```
# ! pip install ydata-profiling
# ! pip install -q -U keras-tuner
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Other libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')
from sklearn.ensemble import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
df = pd.read_csv("C:\Users\hp\Downloads\heart-disease.csv")
df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

```
df.shape
```

```
(303, 14)
```

```
df.value_counts()
```

```
age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
29   1    2   130        204   0    2        202      0      0.0      2      0   2     1
34   0    1   118        210   0    1        192      0      0.7      2      0   2     1
35   1    0   120        198   0    1        130      1      1.6      1      0   3     0
37   1    2   130        250   0    1        187      0      3.5      0      0   2     1
     1    3   130        153   0    1        172      0      1.4      2      1   3     1
                                                                                        ...
51   1    2   94         227   0    1        154      1      0.0      2      1   3     1
                           239   0    1        158      1      1.0      1      1   3     1
                           295   0    1        122      1      1.2      1      1   3     0
                           342   0    1        166      1      0.8      2      0   3     1
77   1    0   125        304   0    0        162      1      1.1      2      3   3     0
Name: count, Length: 302, dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.1 KB
```

```
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

```
df.columns
```

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

```
df.isnull().sum()
```

```
age       0
sex       0
cp        0
trestbps  0
chol      0
fbs       0
restecg   0
thalach   0
exang     0
oldpeak   0
slope     0
ca        0
thal      0
target    0
dtype: int64
```

```
# Checking data with disease and no disease
disease = lambda(df) target == 1
no_disease = lambda(df) target == 0
```

```
plt.rcParmas(
fig, ax = plt.subplots(
ax.set_title("Target", fontsize=20, f"No Disease")
ax.pie(no_disease(df).shape[0],
       no_disease(df).shape[0],
       explode=(0, 0.1),
       autopct='%1.1f%%',
       shadow=True,
       colors=["red", "green"]
ax.legend(["Heart Disease", "No Disease"])
ax.set_ylabel(None)
ax.set_xlabel(None)
ax.tick_params(axis='y', which='both', labelleft=False)
fig.show()
# ax.pie(disease(df).shape[0],
# ax.legend(["Heart Disease", "No Disease"])
# ax.tick = 16, 1, 14(rx), color='black', ver='center', fontweight='normal')
plt.show()
```

```
# Obtaining qualitative and quantitative data
qualitative = []
quantitative = []
for feature in df.columns:
    if len(df[feature].unique()) <= 2:
        qualitative.append(feature)
    else:
        quantitative.append(feature)
```

```
qualitative
```

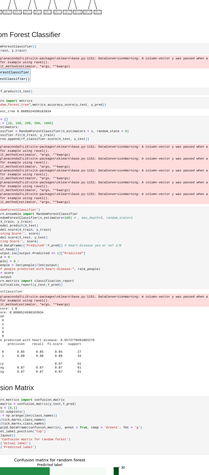
```
['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```

```
quantitative
```

```
['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
```

Correlation Matrix

```
cmap = 'RdBu'
df = df.corr()
table = df[quantitative, 'target'][['target'].index
table = table.sort_values('target')
# sns.heatmap(table, annot=True, cmap='coolwarm')
# sns.heatmap(table.corr, annot=True, cmap='coolwarm', fmt='.2f', annot_kws={'size':12})
plt.title('Top correlated features of dataset', size=14)
plt.show()
```


Top correlated features of dataset
target
oldpeak
slope
exang
thal
ca
trestbps
age
sex
chol
fbs
restecg
thalach
1.00
-0.42
-0.43
-0.44
-0.46
-0.46
-0.46
-0.46
-0.47
-0.47
-0.47
-0.48
-0.49
-0.50
-0.54
-0.61
0.42
1.00
0.12
0.21
0.13
0.09
0.08
0.05
0.07
0.05
0.14
0.04
0.06
0.10
0.09
0.43
0.12
1.00
0.16
0.31
0.12
0.04
0.09
0.05
0.04
0.03
0.09
0.04
0.09
0.09
0.08
0.44
0.21
0.16
1.00
0.12
0.14
0.04
0.09
0.01
0.04
0.01
0.09
0.04
0.09
0.09
0.09
0.46
0.13
0.31
0.12
1.00
0.04
0.01
0.01
0.04
0.06
0.04
0.09
0.04
0.09
0.09
0.09
0.46
0.09
0.12
0.14
0.04
1.00
0.09
0.09
0.09
0.09
0.09
0.09
0.09
0.09
0.09
0.09
0.46
0.08
0.04
0.04
0.09
0.09
1.00
0.44
0.38
0.28
0.12
0.15
0.11
0.12
0.06
0.10
0.19
0.07
0.05
0.09
0.01
0.01
0.09
0.44
1.00
0.38
0.52
0.22
0.15
0.21
0.12
0.04
0.13
0.37
0.06
0.07
0.09
0.04
0.09
0.38
0.38
1.00
0.52
0.19
0.15
0.15
0.21
0.09
0.09
0.28
0.52
0.52
1.00
0.27
0.12
0.21
0.22
0.09
0.09
0.12
0.22
0.27
1.00
0.21
0.32
0.10
0.14
0.05
0.09
0.15
0.15
0.12
0.21
1.00
0.31
0.54
0.43
0.09
0.09
0.21
0.15
0.22
0.32
0.31
1.00
0.61
0.46
0.46
0.09
0.09
0.09
0.09
0.09
0.10
0.54
0.61
1.00
0.46
0.46
0.09
0.09
0.09
0.09
0.09
0.14
0.43
0.46
0.46
1.00
1.00
1.00