

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
import warnings
```

```
In [2]: df = pd.read_csv("Dataset.csv")
df.head()
```

| | 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 |

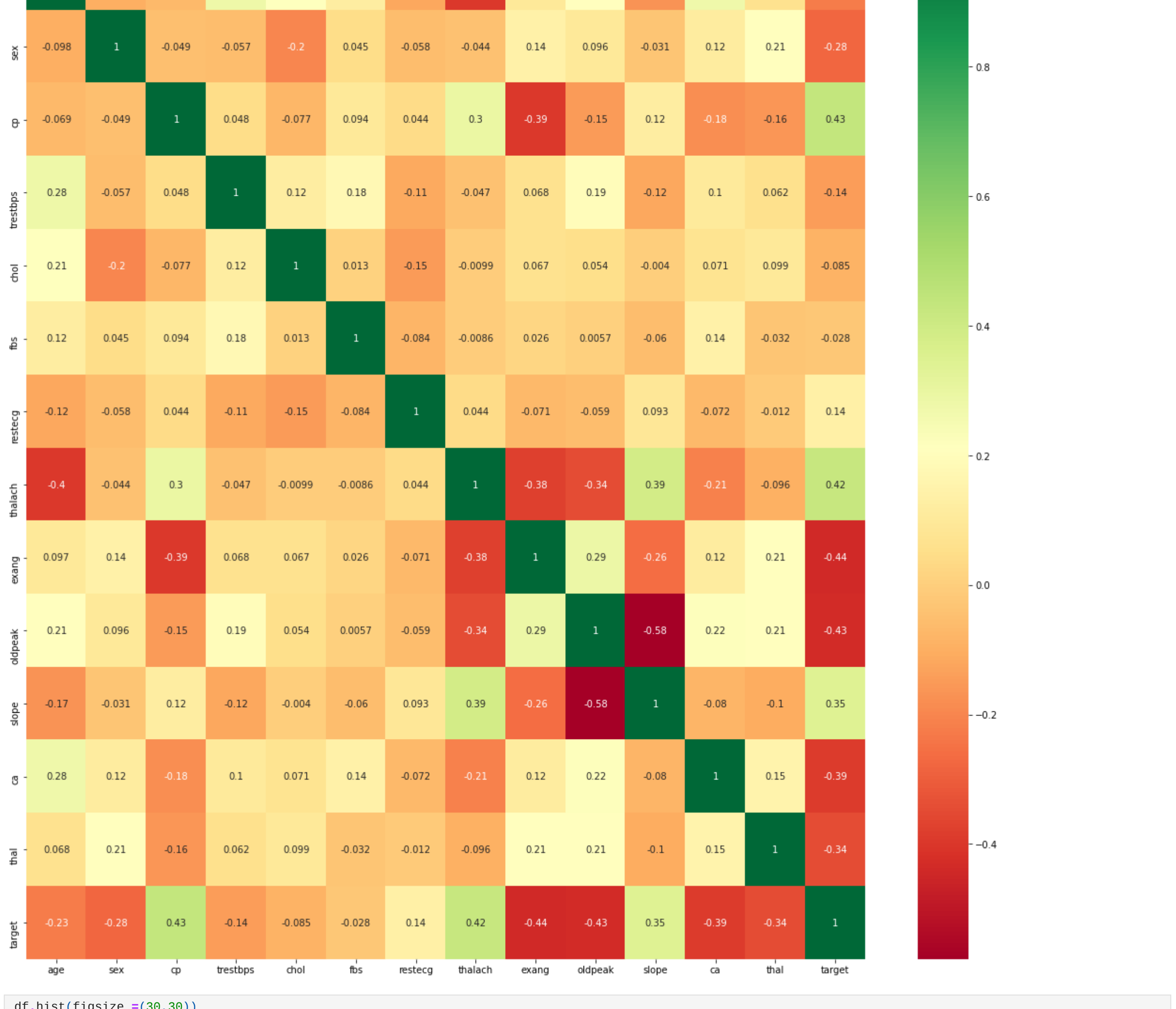
```
In [3]: df.info()

<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
```

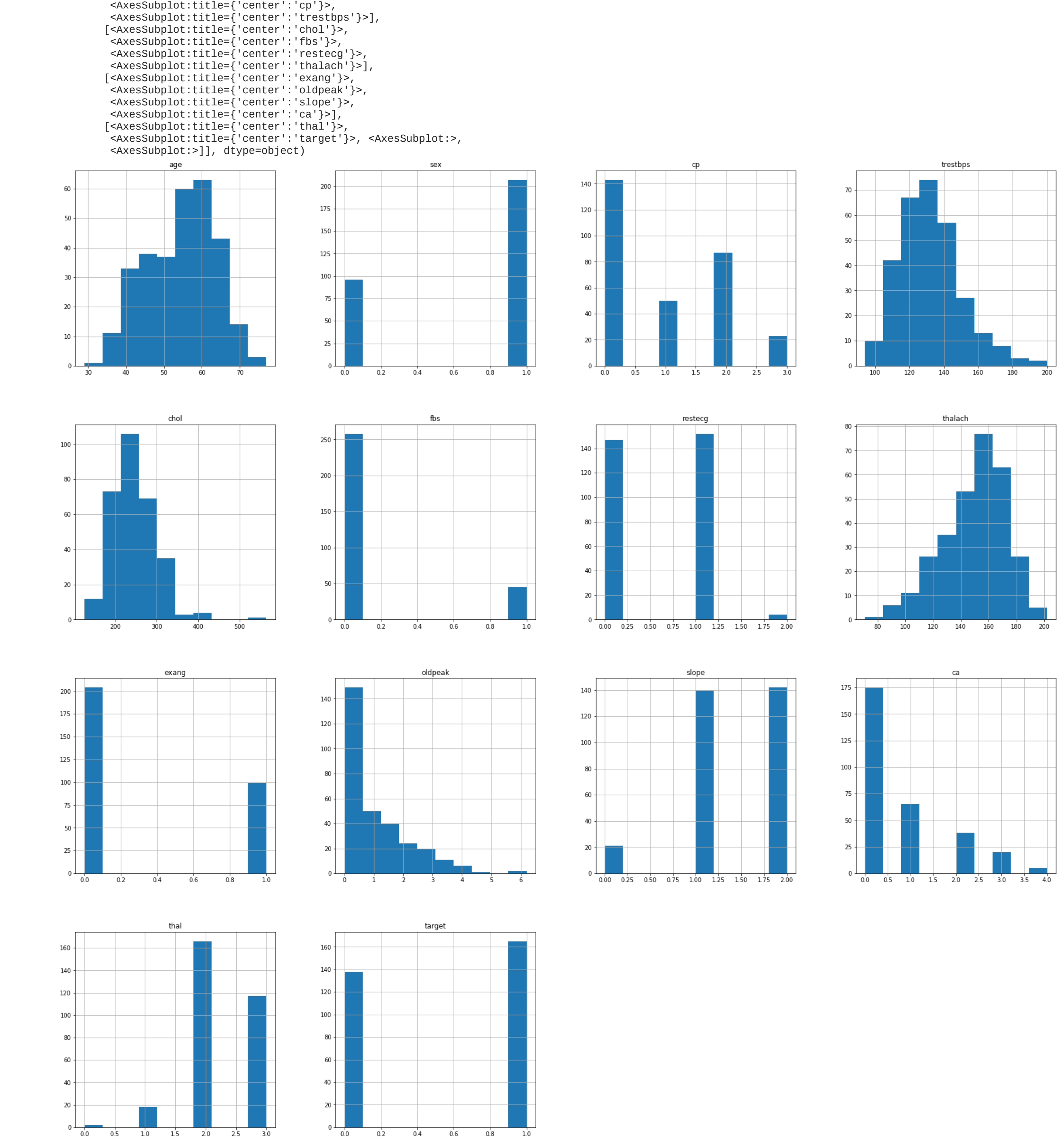
```
In [5]: df.describe()
```

| | 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 | 0.883168 | 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 |

```
In [6]: import seaborn as sns
corrmat = df.corr()
plt.figure(figsize = (20,20))
top = corrmat.index
g = sns.heatmap(df[top].corr(), annot = True, cmap = 'RdYlGn')
```

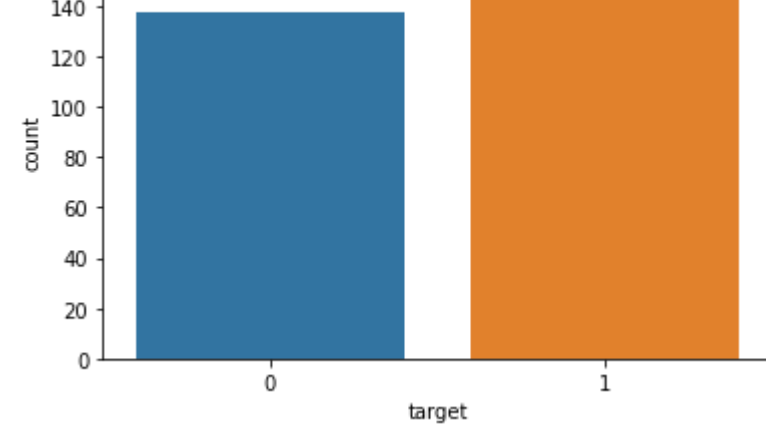


```
In [8]: df.hist(figsize =(30,30))
```



```
In [10]: sns.countplot(x = 'target', data = df)

Out[10]: <AxesSubplot:xlabel='target', ylabel='count'>
```



```
In [11]: dataset = pd.get_dummies(df, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal'])

In [14]: dataset.head()
```

| | age | trestbps | chol | thalach | oldpeak | target | sex_0 | sex_1 | cp_0 | cp_1 | ... | slope_2 | ca_0 | ca_1 | ca_2 | ca_3 | ca_4 | thal_0 | thal_1 | thal_2 | thal_3 |
|---|-----|----------|------|---------|---------|--------|-------|-------|------|------|-----|---------|------|------|------|------|------|--------|--------|--------|--------|
| 0 | 63 | 145 | 233 | 150 | 2.3 | 1 | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 37 | 130 | 250 | 187 | 3.5 | 1 | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 41 | 130 | 204 | 172 | 1.4 | 1 | 1 | 0 | 0 | 1 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 56 | 120 | 236 | 178 | 0.8 | 1 | 0 | 1 | 0 | 1 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 57 | 120 | 354 | 163 | 0.6 | 1 | 1 | 0 | 1 | 0 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

5 rows × 31 columns

```
In [15]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dataset[columns_to_scale] = scaler.fit_transform(dataset[columns_to_scale])

In [17]: dataset.head()
```

| | age | trestbps | chol | thalach | oldpeak | target | sex_0 | sex_1 | cp_0 | cp_1 | ... | slope_2 | ca_0 | ca_1 | ca_2 | ca_3 | ca_4 | thal_0 | thal_1 | thal_2 | thal_3 |
|---|-----------|-----------|-----------|----------|-----------|--------|-------|-------|------|------|-----|---------|------|------|------|------|------|--------|--------|--------|--------|
| 0 | 0.952197 | 0.763956 | -0.256334 | 0.015443 | 1.087338 | 1 | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | -1.915313 | -0.092738 | 0.072199 | 1.633471 | 2.122573 | 1 | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | -1.474158 | -0.092738 | -0.816773 | 0.977514 | 0.310912 | 1 | 1 | 0 | 0 | 1 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0.180175 | -0.663867 | -0.198357 | 1.239897 | -0.206705 | 1 | 0 | 1 | 0 | 1 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0.290464 | -0.663867 | 2.082050 | 0.583939 | -0.379244 | 1 | 1 | 0 | 1 | 0 | ... | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

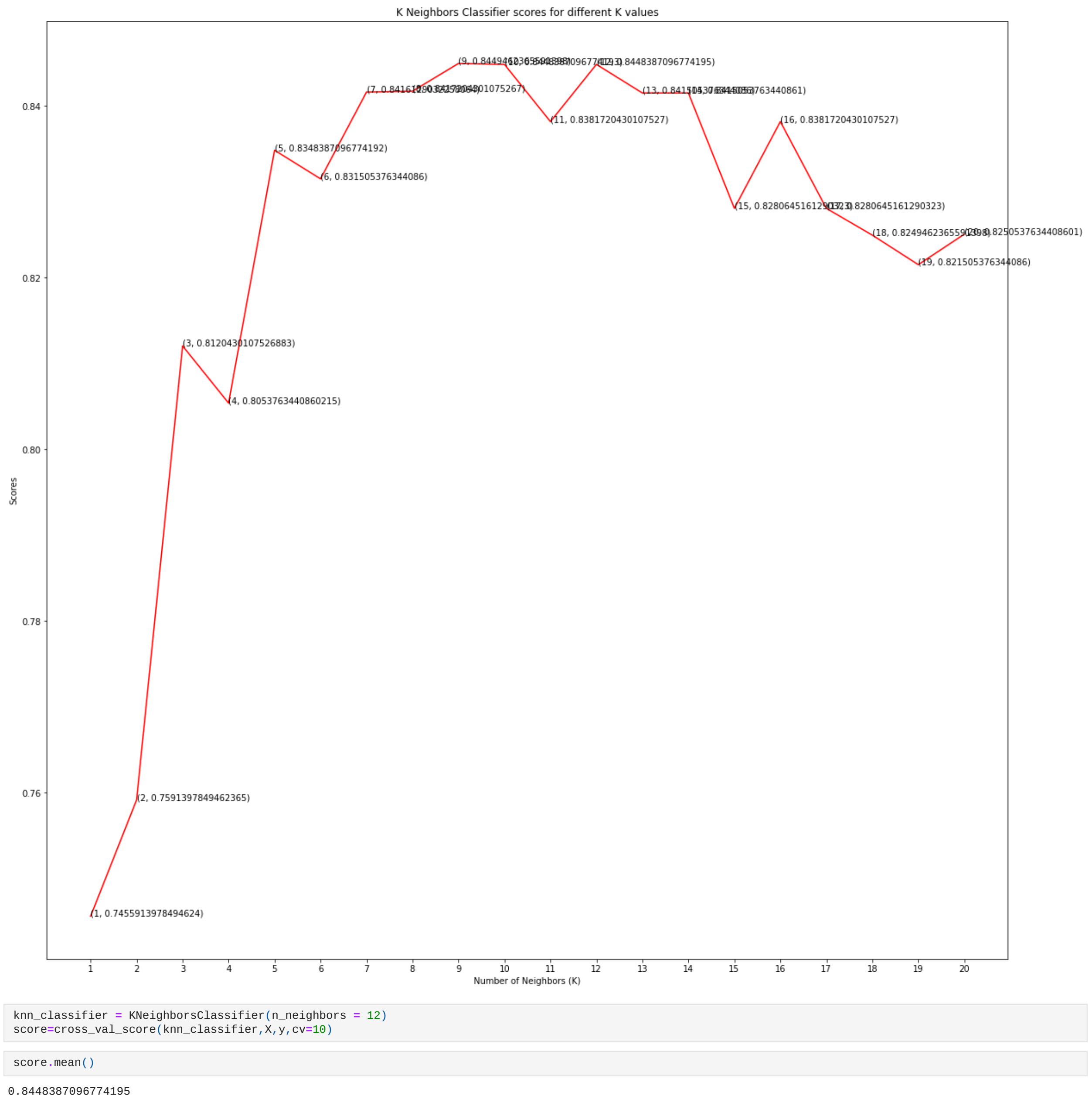
5 rows × 31 columns

```
In [18]: y = dataset['target']
X = dataset.drop(['target'], axis = 1)

In [29]: from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
knn_scores = []
for k in range(1,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    scores=cross_val_score(knn_classifier,X,y,cv=10)
    knn_scores.append(scores.mean())
```

```
In [34]: plt.figure(figsize =(20,20))
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')
```

Out[34]: Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')



```
In [31]: knn_classifier = KNeighborsClassifier(n_neighbors = 12)
score=cross_val_score(knn_classifier,X,y,cv=10)

In [23]: score.mean()

Out[23]: 0.8448387096774195

In [24]: from sklearn.ensemble import RandomForestClassifier
randomforest_classifier= RandomForestClassifier(n_estimators=13)
score=cross_val_score(randomforest_classifier,X,y,cv=10)

In [25]: score.mean()

Out[25]: 0.831505376344086

In [ ]: 
```