oldpeak 1.6380757926488272

Skew of  slope -0.5605642862682565
Kurtosis of  slope -0.6259790268452767
Mean of  slope 1.396551724137931
Variance of  slope 0.4025313932629061

Skew of  thal -0.30412815102818863
Kurtosis of  thal -0.6446430199180906
Mean of  thal 2.3518800831489513
Variance of  thal 0.3769508498064522
```

This shows the skew, kurtosis, mean and variance of the features we have selected. cp and slope are two features that have high variance. They are indeed spread out more in comparison with other attributes we have selected. The feature 'oldpeak' has the most amount of positive skew indicating peak off center. One another interesting observation is that, apart from 'oldpeak' none of the features are of type numeric. These measures might not have the same significance for categorical attributes as they have for numeric types.

In [ ]: