In [3]:

```
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
```

Libraries and References

Python Libraries : sklearn, pandas, numpy, seaborn, matplotlib, scipy, statistics and os

References

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html ; https://scikit-learn.org/ ; https://developers.google.com/machine-learning/data-prep/transform/normalization ; https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4 ; https://www.analyticsvidhya.com/blog/2020/07/what-is-skewness-statistics/ ; https://www.simplypsychology.org/kurtosis.html

Loading datasets

Here are we are loading the respective datasets preprocessing them by removing nan values. We are also encoding string data into numerical data that is suitable for analysis and prediction using KNN model. The encoded species of iris data is stored in a new column called 'Class' for further use down the line.

In [4]:

```
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)
```

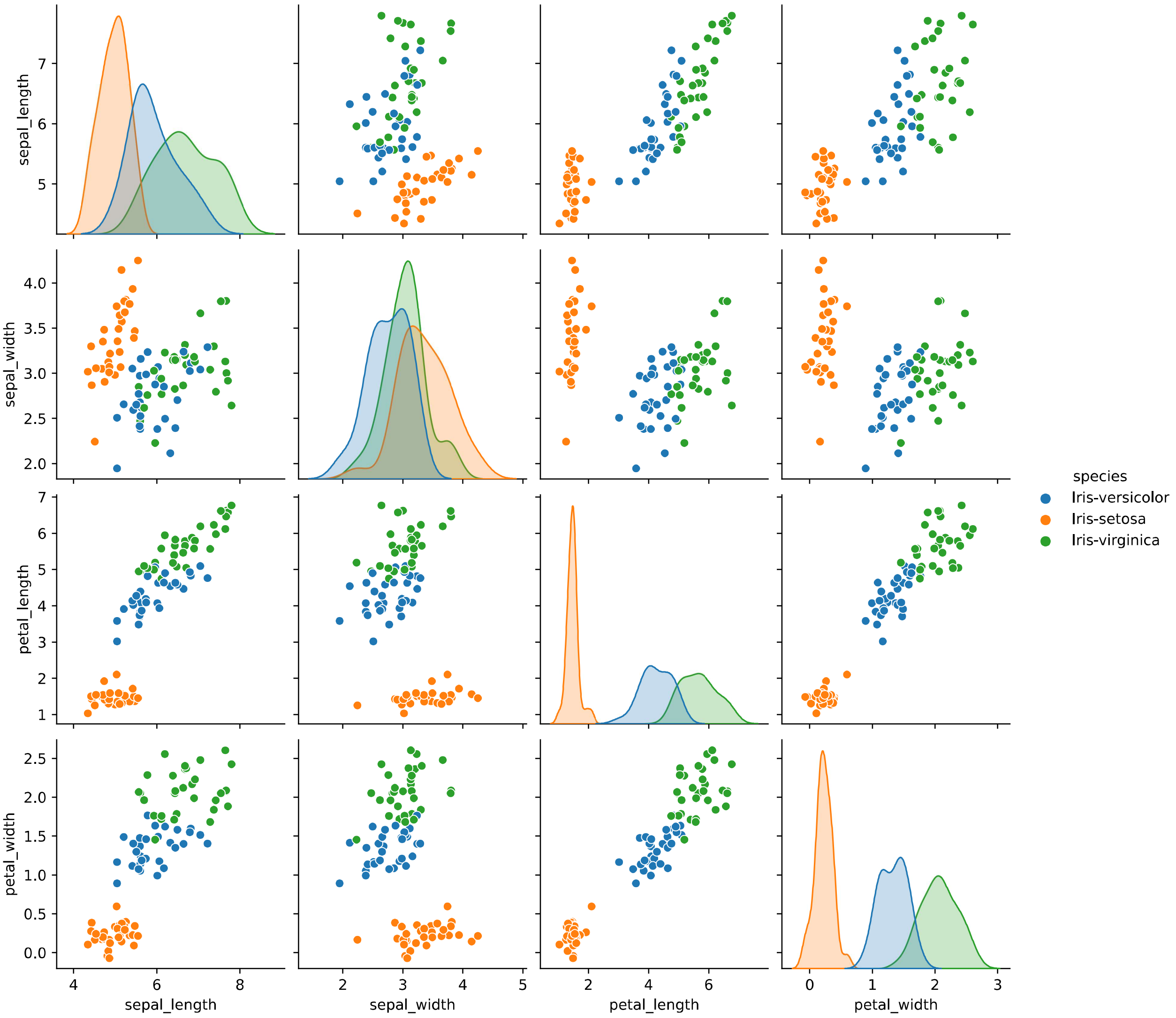
CM1

Pair plot of iris dataset downloaded from learn website is shown below

In [6]:

```
fig1 = plt.figure(figsize=(6,6))
sn = sns.pairplot(iris_df_nona.drop(columns=["Class"]),hue='species', dropna=True)
```

```
<Figure size 432x432 with 0 Axes>
```



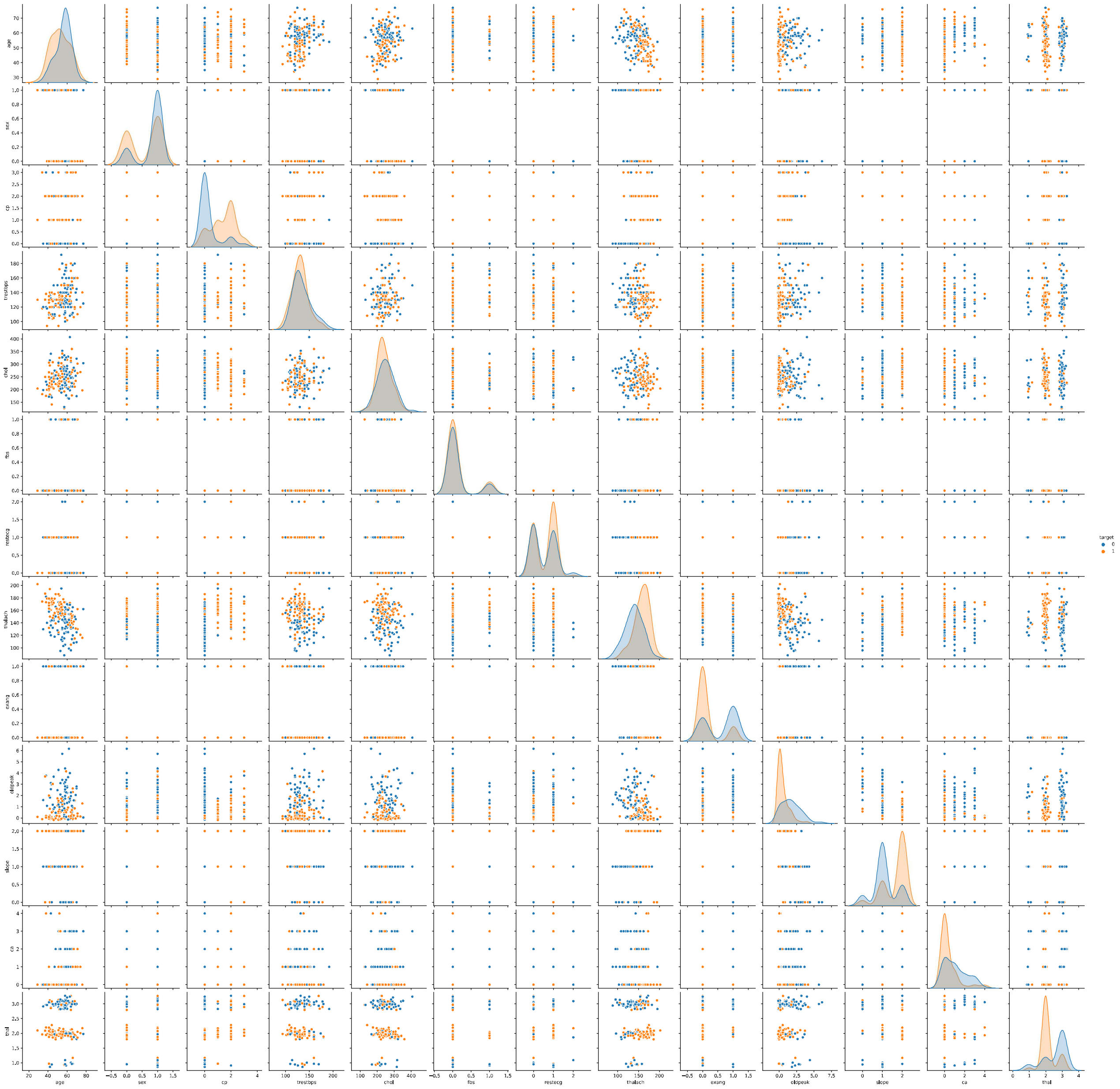
The above plot shows the feature wise scatter plot and species distribution per attribute. We can see how features like petal_length and petal_width have minimum overlap in class distribution as compared to other features. This gives an idea of how well we are able to classify using that attribute. Since we have only 4 features and all of them seems to have decent amount of class separation, we will use all 4 features for our classification problem.

Pair plot for Heart Disease Dataset from Learn

In [7]:

```
fig1 = plt.figure(figsize=(6,6))
sn = sns.pairplot(heart_df_nona,hue='target', dropna=True)
```

```
<Figure size 432x432 with 0 Axes>
```



The above figure shows the pairplot for all the features in heart disease dataset. This plot will help us make informed decision on what features to choose and what not to choose to some extent. From this plot, we can see that there are several features, for instance 'fbs', where there is a good amount of overlap between classes making this attribute by itself not very interesting for us to be used for classification. Other the other hand, there are features like 'cp', 'thal', 'slope', 'oldpeak' and 'exang' where the class wise overlap is minimal in comparison. These are properties that are interesting to us. In all of these features, we can notice how their pairplot with other features show a good amount of separation between classes in the scatter plot.

We will shortlist the following features based on our observation from the pairplot and take them up for further analysis.

Selected Features = {"exang","thal","slope","cp","oldpeak"}

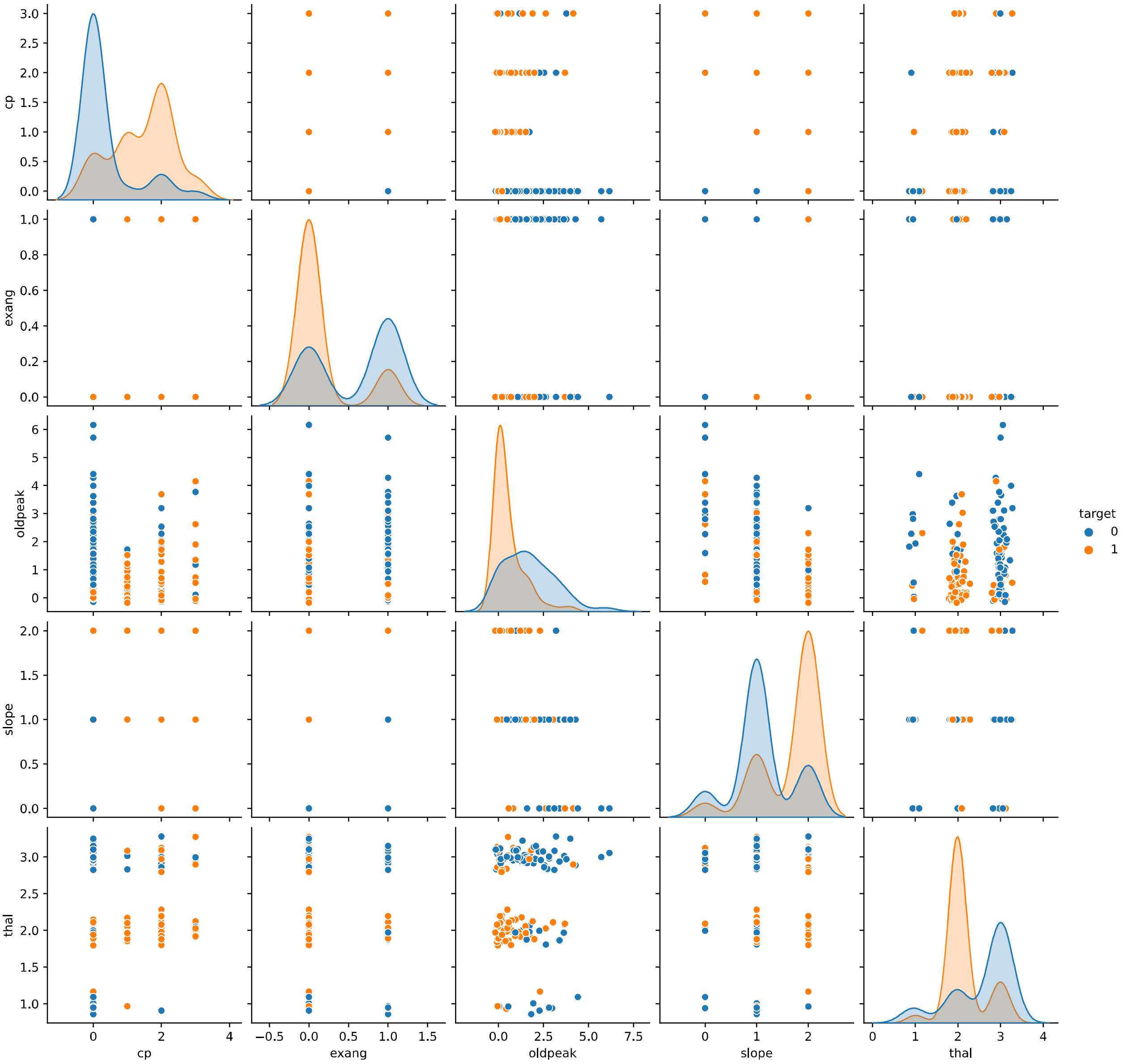
In [5]:

```
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)
```

In [6]:

```
fig1 = plt.figure(figsize=(6,6))
sn = sns.pairplot(heart_df_sub,hue='target', dropna=True)
```

```
<Figure size 432x432 with 0 Axes>
```



We have chosen these features because, from the correlation plots, we can see a good amount of clustering in all scatter pairplots. This will allow us to perform classification effectively. Some interesting patterns are than there is a higher chance of having heart disease if thal is 'Fixed' (2). Other interesting patterns include higher chance of heart disease if slope is 'up' and lower chance if slope is 'flat'. Similarly, lower chances if cp is 'Asymptomatic'(0).

```
In [ ]:
```