CM1

January 29, 2021

1 Finding correlations between features using pair plots

1.1 Required libraries

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

1.2 Heart Diseases Dataset

Loading the dataset The table below displays 5 rows randomly selected from dataset of heart disease.

[2]:
```
df_heart= pd.read_csv("heart_disease_missing.csv")
df_heart.sample(5)
```

[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	\
20	56	1	3	119.968114	192.953625	0	0.0	161.944248	0	
45	52	1	2	172.122494	198.921917	1	1.0	162.100664	0	
84	34	1	3	118.187876	181.999626	0	0.0	174.145214	0	
161	48	1	0	130.080421	255.871039	1	0.0	150.190570	1	
150	71	0	1	159.963230	302.009742	0	1.0	162.062949	0	

	oldpeak	slope	ca	thal	target
20	NaN	1.0	0	3.096325	1
45	0.449844	2.0	0	2.837360	1
84	-0.110147	2.0	0	1.964529	1
161	0.122920	2.0	2	2.970361	0
150	0.393740	2.0	2	1.852958	1

Selection of feature

[3]:
```
df_heart = pd.read_csv("heart_disease_missing.csv")
df_heart = df_heart.drop(['target'], axis = 'columns')
df_heart.corr(method="pearson")
```

[3]:

	age	sex	cp	trestbps	chol	fbs	\
age	1.000000	-0.140074	-0.084230	0.335944	0.185861	0.050823	
sex	-0.140074	1.000000	-0.057939	-0.049906	-0.195213	0.081750	
cp	-0.084230	-0.057939	1.000000	-0.007449	-0.061591	0.057205	

```
trestbps  0.335944 -0.049906 -0.007449  1.000000  0.162162  0.138907
chol      0.185861 -0.195213 -0.061591  0.162162  1.000000 -0.025549
fbs       0.050823  0.081750  0.057205  0.138907 -0.025549  1.000000
restecg  -0.124819 -0.050203  0.035935 -0.106940 -0.079196 -0.110983
thalach  -0.382280  0.017446  0.246019 -0.100284 -0.057679  0.036934
exang     0.114545  0.122773 -0.349369  0.088717  0.065738  0.098474
oldpeak   0.131735  0.098912 -0.187518  0.164742  0.048991 -0.094862
slope    -0.117989 -0.057160  0.156145 -0.134180  0.028301 -0.019514
ca        0.266278  0.120466 -0.180350  0.080824  0.037820  0.128097
thal      0.049743  0.208391 -0.141699  0.022495 -0.032143  0.009830

             restecg   thalach     exang   oldpeak     slope        ca      thal
age      -0.124819 -0.382280  0.114545  0.131735 -0.117989  0.266278  0.049743
sex      -0.050203  0.017446  0.122773  0.098912 -0.057160  0.120466  0.208391
cp        0.035935  0.246019 -0.349369 -0.187518  0.156145 -0.180350 -0.141699
trestbps -0.106940 -0.100284  0.088717  0.164742 -0.134180  0.080824  0.022495
chol     -0.079196 -0.057679  0.065738  0.048991  0.028301  0.037820 -0.032143
fbs      -0.110983  0.036934  0.098474 -0.094862 -0.019514  0.128097  0.009830
restecg   1.000000  0.016873 -0.036140 -0.065872  0.056843 -0.079777 -0.028263
thalach   0.016873  1.000000 -0.360246 -0.351900  0.463824 -0.177231 -0.147013
exang    -0.036140 -0.360246  1.000000  0.279862 -0.314675  0.101805  0.187304
oldpeak  -0.065872 -0.351900  0.279862  1.000000 -0.652509  0.194648  0.216788
slope     0.056843  0.463824 -0.314675 -0.652509  1.000000 -0.084256 -0.180782
ca       -0.079777 -0.177231  0.101805  0.194648 -0.084256  1.000000  0.134459
thal     -0.028263 -0.147013  0.187304  0.216788 -0.180782  0.134459  1.000000
```

Pearson correlation method is being used to find the correlation between features and we have selected the features which shows similarity or dissimilarity.For data points to be strongly similar the correlation coefficient should be above 0.7 value.And for the data set to be strongly disimilar the correlation coefficient should be below -0.7 value.

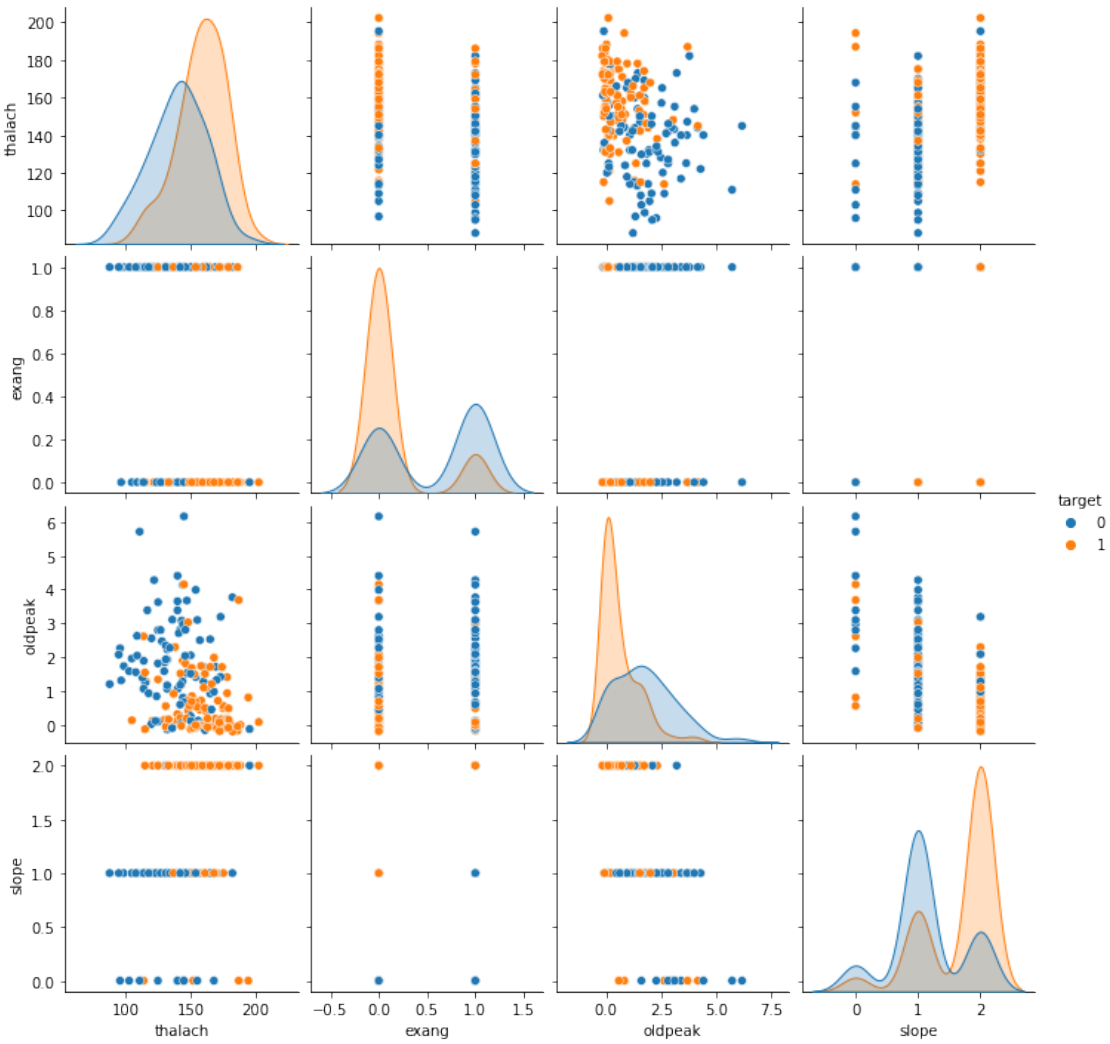
So based on the data available, we have selected features which had highest correlational coefficient. You can see in the table, 'slope' shows highest negative correlation coefficient with 'oldpeak' and positive correlation coefficient with 'thalach'. 'Exang' shows highest negative correlation coefficient with 'thalach' and positive correlation coefficient with 'oldpeak'. Thus the features selected for pair plots are 'exang','oldpeak','trestbps','slope','thalach'.

1.2.1 Scatter Plot

Below we have plotted the pairs plot of the features we selected using correlation method.

[4]:
```
df_heart = pd.read_csv("heart_disease_missing.csv",␣
 ↪usecols=['slope','oldpeak','thalach','exang','target'])
sns.pairplot(df_heart,hue="target")
```

[4]:
```
<seaborn.axisgrid.PairGrid at 0x195af753460>
```



In the subplot of exang vs slope, it is evident that if exercise induced angina is absent and slope is flat then the person has the high possibility of having heart disease.Whereas, exercise induced angina is present and slope is downslopping then the person is not likely to posses the heart disease.

Similarly, in subplot thalach vs slope, you can observe if the heart rate achieved is lower in thalium stress test then the person is less likely to have heart dieases. Also, if the slope of peak is flat then the person is most likely to be diagonised by heart diseases, it is the same observation we made in the previous subplot of exang vs slope.

1.3 Iris Dataset

Loading the dataset The table below displays 5 rows randomly selected from dataset of heart disease.

[5]:
```
df_iris= pd.read_csv("iris_dataset_missing.csv")
df_iris.sample(5)
```

[5]:

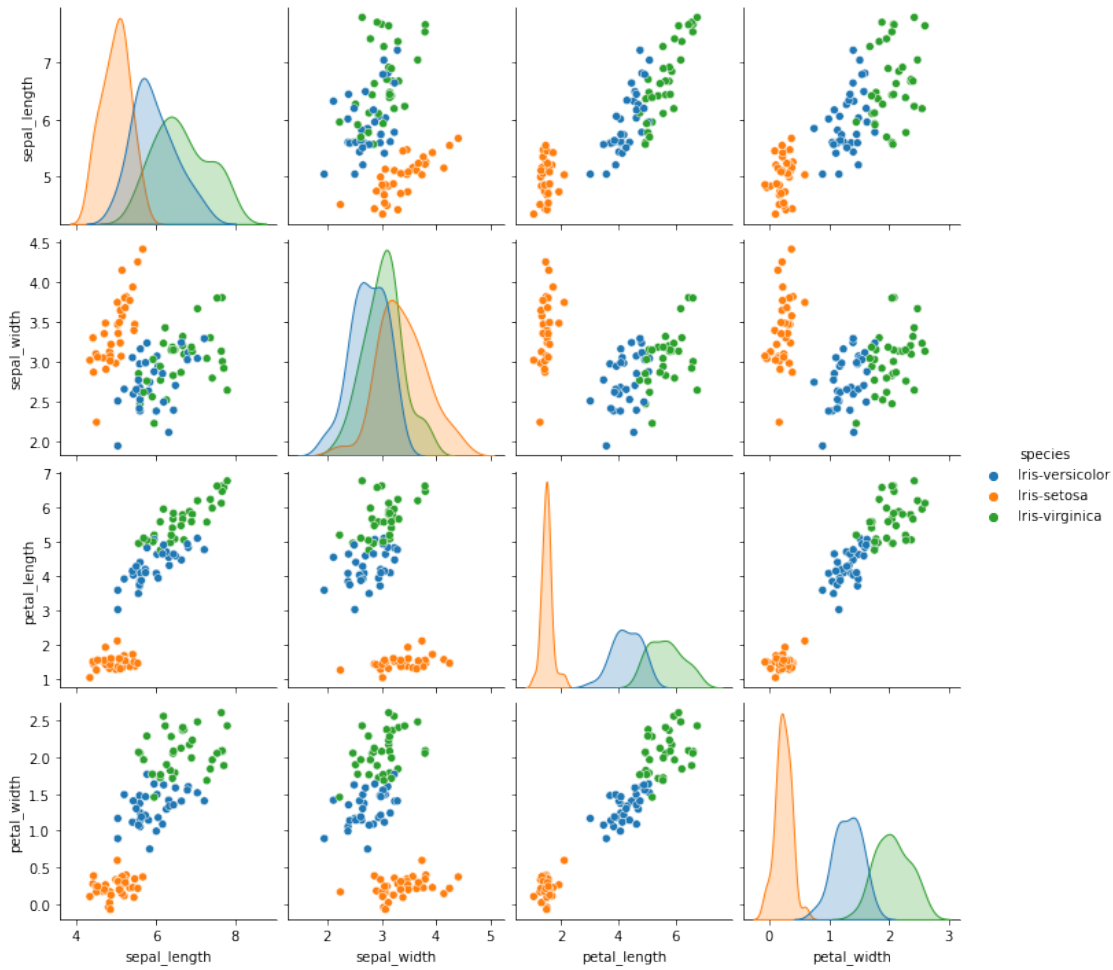
	sepal_length	sepal_width	petal_length	petal_width	species
62	7.048442	3.664169	6.192330	2.478509	Iris-virginica
40	6.435413	2.827586	5.661419	2.050435	Iris-virginica
73	6.387364	3.146142	5.184828	2.277872	Iris-virginica
85	5.911822	2.560512	NaN	1.766513	Iris-virginica
99	4.705739	3.350510	1.541564	0.200065	Iris-setosa

1.3.1 Scatter Plot

Below we have plotted the pairs plot for all the featurs of iris dataset.

[6]:
```
df_iris = pd.read_csv("iris_dataset_missing.csv")
sns.pairplot(df_iris, hue="species")
```

[6]: <seaborn.axisgrid.PairGrid at 0x195b4ec5970>



From the subplots, we can easily observe spieces Iris-setosa as it has lower petal width and petal length compared to other two spieces. The three species can be divided into classes in the subplot

petal width vs petal length, with Iris-setos having smallest petal width and length, and Iris-virginica having biggest petal width and length.

1.4 References

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sample.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html
https://seaborn.pydata.org/generated/seaborn.pairplot.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html