In [23]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
```

In [2]:

```
os.chdir('C:\\Users\\ANIL YADAV\\Downloads')
data = pd.read_csv("heart_disease_uci.csv")
data
```

Out[2]:

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	thal	num
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150.0	False	2.3	downsloping	0.0	fixed defect	0
1	2	67	Male	Cleveland	asymptomatic	160.0	286.0	False	lv hypertrophy	108.0	True	1.5	flat	3.0	normal	2
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129.0	True	2.6	flat	2.0	reversable defect	1
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False	normal	187.0	False	3.5	downsloping	0.0	normal	0
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172.0	False	1.4	upsloping	0.0	normal	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
915	916	54	Female	VA Long Beach	asymptomatic	127.0	333.0	True	st-t abnormality	154.0	False	0.0	NaN	NaN	NaN	1
916	917	62	Male	VA Long Beach	typical angina	NaN	139.0	False	st-t abnormality	NaN	NaN	NaN	NaN	NaN	NaN	0
917	918	55	Male	VA Long Beach	asymptomatic	122.0	223.0	True	st-t abnormality	100.0	False	0.0	NaN	NaN	fixed defect	2
918	919	58	Male	VA Long Beach	asymptomatic	NaN	385.0	True	lv hypertrophy	NaN	NaN	NaN	NaN	NaN	NaN	0
919	920	62	Male	VA Long Beach	atypical angina	120.0	254.0	False	lv hypertrophy	93.0	True	0.0	NaN	NaN	NaN	1

920 rows × 16 columns

In [9]:

```
#dropping num records having values 2,3,4
data.drop(data[data['num'] == 4].index, inplace = True)
data.nunique()
data['target']=data['num']
```

In [10]:

```
data.nunique()
```

Out[10]:

```
id          676
age          49
sex           2
dataset       4
cp            4
trestbps     55
chol        200
fbs           2
restecg       3
thalch      107
exang         2
oldpeak      43
slope         3
ca            4
thal          3
num           2
target        2
dtype: int64
```

In [11]:

```
#dropping num
data.drop(['num'], axis=1)
```

Out[11]:

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	thal	target
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150.0	False	2.3	downsloping	0.0	fixed defect	0
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129.0	True	2.6	flat	2.0	reversable defect	1
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False	normal	187.0	False	3.5	downsloping	0.0	normal	0
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172.0	False	1.4	upsloping	0.0	normal	0
5	6	56	Male	Cleveland	atypical angina	120.0	236.0	False	normal	178.0	False	0.8	upsloping	0.0	normal	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
913	914	62	Male	VA Long Beach	asymptomatic	158.0	170.0	False	st-t abnormality	138.0	True	0.0	NaN	NaN	NaN	1
915	916	54	Female	VA Long Beach	asymptomatic	127.0	333.0	True	st-t abnormality	154.0	False	0.0	NaN	NaN	NaN	1
916	917	62	Male	VA Long Beach	typical angina	NaN	139.0	False	st-t abnormality	NaN	NaN	NaN	NaN	NaN	NaN	0
918	919	58	Male	VA Long Beach	asymptomatic	NaN	385.0	True	lv hypertrophy	NaN	NaN	NaN	NaN	NaN	NaN	0
919	920	62	Male	VA Long Beach	atypical angina	120.0	254.0	False	lv hypertrophy	93.0	True	0.0	NaN	NaN	NaN	1

676 rows × 16 columns

In [12]:

```
def str_features_to_numeric(data):
    # Transforms all string features of the df to numeric features

    # Determination categorical features
    categorical_columns = []
    numerics = ['int8', 'int16', 'int32', 'int64', 'float16', 'float32', 'float64']
    features = data.columns.values.tolist()
    for col in features:
        if data[col].dtype in numerics: continue
        categorical_columns.append(col)

    # Encoding categorical features
    for col in categorical_columns:
        if col in data.columns:
            le = LabelEncoder()
            le.fit(list(data[col].astype(str).values))
            data[col] = le.transform(list(data[col].astype(str).values))

    return data
```

In [13]:

```
from sklearn.preprocessing import LabelEncoder
data = str_features_to_numeric(data)
data
```

Out[13]:

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	thal	num	target
0	1	63	1	0	3	145.0	233.0	1	0	150.0	0	2.3	0	0.0	0	0	0
2	3	67	1	0	0	120.0	229.0	0	0	129.0	1	2.6	1	2.0	3	1	1
3	4	37	1	0	2	130.0	250.0	0	2	187.0	0	3.5	0	0.0	2	0	0
4	5	41	0	0	1	130.0	204.0	0	0	172.0	0	1.4	3	0.0	2	0	0
5	6	56	1	0	1	120.0	236.0	0	2	178.0	0	0.8	3	0.0	2	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
913	914	62	1	3	0	158.0	170.0	0	3	138.0	1	0.0	2	NaN	1	1	1
915	916	54	0	3	0	127.0	333.0	1	3	154.0	0	0.0	2	NaN	1	1	1
916	917	62	1	3	3	NaN	139.0	0	3	NaN	2	NaN	2	NaN	1	0	0
918	919	58	1	3	0	NaN	385.0	1	0	NaN	2	NaN	2	NaN	1	0	0
919	920	62	1	3	1	120.0	254.0	0	0	93.0	1	0.0	2	NaN	1	1	1

676 rows × 17 columns

In [14]:

```
data.isnull().sum()
```

Out[14]:

```
id            0
age           0
sex           0
dataset       0
cp            0
trestbps     33
chol         26
fbs           0
restecg       0
thalch       33
exang         0
oldpeak      36
slope         0
ca          453
thal          0
num           0
target        0
dtype: int64
```

In [19]:

```
data.trestbps=data.trestbps.fillna(data['trestbps'].median())

data.chol=data.chol.fillna(data['chol'].median())

data.fbs=data.fbs.fillna('False')

data.restecg=data.restecg.fillna('normal')

data.thalch=data.thalch.fillna(data['thalch'].mean())

data.exang=data.exang.fillna('False')

data.oldpeak=data.oldpeak.fillna(data['oldpeak'].mean())

data.slope=data.slope.fillna('flat')
data.ca=data.ca.fillna(data['ca'].median())
```

In [20]:

```
data.isnull().sum()
```

Out[20]:

```
id          0
age         0
sex         0
dataset     0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalch      0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
num         0
target      0
dtype: int64
```

In [21]:

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

In [22]:

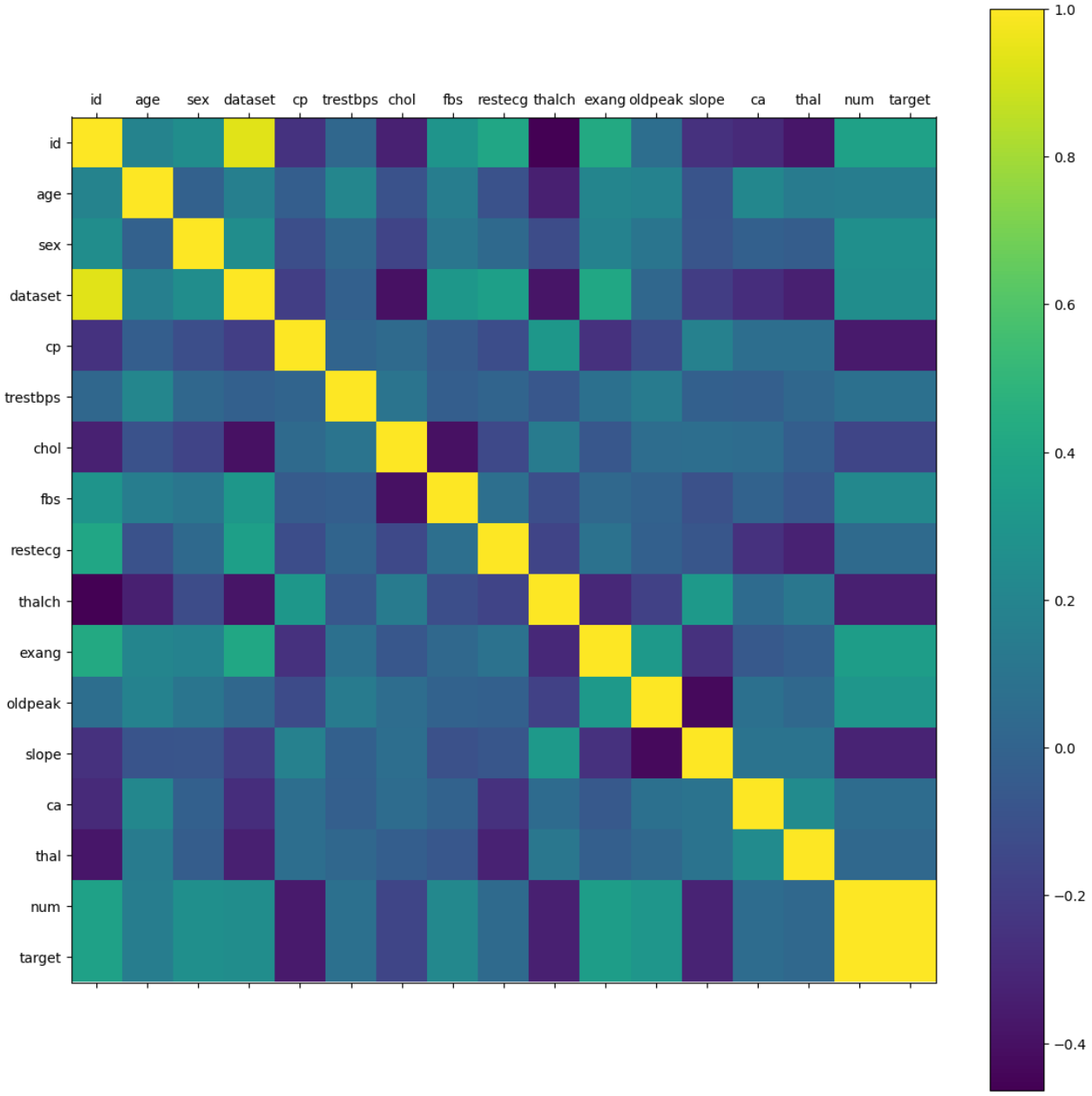
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

In [26]:

```
rcParams['figure.figsize'] = 20, 14
plt.matshow(data.corr())
plt.yticks(np.arange(data.shape[1]), data.columns)
plt.xticks(np.arange(data.shape[1]), data.columns)
plt.colorbar()
```

Out[26]:

```
<matplotlib.colorbar.Colorbar at 0x285adae1c40>
```

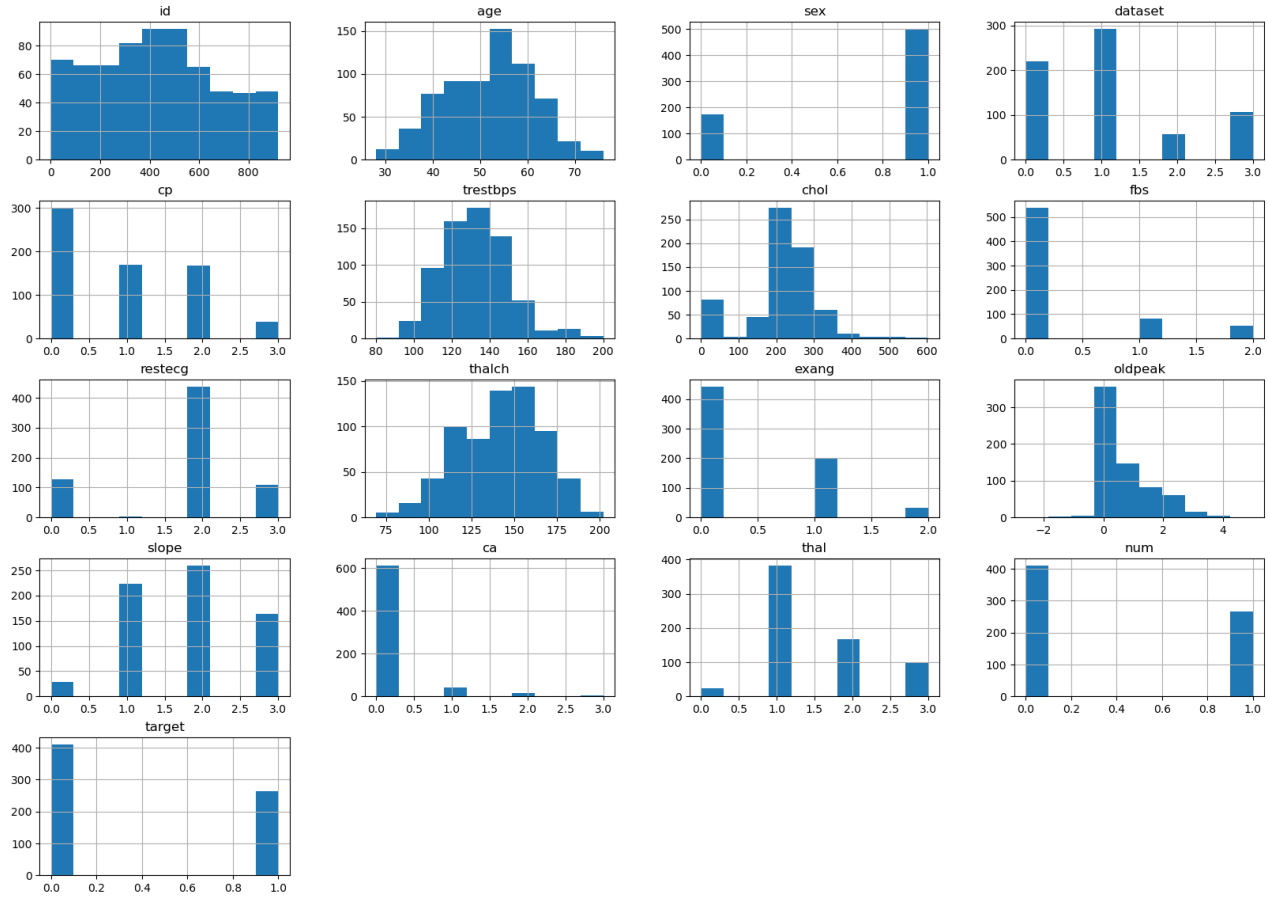


In [27]:

```
data.hist()
```

Out[27]:

```
array([[<AxesSubplot:title={'center':'id'}>,
        <AxesSubplot:title={'center':'age'}>,
        <AxesSubplot:title={'center':'sex'}>,
        <AxesSubplot:title={'center':'dataset'}>],
       [<AxesSubplot:title={'center':'cp'}>,
        <AxesSubplot:title={'center':'trestbps'}>,
        <AxesSubplot:title={'center':'chol'}>,
        <AxesSubplot:title={'center':'fbs'}>],
       [<AxesSubplot:title={'center':'restecg'}>,
        <AxesSubplot:title={'center':'thalch'}>,
        <AxesSubplot:title={'center':'exang'}>,
        <AxesSubplot:title={'center':'oldpeak'}>],
       [<AxesSubplot:title={'center':'slope'}>,
        <AxesSubplot:title={'center':'ca'}>,
        <AxesSubplot:title={'center':'thal'}>,
        <AxesSubplot:title={'center':'num'}>],
       [<AxesSubplot:title={'center':'target'}>, <AxesSubplot:>,
        <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```

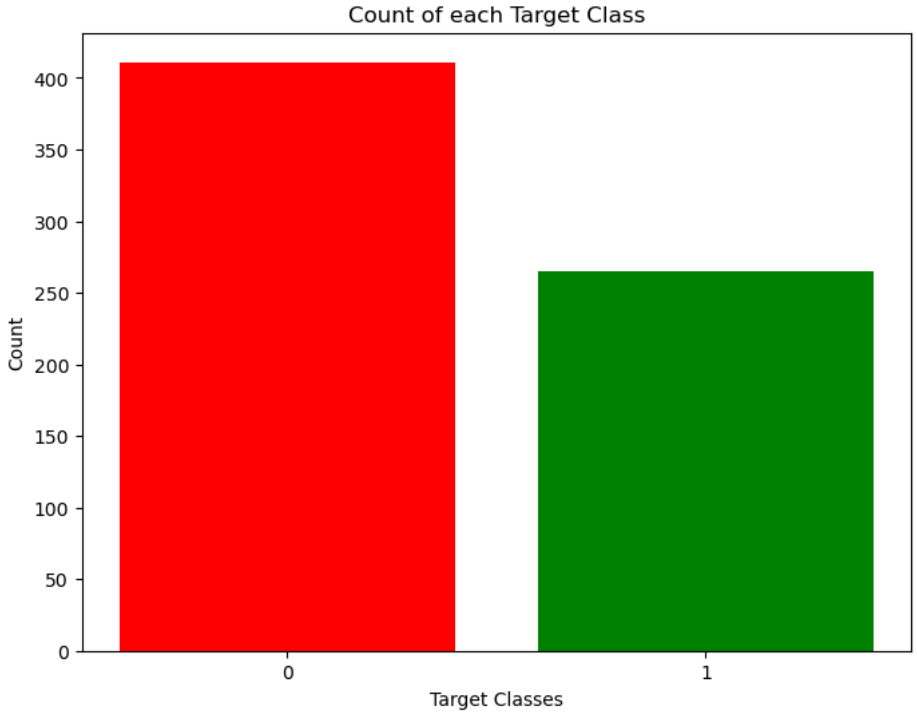


In [28]:

```
rcParams['figure.figsize'] = 8,6
plt.bar(data['target'].unique(), data['target'].value_counts(), color = ['red', 'green'])
plt.xticks([0, 1])
plt.xlabel('Target Classes')
plt.ylabel('Count')
plt.title('Count of each Target Class')
```

Out[28]:

```
Text(0.5, 1.0, 'Count of each Target Class')
```



In [29]:

```
data = pd.get_dummies(data, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal'])
```

In [31]:

```
standardScaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalch', 'oldpeak']
data[columns_to_scale] = standardScaler.fit_transform(data[columns_to_scale])
```

In [33]:

```
y = data['target']
X = data.drop(['target'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)
```

In [34]:

```
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))
```

```
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
```

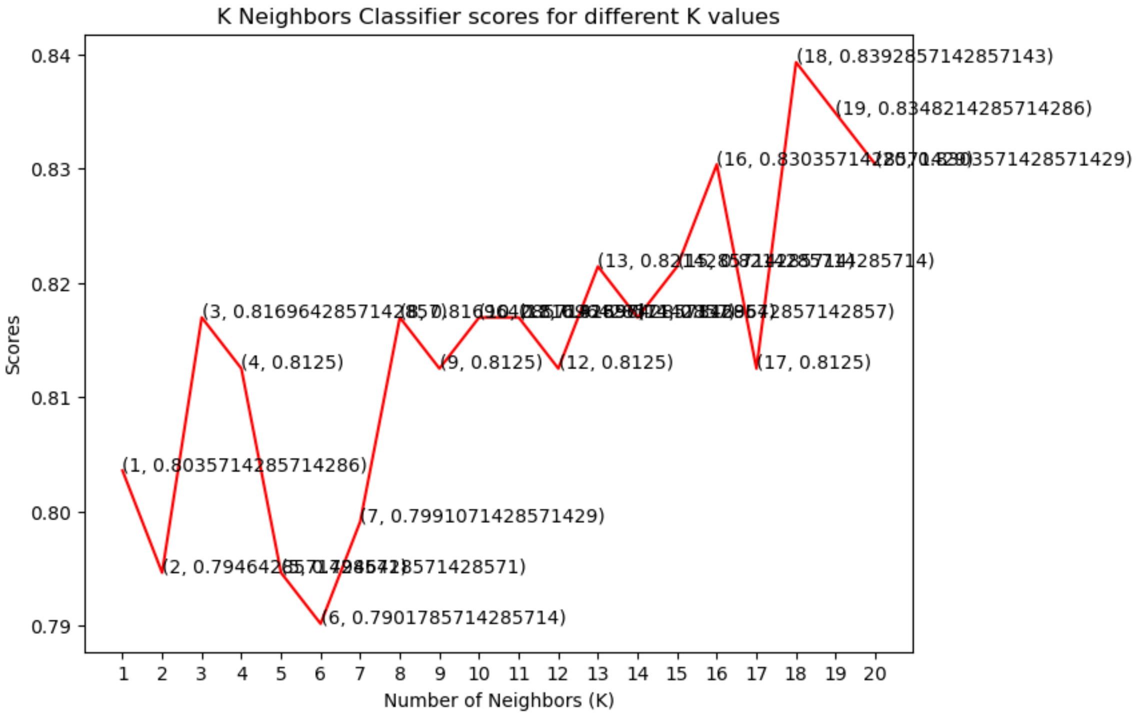
```
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\ANIL YADAV\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduc
tion functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In Sci
Py 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statisti
c is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this
warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

In [35]:

```
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[35]:

```
Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')
```



In [36]:

```
print("The score for K Neighbors Classifier is {}% with {} nieghbors.".format(knn_scores[7]*100, 8))
```

```
The score for K Neighbors Classifier is 81.69642857142857% with 8 nieghbors.
```

In [ ]: