Importing Dataset of Covid-19

In [1]:
```
import pandas as pd

d1 = pd.read_csv(r'country_wise_latest.csv')
d1
```

Out[1]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Region
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	69.49	5.04	35526	737	2.07	Eastern Mediterranean
1	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	5.25	4171	709	17.00	Europe
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	6.17	23691	4282	18.07	Africa
3	Andorra	907	52	803	52	10	0	0	5.73	88.53	6.48	884	23	2.60	Europe
4	Angola	950	41	242	667	18	1	0	4.32	25.47	16.94	749	201	26.84	Africa
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	78	3752	6791	152	2	0	0.73	35.33	2.08	8916	1705	19.12	Eastern Mediterranean
183	Western Sahara	10	1	8	1	0	0	0	10.00	80.00	12.50	10	0	0.00	Africa
184	Yemen	1691	483	833	375	10	4	36	28.56	49.26	57.98	1619	72	4.45	Eastern Mediterranean
185	Zambia	4552	140	2815	1597	71	1	465	3.08	61.84	4.97	3326	1226	36.86	Africa
186	Zimbabwe	2704	36	542	2126	192	2	24	1.33	20.04	6.64	1713	991	57.85	Africa

187 rows × 15 columns

Modifing The Dataset and Cleaning it

In [282]:
```
x = d1.drop(columns = ['Deaths','WHO Region','1 week % increase','1 week change','Confirmed last week','Deaths / 100 Recovered','Recovered /
yd = d1['Deaths']
yr = d1['Recovered']
y = d1['Country/Region']
h = d1.drop(columns = ['WHO Region','1 week % increase','1 week change','Confirmed last week','Deaths / 100 Recovered','Recovered / 100 Cases
h
```

Out[282]:

	Country/Region	Deaths
0	Afghanistan	1269
1	Albania	144
2	Algeria	1163
3	Andorra	52
4	Angola	41
...	...	...
182	West Bank and Gaza	78
183	Western Sahara	1
184	Yemen	483
185	Zambia	140
186	Zimbabwe	36

187 rows × 2 columns

Creating the Model

In [163]:
```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
modeld = DecisionTreeClassifier()
modelr = DecisionTreeClassifier()

model.fit(x.values,y)
modeld.fit(x.values,yd)
modelr.fit(x.values,yr)

d1
```

Out[163]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Region
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	69.49	5.04	35526	737	2.07	Eastern Mediterranean
1	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	5.25	4171	709	17.00	Europe
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	6.17	23691	4282	18.07	Africa
3	Andorra	907	52	803	52	10	0	0	5.73	88.53	6.48	884	23	2.60	Europe
4	Angola	950	41	242	667	18	1	0	4.32	25.47	16.94	749	201	26.84	Africa
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	78	3752	6791	152	2	0	0.73	35.33	2.08	8916	1705	19.12	Eastern Mediterranean
183	Western Sahara	10	1	8	1	0	0	0	10.00	80.00	12.50	10	0	0.00	Africa
184	Yemen	1691	483	833	375	10	4	36	28.56	49.26	57.98	1619	72	4.45	Eastern Mediterranean
185	Zambia	4552	140	2815	1597	71	1	465	3.08	61.84	4.97	3326	1226	36.86	Africa
186	Zimbabwe	2704	36	542	2126	192	2	24	1.33	20.04	6.64	1713	991	57.85	Africa

187 rows × 15 columns

Training The Model

In [72]:
```
p = model.predict([ [36000,9700],[4800,1990] ])
pd = modeld.predict([ [36000,9700] ])
pr = modelr.predict([ [36000,9700] ])

pr
```

Out[72]:
```
array([25198], dtype=int64)
```

In [88]:
```
from sklearn.model_selection import train_test_split as tts
x_train , x_test , y_train, y_test = tts(x,yd,test_size=0.2)

model.fit(x_train,y_train)
p = model.predict(x_test)
p
```

Out[88]:
```
array([    0, 35112,  2760, 30212,   294,   255,   408,    15,   294,
          15,    34,    11,     0,  1945,   474,  1676,     1,  1978,
         373,     0,     2,  1761,   393,    48,    14,    15,     0,
         255,    53,   294,    91,     2,  1676,  4656,     0, 33408,
           5,   474], dtype=int64)
```

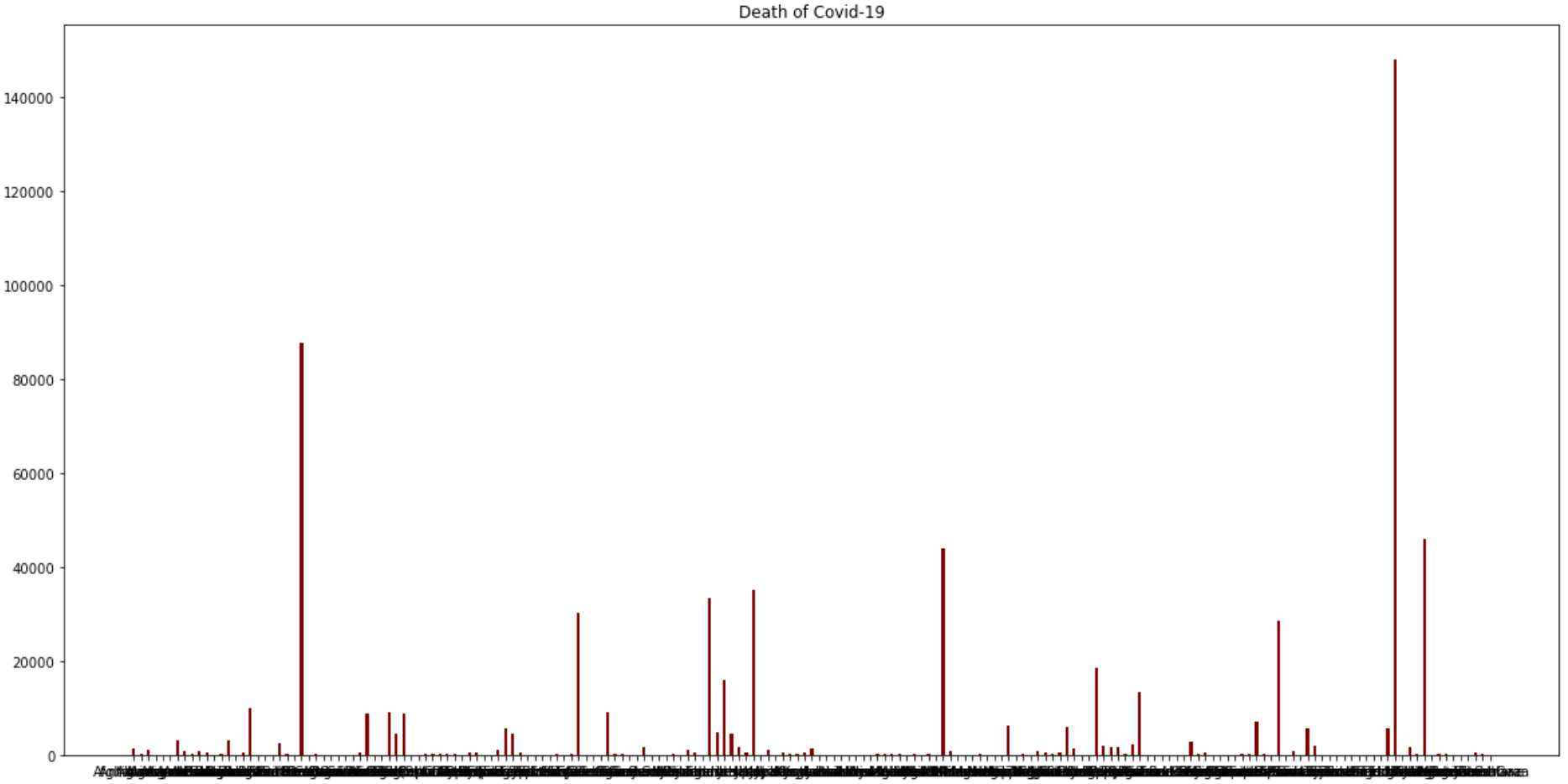
Accuracy Score

In [279]:
```
from sklearn.metrics import accuracy_score as acc

x_train , x_test , y_train, y_test = tts(x,yd,test_size=0.1)

modeld.fit(x_train,y_train)
p = model.predict(x_test)

score = acc(y_test,p)
score
```

Out[279]:
```
0.8947368421052632
```

In [ ]:
```

```

In [294]:
```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure(figsize = (20, 10))

plt.bar(d1['Country/Region'], d1['Deaths'], color ='maroon',width = 0.4)

plt.title("Death of Covid-19")
plt.show()
```



In [ ]: