Heart Disease Prediction with Machine Learning

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')


from sklearn.neighbors import KNeighborsClassifier


df = pd.read_csv('/content/heart.csv')
print(df.head())
```

```
   age  sex  cp  trestbps  chol  fbs  ...  exang  oldpeak  slope  ca  thal  target
0   63    1   3       145   233    1  ...      0      2.3      0   0     1       1
1   37    1   2       130   250    0  ...      0      3.5      0   0     2       1
2   41    0   1       130   204    0  ...      0      1.4      2   0     2       1
3   56    1   1       120   236    0  ...      0      0.8      2   0     2       1
4   57    0   0       120   354    0  ...      1      0.6      2   0     2       1

[5 rows x 14 columns]
```

```
print(df.info())
```

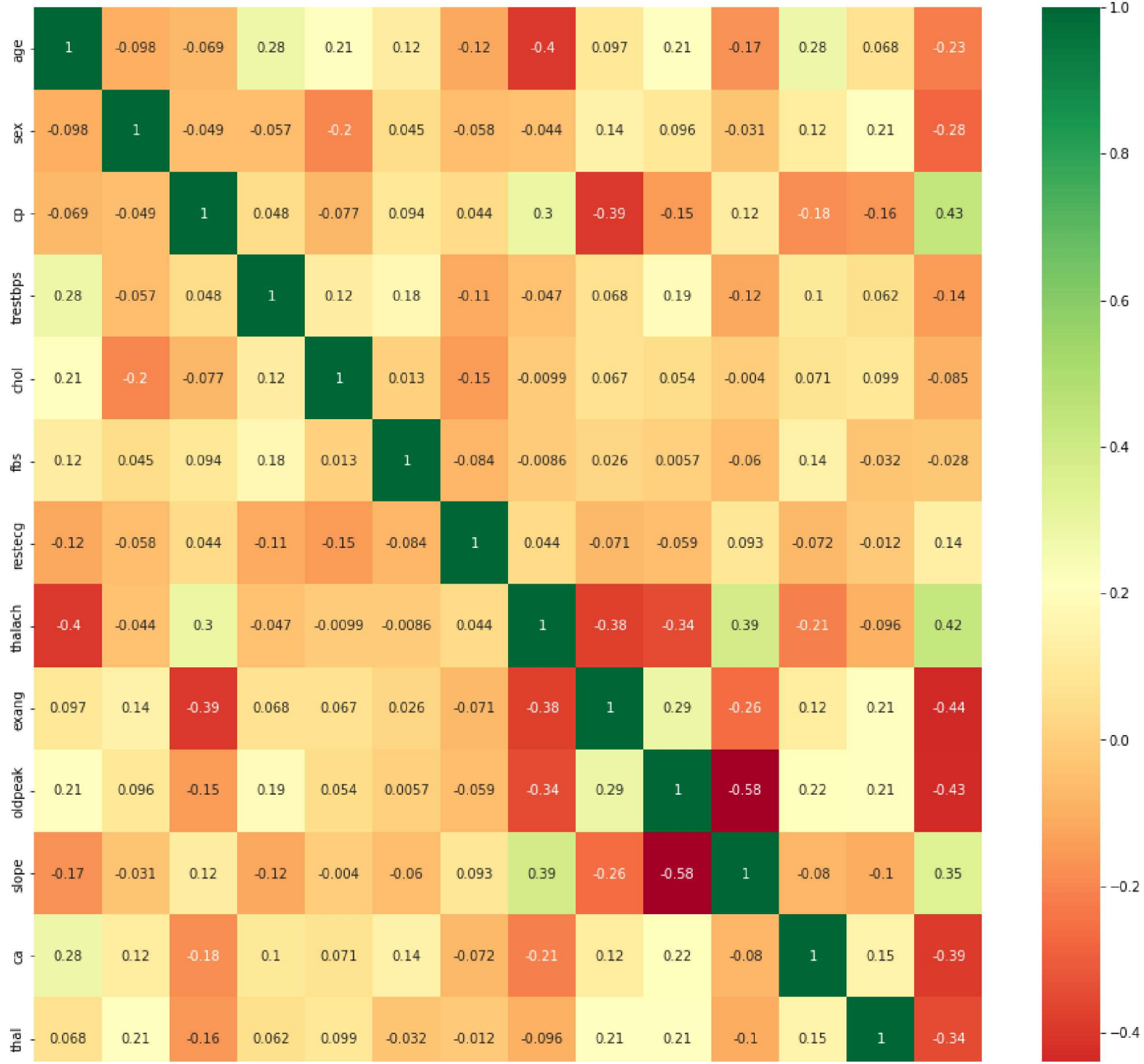
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
```

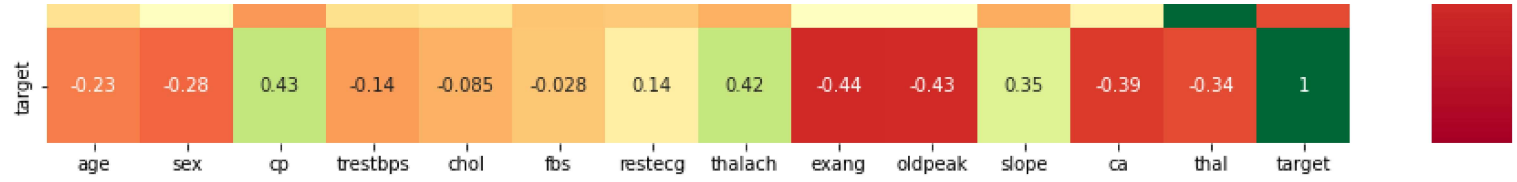
```
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
None
```

```
print(df.describe())
```

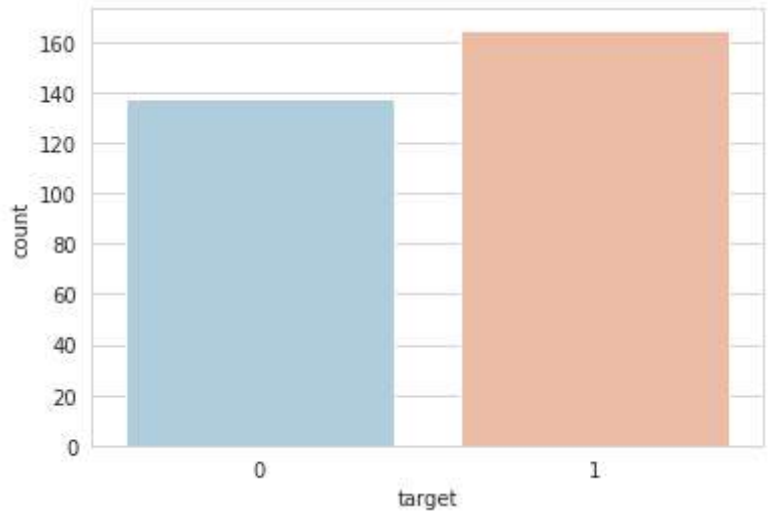
```
              age         sex          cp  ...          ca        thal      target
count  303.000000  303.000000  303.000000  ...  303.000000  303.000000  303.000000
mean    54.366337    0.683168    0.966997  ...    0.729373    2.313531    0.544554
std      9.082101    0.466011    1.032052  ...    1.022606    0.612277    0.498835
min     29.000000    0.000000    0.000000  ...    0.000000    0.000000    0.000000
25%     47.500000    0.000000    0.000000  ...    0.000000    2.000000    0.000000
50%     55.000000    1.000000    1.000000  ...    0.000000    2.000000    1.000000
75%     61.000000    1.000000    2.000000  ...    1.000000    3.000000    1.000000
max     77.000000    1.000000    3.000000  ...    4.000000    3.000000    1.000000

[8 rows x 14 columns]
```

```
import seaborn as sns
corrmat = df.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(16,16))
#plot heat map
g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
plt.show()
```


age 1 -0.098 -0.069 0.28 0.21 0.12 -0.12 -0.4 0.097 0.21 -0.17 0.28 0.068 -0.23
sex -0.098 1 -0.049 -0.057 -0.2 0.045 -0.058 -0.044 0.14 0.096 -0.031 0.12 0.21 -0.28
cp -0.069 -0.049 1 0.048 -0.077 0.094 0.044 0.3 -0.39 -0.15 0.12 -0.18 -0.16 0.43
trestbps 0.28 -0.057 0.048 1 0.12 0.18 -0.11 -0.047 0.068 0.19 -0.12 0.1 0.062 -0.14
chol 0.21 -0.2 -0.077 0.12 1 0.013 -0.15 -0.0099 0.067 0.054 -0.004 0.071 0.099 -0.085
fbs 0.12 0.045 0.094 0.18 0.013 1 -0.084 -0.0086 0.026 0.0057 -0.06 0.14 -0.032 -0.028
restecg -0.12 -0.058 0.044 -0.11 -0.15 -0.084 1 0.044 -0.071 -0.059 0.093 -0.072 -0.012 0.14
thalach -0.4 -0.044 0.3 -0.047 -0.0099 -0.0086 0.044 1 -0.38 -0.34 0.39 -0.21 -0.096 0.42
exang 0.097 0.14 -0.39 0.068 0.067 0.026 -0.071 -0.38 1 0.29 -0.26 0.12 0.21 -0.44
oldpeak 0.21 0.096 -0.15 0.19 0.054 0.0057 -0.059 -0.34 0.29 1 -0.58 0.22 0.21 -0.43
slope -0.17 -0.031 0.12 -0.12 -0.004 -0.06 0.093 0.39 -0.26 -0.58 1 -0.08 -0.1 0.35
ca 0.28 0.12 -0.18 0.1 0.071 0.14 -0.072 -0.21 0.12 0.22 -0.08 1 0.15 -0.39
thal 0.068 0.21 -0.16 0.062 0.099 -0.032 -0.012 -0.096 0.21 0.21 -0.1 0.15 1 -0.34
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4



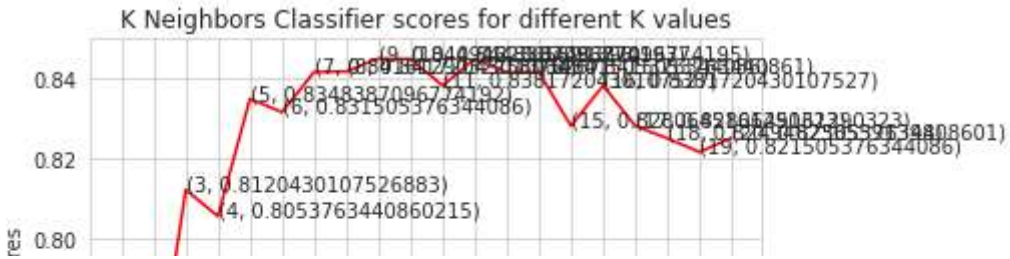

```
sns.set_style('whitegrid')
sns.countplot(x='target',data=df,palette='RdBu_r')
plt.show()
```




```
dataset = pd.get_dummies(df, columns = ['sex', 'cp',
                                        'fbs','restecg',
                                        'exang', 'slope',
                                        'ca', 'thal'])
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
standardScaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dataset[columns_to_scale] = standardScaler.fit_transform(dataset[columns_to_scale])
dataset.head()
```

	age	trestbps	chol	thalach	oldpeak	target	sex_0	sex_1	cp_0	cp_1	cp_2	cp_3	fbs_
0	0.952197	0.763956	-0.256334	0.015443	1.087338	1	0	1	0	0	0	1	
1	-1.915313	-0.092738	0.072199	1.633471	2.122573	1	0	1	0	0	1	0	
2	-1.474158	-0.092738	-0.816773	0.977514	0.310912	1	1	0	0	1	0	0	
3	0.180175	-0.663867	-0.198357	1.239897	-0.206705	1	0	1	0	1	0	0	
4	0.290464	-0.663867	2.082050	0.583939	-0.379244	1	1	0	1	0	0	0	

```
y = dataset['target']
X = dataset.drop(['target'], axis = 1)


from sklearn.model_selection import cross_val_score
knn_scores = []
for k in range(1,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    score=cross_val_score(knn_classifier,X,y,cv=10)
    knn_scores.append(score.mean())


plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
for i in range(1,21):
    plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
plt.xticks([i for i in range(1, 21)])
plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Scores')
plt.title('K Neighbors Classifier scores for different K values')
plt.show()
```




```
knn_classifier = KNeighborsClassifier(n_neighbors = 12)
score=cross_val_score(knn_classifier,X,y,cv=10)
score.mean()
```

```
0.8448387096774195
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

```
from sklearn.ensemble import RandomForestClassifier
randomforest_classifier= RandomForestClassifier(n_estimators=10)
score=cross_val_score(randomforest_classifier,X,y,cv=10)
score.mean()
```

```
0.8343010752688171
```

https://colab.research.google.com/drive/1XXYceCxTGZZpCPqoIJA4dR8VareBRZSS?usp=sharing

✓ 0s completed at 9:26 AM