

In [1]: `#EXP -9`

In [2]: `#Aim:To perform and analysis of KNN Algorithm`

In [3]: `# Name:Payal Devanand Manwar
Roll no.: 37
Sec:A
Subject:ET1
Date: 29/09/2025`

KNN CLASSIFIER

In [9]: `import pandas as pd
import numpy as np`

In [10]: `import os`

In [11]: `os.getcwd()`

Out[11]: 'C:\\Users\\USER'

In [12]: `os.chdir("C:\\Users\\USER\\Desktop")`

In [13]: `data=pd.read_csv("heart - heart.csv")`

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

Out[14]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [15]: `data.tail()`

Out[15]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

```
In [16]: x=data.drop("target", axis=1)
         y=data["target"]
```

```
In [17]: from sklearn.neighbors import KNeighborsClassifier
         from sklearn.metrics import accuracy_score
```

```
In [18]: #splitting the data into training and testing data sets
         from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2 ,random_state=42)
```

```
In [19]: knn=KNeighborsClassifier()
```

```
In [20]: knn.fit(x_train, y_train)
```

```
Out[20]: KNeighborsClassifier()
```

```
In [21]: y_pred2=knn.predict(x_test)
```

```
In [22]: accuracy = accuracy_score(y_test, y_pred2)
```

```
In [23]: accuracy
```

```
Out[23]: 0.7317073170731707
```

```
In [24]: x_train
```

```
Out[24]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
835	49	1	2	118	149	0	0	126	0	0.8	2	3	2
137	64	0	0	180	325	0	1	154	1	0.0	2	0	2
534	54	0	2	108	267	0	0	167	0	0.0	2	0	2
495	59	1	0	135	234	0	1	161	0	0.5	1	0	3
244	51	1	2	125	245	1	0	166	0	2.4	1	0	2
...
700	41	1	2	130	214	0	0	168	0	2.0	1	0	2
71	61	1	0	140	207	0	0	138	1	1.9	2	1	3
106	51	1	0	140	299	0	1	173	1	1.6	2	0	3
270	43	1	0	110	211	0	1	161	0	0.0	2	0	3

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
860	52	1	0	112	230	0	1	160	0	0.0	2	1	2

820 rows × 13 columns

In [25]:

x_test

Out[25]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
527	62	0	0	124	209	0	1	163	0	0.0	2	0	2
359	53	0	2	128	216	0	0	115	0	0.0	2	0	0
447	55	1	0	160	289	0	0	145	1	0.8	1	1	3
31	50	0	1	120	244	0	1	162	0	1.1	2	0	2
621	48	1	0	130	256	1	0	150	1	0.0	2	2	3
...
832	68	1	2	118	277	0	1	151	0	1.0	2	1	3
796	41	1	1	135	203	0	1	132	0	0.0	1	0	1
644	44	1	2	120	226	0	1	169	0	0.0	2	0	2
404	61	1	0	140	207	0	0	138	1	1.9	2	1	3
842	58	1	2	112	230	0	0	165	0	2.5	1	1	3

205 rows × 13 columns

In [26]:

y_train

Out[26]:

```

835    0
137    1
534    1
495    1
244    1
..
700    1
71     0
106    0
270    1
860    0
Name: target, Length: 820, dtype: int64

```

In [27]:

y_test

Out[27]:

```

527    1
359    1
447    0
31     1
621    0
..
832    1
796    1
644    1
404    0

```

842 0

Name: target, Length: 205, dtype: int64

In [28]:

```
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Generate random numbers
data = np.random.randint(1, 20, size=8) # 8 random numbers between 1 and 20

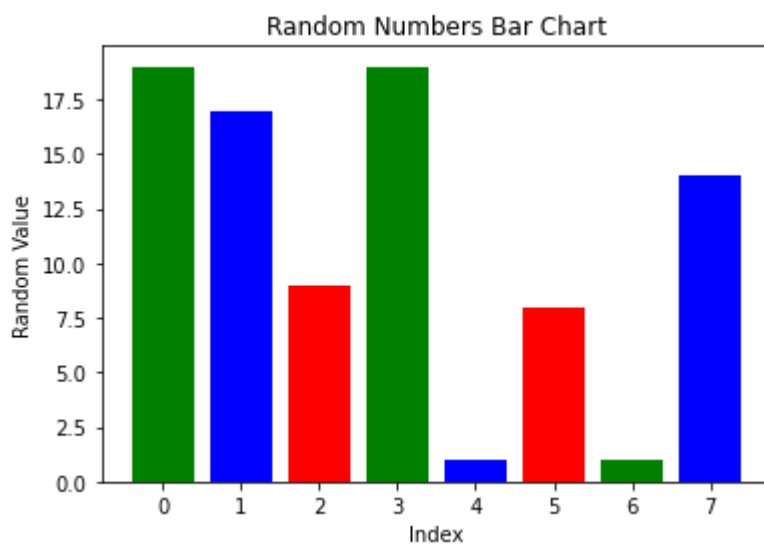
# Step 2: Create bar chart
x = np.arange(len(data)) # positions for bars

# Step 3: Change bar colors
colors = ['green', 'blue', 'red']

plt.bar(x, data, color=colors)

# Add Labels
plt.xlabel("Index")
plt.ylabel("Random Value")
plt.title("Random Numbers Bar Chart")

# Show plot
plt.show()
```



In []: