ECE 657A : Data and Knowledge Modeling and Analysis

Assignment 1 : Basic Environment Set-up and Classification

Heart-Disease dataset

Libraries Used:

- numpy
- pandas
- seaborn
- matplotlib
- scipy
- scikit-learn

Question 1: Data Exploration

[CM1]

Importing libraries

[1]:
```
#importing libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Load Heart Disease dataset

[2]:
```
# load dataset
df_heart = pd.read_csv('train.csv')
df_heart_test = pd.read_csv('test.csv')
```

Displaying and exploring the Heart disease DataFrame created:

Feature descriptions

Below is the group of features presents in the dataset segregated by their type (numerical, categorical, ordinal, binary)

Binary

sex (0 = female; 1 = male)

fbs: Fasting blood sugar > 120 mg/dl

exang: Exercise induced angina (0 = no; 1 = yes)

Categorical

cp: Chest pain type (0 = Asymptomatic angina; 1 = Atypical angina; 2 = Non-angina; 3 = Typical angina)

restecg: Resting ECG (0 = Left ventricular hypertrophy; 1 = Normal; 2 = ST-T wave abnormality)

slope: Slope of the peak exercise ST segment (0 = downsloping; 1 = upsloping; 2 = flat)

thal: Thalium stress test result (0 = NA; 1 = Fixed defect; 2 = Normal; 3 = Reversible defect)

Ordinal

ca: number of major vessels (0-3) colored by flourosopy

Numeric

age

oldpeak: ST depression induced by exercise relative to rest

trestbps: Resting blood pressure

chol: Serum cholestoral in mg/dl

thalach: Maximum heart rate achieved during thalium stress test

Target

target: 1 = heart disease; 0 = no heart disease

[3]:
```
# datatypes 'binary' , 'categorical' , 'ordinal , 'numeric' , 'target'
bins = ['sex', 'fbs', 'exang']
cats = ['cp', 'restecg', 'slope', 'thal']
ords = ['ca']
nums = ['age', 'oldpeak', 'trestbps', 'chol', 'thalach']
target = ['target']
```

[4]:
```
df_heart.describe()
```

[4]:

	age	sex	cp	trestbps	chol	fbs
count	212.000000	212.000000	212.000000	205.000000	202.000000	212.000000
mean	54.311321	0.688679	0.957547	131.784610	244.133256	0.132075
std	9.145339	0.464130	1.022537	18.057222	46.444257	0.339374
min	29.000000	0.000000	0.000000	93.944184	126.085811	0.000000
25%	47.000000	0.000000	0.000000	119.968114	211.969594	0.000000
50%	55.000000	1.000000	1.000000	130.010256	241.467023	0.000000
75%	61.000000	1.000000	2.000000	139.965470	272.484222	0.000000
max	77.000000	1.000000	3.000000	192.020200	406.932689	1.000000

	restecg	thalach	exang	oldpeak	slope	ca
count	207.000000	208.000000	212.000000	200.000000	210.000000	212.000000
mean	0.560386	149.647978	0.344340	1.113106	1.423810	0.731132
std	0.535149	22.076206	0.476277	1.255908	0.623622	1.038762
min	0.000000	88.032613	0.000000	-0.185668	0.000000	0.000000
25%	0.000000	135.946808	0.000000	0.050778	1.000000	0.000000
50%	1.000000	151.939216	0.000000	0.726060	1.000000	0.000000
75%	1.000000	165.260092	1.000000	1.816733	2.000000	1.000000
max	2.000000	202.138041	1.000000	6.157114	2.000000	4.000000

	thal	target
count	211.000000	212.000000
mean	2.349112	0.542453
std	0.602117	0.499374
min	0.858554	0.000000
25%	1.949795	0.000000
50%	2.078759	1.000000
75%	2.970842	1.000000
max	3.277466	1.000000

Column 'thal' which is categorical type, has decimal values.

[5]:

```
df_heart.head()
```

[5]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang
0	76	0	2	140.102822	197.105970	0	2.0	115.952071	0
1	43	0	0	132.079599	341.049462	1	0.0	135.970028	1
2	47	1	2	107.899290	242.822816	0	1.0	152.210039	0
3	51	1	2	99.934001	NaN	0	1.0	143.049207	1
4	57	1	0	110.103508	334.952353	0	1.0	143.099327	1

	oldpeak	slope	ca	thal	target
0	1.284822	1.0	0	2.175904	1
1	3.110483	1.0	0	3.082071	0
2	-0.023723	2.0	0	2.020827	0
3	1.195082	1.0	0	2.100312	1
4	3.082052	1.0	1	2.831509	0

[6]:

```
# choosing features
choosen_features = ['cp', 'oldpeak', 'exang', 'slope', 'thalach']

# numeric
```

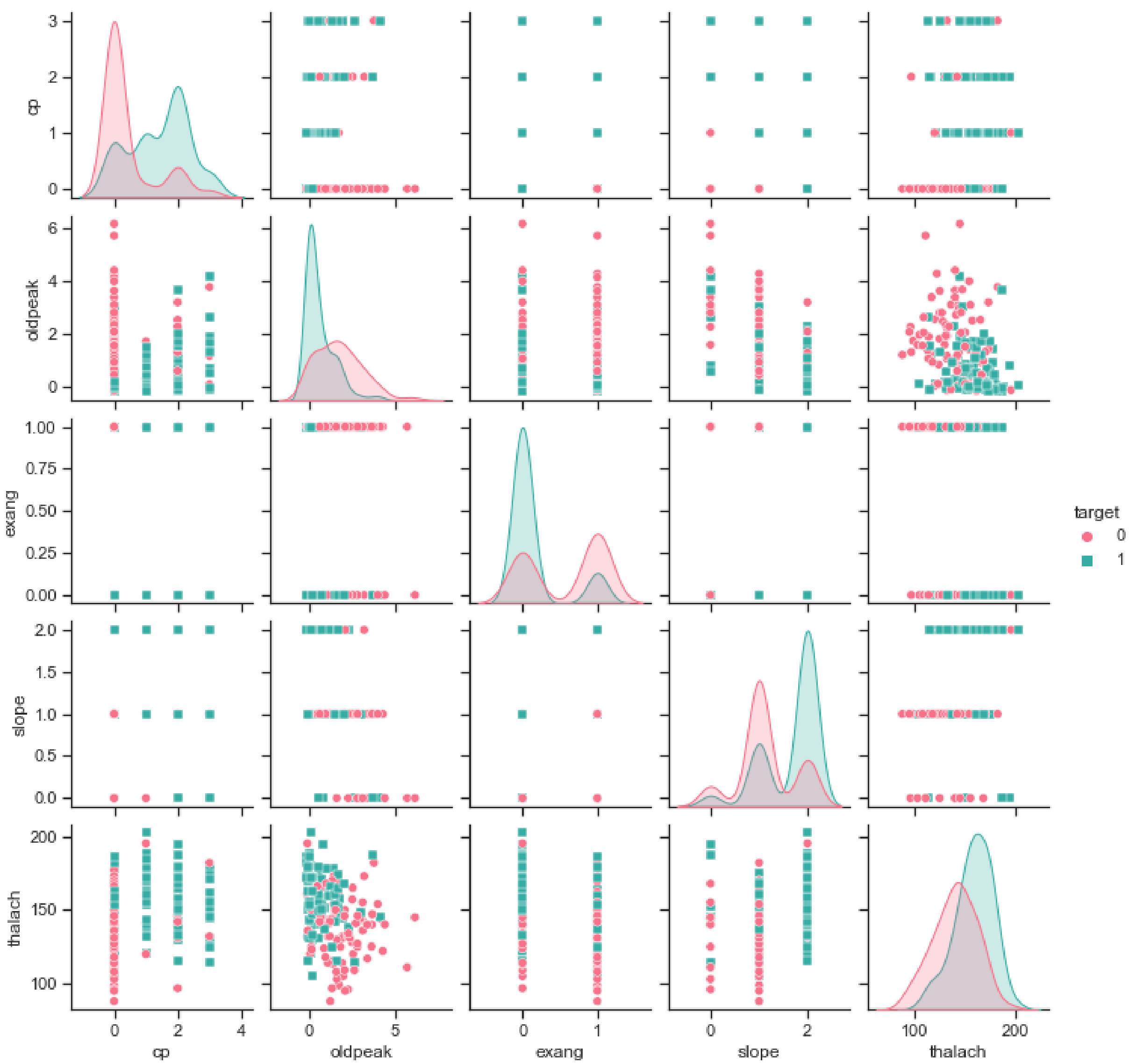
```python
choosen_features_nums = ['oldpeak','thalach']

# categorical
choosen_features_cats = ['cp','exang','slope']
```

Visualizing the data distribution by generating "pair plots" (using pairplot method of the seaborn library)

[7]:
```python
sns.set(style='ticks', color_codes=True)
sns.pairplot(df_heart, x_vars= choosen_features,y_vars= choosen_features,␣
↪hue='target', palette='husl', markers=['o', 's'],height=2)
```

[7]:
```
<seaborn.axisgrid.PairGrid at 0x17ce58ee1c0>
```



From the "pair plot" visualization, we observe that :

- oldpeak having a significant separation relation i.e. low overlapping between disease and non-disease.

- thalach having a mild separation relation between disease and non-disease.
- similarly we see that cp , slope and exang have observable seperation between disease and non-disease.
- Other features don't form any clear separation and are mostly overlapping between disease and non-disease.

We are selecting a subset of the feature set as using all features run the risk of overfitting the training and validation sets. Using fewer features can speed up inference at the cost of predictive performance.

The features with less overlapping will lead to better model training and performance. Hence, choosing the features: 'cp', 'oldpeak', 'exang', 'slope', 'thalach'.

[CM2]

Correlation coefficient of each pair of features

Heat map is used to find out the correlation between different features in the dataset. High positive or negative value shows that the features have high correlation

[8]:
```
# plotting correlation coeeficients using heat map
#get correlations of each features in dataset
corrmat = df_heart.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
#plot heat map
heart_heat_map=sns.heatmap(df_heart[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```