

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

```
In [3]: from sklearn.neighbors import KNeighborsClassifier
```

```
In [4]: df = pd.read_csv('dataset.csv')
```

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

Out[5]:

	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 [6]: 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 [7]: `df.describe()`

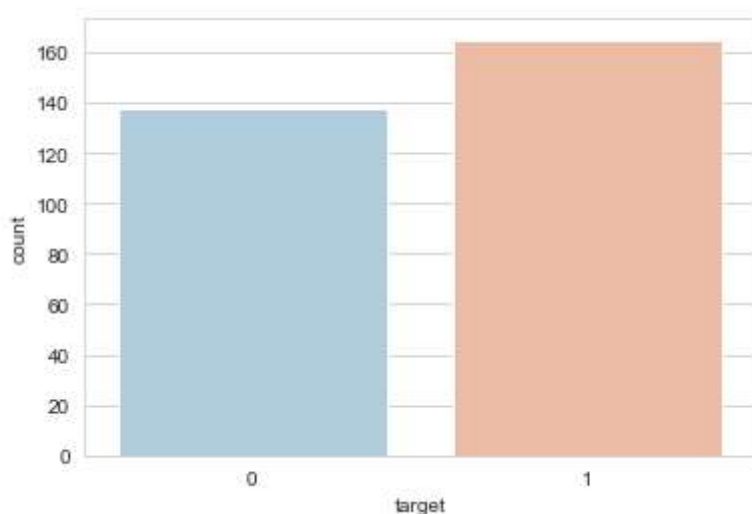
Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thal
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.00
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.64
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.90
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.00
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.50
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.00
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.00
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.00

In [8]: `import seaborn as sns`

In [9]: `sns.set_style('whitegrid')`
`sns.countplot(x='target', data=df, palette='RdBu_r')`

Out[9]: `<matplotlib.axes._subplots.AxesSubplot at 0x136bf9184c0>`



In [10]: `dataset = pd.get_dummies(df, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'oldpeak'])`

In [11]: `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])`

In [12]: `dataset.head()`

Out[12]:

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

5 rows × 31 columns



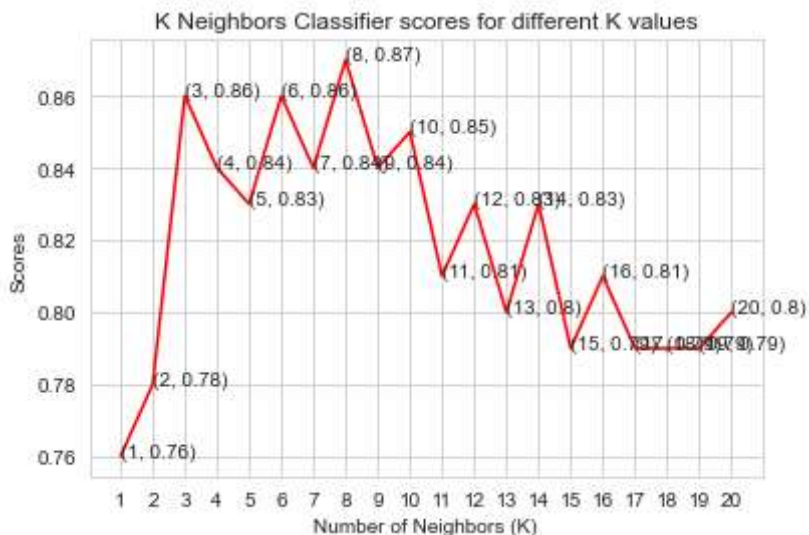
In [13]: `y = dataset['target']`
`X = dataset.drop(['target'], axis = 1)`

In [15]: `y = dataset['target']`
`X = dataset.drop(['target'], axis = 1)`
`X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)`

In [16]: `knn_scores = []`
`for k in range(1,21):`
 `knn_classifier = KNeighborsClassifier(n_neighbors = k)`
 `knn_classifier.fit(X_train, y_train)`
 `knn_scores.append(knn_classifier.score(X_test, y_test))`

```
In [17]: 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[17]: Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')



```
In [22]: print("The score for K Neighbors Classifier is {}% with {} nieghbors.".format(knn_scores[7], 8))
```

The score for K Neighbors Classifier is 87.0% with 8 nieghbors.

In []: