

```
In [1]: import pandas as pd
import numpy as np
import pickle
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: data=pd.read_csv(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis an
```

```
In [3]: data
```

Out[3]:

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	...	Gastroint
0	1	77	M	Yes	Yes	Yes	Yes	Yes	No	No	...	
1	2	38	F	Yes	Yes	Yes	Yes	No	Yes	Yes	...	
2	3	75	F	Yes	Yes	Yes	Yes	Yes	Yes	Yes	...	
3	4	84	F	Yes	Yes	Yes	No	No	Yes	No	...	
4	5	66	M	Yes	Yes	Yes	Yes	Yes	No	Yes	...	
...	
5429	5430	68	F	Yes	Yes	No	Yes	Yes	Yes	Yes	...	
5430	5431	22	F	Yes	Yes	Yes	No	Yes	Yes	No	...	
5431	5432	78	F	Yes	Yes	Yes	No	No	No	No	...	
5432	5433	55	M	Yes	Yes	Yes	No	Yes	No	No	...	
5433	5434	33	M	Yes	Yes	Yes	No	Yes	Yes	No	...	

5434 rows × 25 columns



```
In [4]: data.describe()
```

```
Out[4]:
```

	User_id	Age
count	5434.000000	5434.000000
mean	2717.500000	50.550975
std	1568.805012	23.310484
min	1.000000	11.000000
25%	1359.250000	30.000000
50%	2717.500000	50.000000
75%	4075.750000	71.000000
max	5434.000000	90.000000

In [5]: `data.shape`

Out[5]: (5434, 25)

In [6]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5434 entries, 0 to 5433
Data columns (total 25 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   User_id                                   5434 non-null   int64
 1   Age                                       5434 non-null   int64
 2   Sex                                       5434 non-null   object
 3   Breathing Problem                       5434 non-null   object
 4   Fever                                   5434 non-null   object
 5   Dry Cough                               5434 non-null   object
 6   Sore throat                             5434 non-null   object
 7   Running Nose                            5434 non-null   object
 8   Asthma                                  5434 non-null   object
 9   Chronic Lung Disease                    5434 non-null   object
10   Headache                                5434 non-null   object
11   Heart Disease                           5434 non-null   object
12   Diabetes                                 5434 non-null   object
13   Hyper Tension                           5434 non-null   object
14   Fatigue                                 5434 non-null   object
15   Gastrointestinal                        5434 non-null   object
16   Abroad travel                           5434 non-null   object
17   Contact with COVID Patient              5434 non-null   object
18   Attended Large Gathering                 5434 non-null   object
19   Visited Public Exposed Places            5434 non-null   object
20   Family working in Public Exposed Places  5434 non-null   object
21   Wearing Masks                           5434 non-null   object
22   Sanitization from Market                 5434 non-null   object
23   COVID-19                                5434 non-null   object
24   SARS-Cov-2 exam result                  5434 non-null   object
dtypes: int64(2), object(23)
memory usage: 1.0+ MB
```

In [7]: `data.head(10)`

Out[7]:

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	...	Gastrointesti
0	1	77	M	Yes	Yes	Yes	Yes	Yes	No	No	...	
1	2	38	F	Yes	Yes	Yes	Yes	No	Yes	Yes	...	
2	3	75	F	Yes	Yes	Yes	Yes	Yes	Yes	Yes	...	
3	4	84	F	Yes	Yes	Yes	No	No	Yes	No	...	
4	5	66	M	Yes	Yes	Yes	Yes	Yes	No	Yes	...	
5	6	25	M	Yes	Yes	Yes	No	No	No	No	...	
6	7	80	F	Yes	Yes	Yes	No	No	No	Yes	...	

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	...	Gastrointesti
7	8	71	F	Yes	Yes	Yes	No	Yes	Yes	No	...	
8	9	67	F	Yes	Yes	Yes	No	Yes	No	Yes	...	
9	10	73	F	Yes	Yes	Yes	No	No	Yes	No	...	

10 rows × 25 columns

In [8]:

```
data.tail(5)
```

Out[8]:

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	...	Gastroint
5429	5430	68	F	Yes	Yes	No	Yes	Yes	Yes	Yes	...	
5430	5431	22	F	Yes	Yes	Yes	No	Yes	Yes	No	...	
5431	5432	78	F	Yes	Yes	Yes	No	No	No	No	...	
5432	5433	55	M	Yes	Yes	Yes	No	Yes	No	No	...	
5433	5434	33	M	Yes	Yes	Yes	No	Yes	Yes	No	...	

5 rows × 25 columns



In [9]:

```
data.isna().sum()
```

Out[9]:

```
User_id      0
Age          0
Sex          0
Breathing Problem  0
Fever        0
Dry Cough    0
Sore throat  0
Running Nose 0
Asthma       0
Chronic Lung Disease 0
Headache     0
Heart Disease 0
Diabetes     0
Hyper Tension 0
Fatigue      0
Gastrointestinal 0
Abroad travel 0
Contact with COVID Patient 0
Attended Large Gathering 0
Visited Public Exposed Places 0
Family working in Public Exposed Places 0
```

```

Wearing Masks          0
Sanitization from Market 0
COVID-19                0
SARS-Cov-2 exam result  0
dtype: int64

```

```
In [10]: data.groupby(['COVID-19']).count()
```

```
Out[10]:
```

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	...	Fatig
COVID-19												
No	1051	1051	1051	1051	1051	1051	1051	1051	1051	1051	...	10
Yes	4383	4383	4383	4383	4383	4383	4383	4383	4383	4383	...	43

2 rows × 24 columns



```
In [11]: data.groupby(['SARS-Cov-2 exam result']).count()
```

```
Out[11]:
```

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	...	Fati
SARS-Cov-2 exam result												
negative	1051	1051	1051	1051	1051	1051	1051	1051	1051	1051	...	1
positive	4383	4383	4383	4383	4383	4383	4383	4383	4383	4383	...	4

2 rows × 24 columns



```
In [12]: data.groupby(['Contact with COVID Patient']).count()
```

Out[12]:

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	...	Fatig
Contact with COVID Patient												
No	2708	2708	2708	2708	2708	2708	2708	2708	2708	2708	...	27
Yes	2726	2726	2726	2726	2726	2726	2726	2726	2726	2726	...	27

2 rows × 24 columns

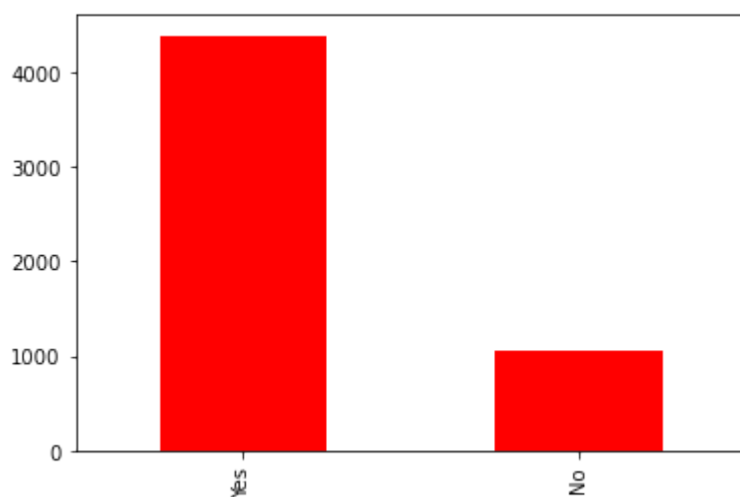


In [13]:

```
import seaborn as sns
import matplotlib.pyplot as plt
data['COVID-19'].value_counts().head(10).plot(kind='bar', color='r')
```

Out[13]:

<AxesSubplot:>

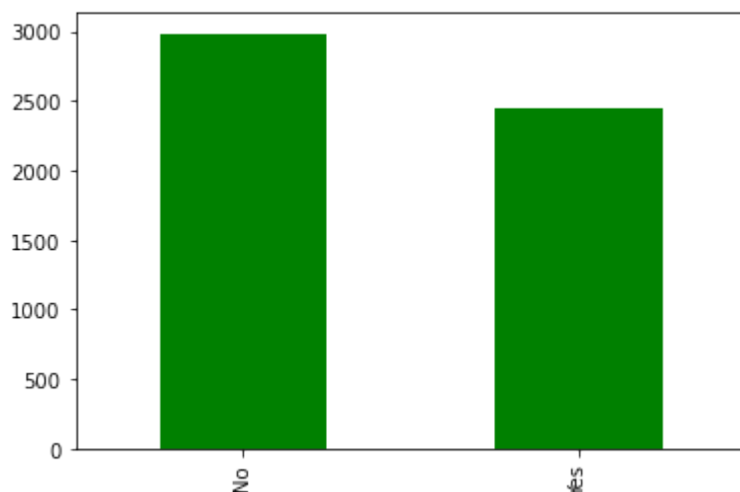


In [14]:

```
data['Abroad travel'].value_counts().head(10).plot(kind='bar', color='g')
```

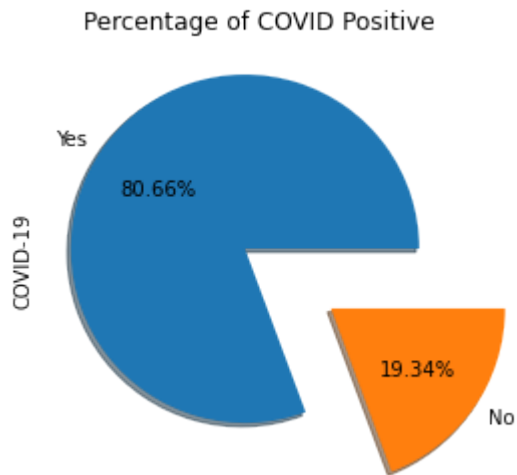
Out[14]:

<AxesSubplot:>



```
In [15]: data["COVID-19"].value_counts().plot.pie(explode=[0.1,0.5],autopct='%1.2f%%',shadow=
plt.title('Percentage of COVID Positive')
```

```
Out[15]: Text(0.5, 1.0, 'Percentage of COVID Positive')
```



```
In [16]: data1=data.drop(['Asthma','Chronic Lung Disease','Fatigue ','Headache','Diabetes','G
data1
```

```
Out[16]:
```

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Heart Disease	Hyper Tension	Abroad travel	Con , CC Pat
0	1	77	M	Yes	Yes	Yes	Yes	Yes	No	Yes	No	
1	2	38	F	Yes	Yes	Yes	Yes	No	No	No	No	
2	3	75	F	Yes	Yes	Yes	Yes	Yes	No	No	Yes	
3	4	84	F	Yes	Yes	Yes	No	No	Yes	No	Yes	
4	5	66	M	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	
...	
5429	5430	68	F	Yes	Yes	No	Yes	Yes	No	No	No	
5430	5431	22	F	Yes	Yes	Yes	No	Yes	No	Yes	No	
5431	5432	78	F	Yes	Yes	Yes	No	No	Yes	Yes	No	
5432	5433	55	M	Yes	Yes	Yes	No	Yes	Yes	No	No	
5433	5434	33	M	Yes	Yes	Yes	No	Yes	No	Yes	No	

5434 rows × 19 columns



```
In [17]: data1.head(10)
```

Out[17]:

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Heart Disease	Hyper Tension	Abroad travel	Contact with COVID Patient
0	1	77	M	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
1	2	38	F	Yes	Yes	Yes	Yes	No	No	No	No	No
2	3	75	F	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No
3	4	84	F	Yes	Yes	Yes	No	No	Yes	No	Yes	No
4	5	66	M	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
5	6	25	M	Yes	Yes	Yes	No	No	Yes	Yes	No	No
6	7	80	F	Yes	Yes	Yes	No	No	Yes	Yes	No	No
7	8	71	F	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No
8	9	67	F	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes
9	10	73	F	Yes	Yes	Yes	No	No	No	Yes	No	No

In [18]:

```
data1['Breathing Problem']=data1['Breathing Problem'].map({'Yes':1,'No':0})
data1['Fever']=data1['Fever'].map({'Yes':1,'No':0})
data1['Running Nose']=data1['Running Nose'].map({'Yes':1,'No':0})
data1['Heart Disease']=data1['Heart Disease'].map({'Yes':1,'No':0})
data1['Dry Cough']=data1['Dry Cough'].map({'Yes':1,'No':0})
data1['Sore throat']=data1['Sore throat'].map({'Yes':1,'No':0})
data1['Hyper Tension']=data1['Hyper Tension'].map({'Yes':1,'No':0})
data1['Abroad travel']=data1['Abroad travel'].map({'Yes':1,'No':0})
data1['Contact with COVID Patient']=data1['Contact with COVID Patient'].map({'Yes':1,'No':0})
data1['Attended Large Gathering']=data1['Attended Large Gathering'].map({'Yes':1,'No':0})
data1['Wearing Masks']=data1['Wearing Masks'].map({'Yes':1,'No':0})
data1['Visited Public Exposed Places']=data1['Visited Public Exposed Places'].map({'Yes':1,'No':0})
data1['Sanitization from Market']=data1['Sanitization from Market'].map({'Yes':1,'No':0})
data1['Family working in Public Exposed Places']=data1['Family working in Public Exposed Places'].map({'Yes':1,'No':0})
data1['COVID-19']=data1['COVID-19'].map({'Yes':1,'No':0})
data1['SARS-Cov-2 exam result']=data1['SARS-Cov-2 exam result'].map({'positive':1,'negative':0})
data1['Sex']=data1['Sex'].map({'F':1,'M':0})
```

In [19]:

```
data1.head()
```

Out[19]:

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Heart Disease	Hyper Tension	Abroad travel	Contact with COVID Patient
0	1	77	0	1	1	1	1	1	0	1	0	1
1	2	38	1	1	1	1	1	0	0	0	0	0
2	3	75	1	1	1	1	1	1	0	0	1	0
3	4	84	1	1	1	1	0	0	1	0	1	0
4	5	66	0	1	1	1	1	1	1	1	0	1

In [20]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5434 entries, 0 to 5433
Data columns (total 19 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   User_id                                   5434 non-null   int64
1   Age                                       5434 non-null   int64
2   Sex                                       5434 non-null   int64
3   Breathing Problem                       5434 non-null   int64
4   Fever                                    5434 non-null   int64
5   Dry Cough                               5434 non-null   int64
6   Sore throat                             5434 non-null   int64
7   Running Nose                            5434 non-null   int64
8   Heart Disease                           5434 non-null   int64
9   Hyper Tension                           5434 non-null   int64
10  Abroad travel                            5434 non-null   int64
11  Contact with COVID Patient               5434 non-null   int64
12  Attended Large Gathering                 5434 non-null   int64
13  Visited Public Exposed Places            5434 non-null   int64
14  Family working in Public Exposed Places  5434 non-null   int64
15  Wearing Masks                           5434 non-null   int64
16  Sanitization from Market                 5434 non-null   int64
17  COVID-19                                5434 non-null   int64
18  SARS-Cov-2 exam result                  5434 non-null   int64
dtypes: int64(19)
memory usage: 806.7 KB
```

In [21]:

```
cor_mat=data1.corr()
cor_mat
```

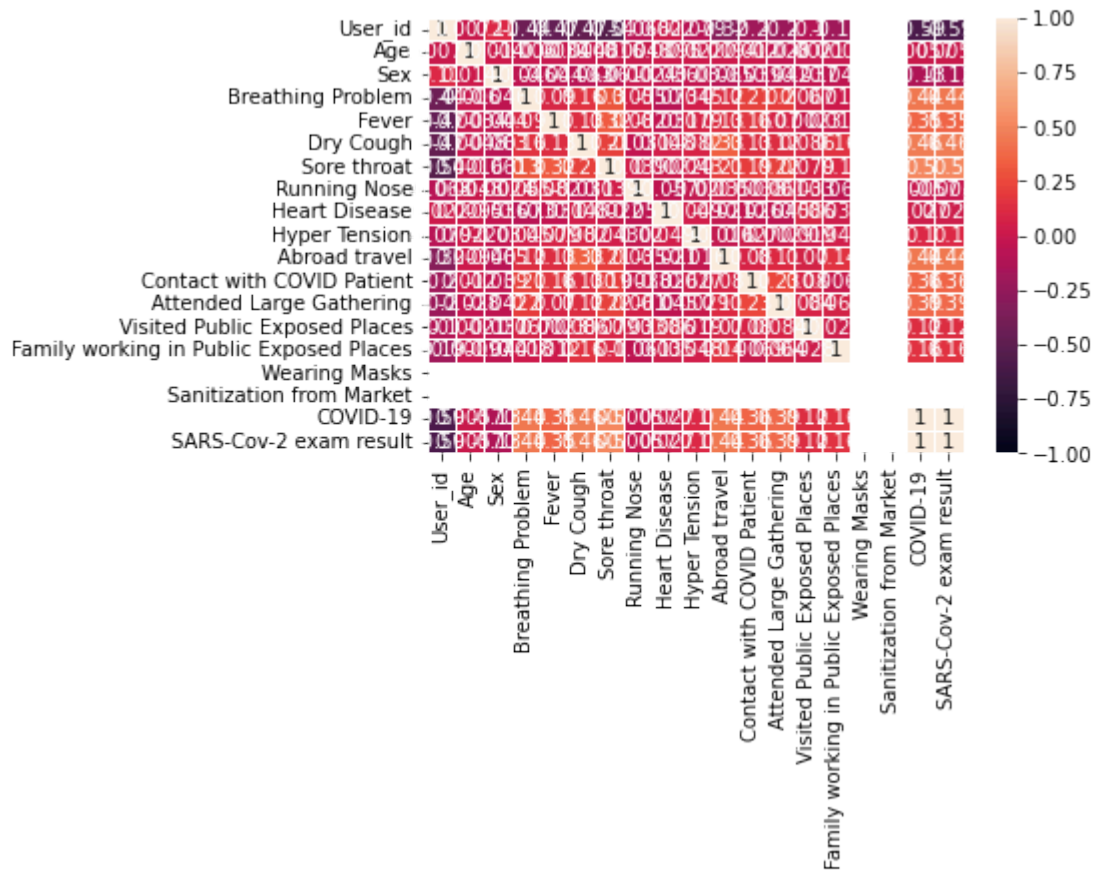
Out[21]:

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	
User_id	1.000000	0.007204	0.107722	-0.437417	-0.468600	-0.472480	-0.542537	-0.068410	(
Age	0.007204	1.000000	0.014471	-0.005952	-0.003358	-0.004772	0.001613	0.004827	(
Sex	0.107722	0.014471	1.000000	-0.045609	-0.044236	-0.063379	-0.060719	-0.002378	-(
Breathing Problem	-0.437417	-0.005952	-0.045609	1.000000	0.089903	0.159562	0.303768	0.055190	-(
Fever	-0.468600	-0.003358	-0.044236	0.089903	1.000000	0.127580	0.322235	0.081758	-(
Dry Cough	-0.472480	-0.004772	-0.063379	0.159562	0.127580	1.000000	0.213907	-0.030763	(
Sore throat	-0.542537	0.001613	-0.060719	0.303768	0.322235	0.213907	1.000000	0.039450	(
Running Nose	-0.068410	0.004827	-0.002378	0.055190	0.081758	-0.030763	0.039450	1.000000	-(
Heart Disease	0.021612	0.009481	-0.036354	-0.073366	-0.031462	0.047566	0.002177	-0.056750	1
Hyper Tension	-0.078966	0.022155	-0.029676	0.045256	0.079001	0.081989	0.042811	-0.020445	(

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	
Abroad travel	-0.318820	0.009418	-0.064556	0.117795	0.128726	0.331418	0.205986	0.034526	-(
Contact with COVID Patient	-0.201967	0.012475	-0.038712	0.214634	0.164704	0.128330	0.189251	0.003776	-(
Attended Large Gathering	-0.203309	-0.028315	-0.042247	0.200304	0.070490	0.117963	0.216438	0.061099	-(
Visited Public Exposed Places	-0.104390	-0.002145	-0.036582	0.066688	0.002252	0.086176	0.079055	0.032568	(
Family working in Public Exposed Places	-0.194553	-0.000966	-0.044399	0.018295	0.012102	0.163102	0.104378	-0.061323	(
Wearing Masks	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Sanitization from Market	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
COVID-19	-0.588240	0.005679	-0.130817	0.443764	0.352891	0.464292	0.502848	-0.005657	(
SARS-Cov-2 exam result	-0.588240	0.005679	-0.130817	0.443764	0.352891	0.464292	0.502848	-0.005657	(

```
In [22]: import seaborn as sns
sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidths=.5,cmap='rocket')
```

```
Out[22]: <AxesSubplot:>
```



Disease Diagnosis

```
In [23]: y=data1['COVID-19']
x=data1.drop(['COVID-19'],axis=1)
```

```
In [24]: x
```

```
Out[24]:
```

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Heart Disease	Hyper Tension	Abroad travel	Con CC Pat
0	1	77	0	1	1	1	1	1	0	1	0	
1	2	38	1	1	1	1	1	0	0	0	0	
2	3	75	1	1	1	1	1	1	0	0	1	
3	4	84	1	1	1	1	0	0	1	0	1	
4	5	66	0	1	1	1	1	1	1	1	0	
...	
5429	5430	68	1	1	1	0	1	1	0	0	0	
5430	5431	22	1	1	1	1	0	1	0	1	0	
5431	5432	78	1	1	1	1	0	0	1	1	0	
5432	5433	55	0	1	1	1	0	1	1	0	0	

	User_id	Age	Sex	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Heart Disease	Hyper Tension	Abroad travel	Con ' CC Pat
5433	5434	33	0	1	1	1	0	1	0	1	0	

5434 rows × 18 columns

In [25]:

y

Out[25]:

```

0      1
1      1
2      1
3      1
4      1
..
5429   1
5430   1
5431   0
5432   0
5433   0
Name: COVID-19, Length: 5434, dtype: int64

```

In [26]:

```

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)

```

In [27]:

```

print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)

```

(3803, 18) (1631, 18) (3803,) (1631,)

Logistic Regression

In [28]:

```

from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train, y_train)

```

Out[28]:

```

▼ LogisticRegression
LogisticRegression()

```

In [29]:

```

y_pred1=classifier.predict(x_test)
y_pred1

```

Out[29]:

array([1, 0, 1, ..., 1, 1, 1], dtype=int64)

In [30]:

```

from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred1)

```

Out[30]:

```

array([[ 316,    0],
       [   0, 1315]], dtype=int64)

```

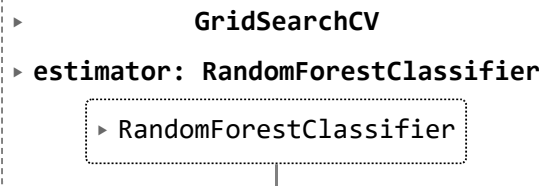
```
In [31]: from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred1)
```

```
Out[31]: 1.0
```

Random Forest Classification

```
In [32]: from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
cls=RandomForestClassifier()
n_estimators=[25,50,75,100,125,150,175,200]
criterion=['gini','entropy']
max_depth=[3,5,10]
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth}
RFC_cls = GridSearchCV(cls, parameters)
RFC_cls.fit(x_train,y_train)
```

```
Out[32]:
```

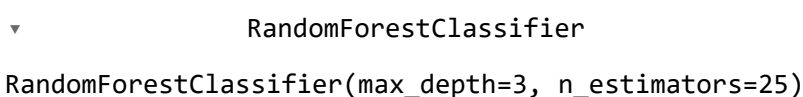


```
In [33]: RFC_cls.best_params_
```

```
Out[33]: {'criterion': 'gini', 'max_depth': 3, 'n_estimators': 25}
```

```
In [34]: cls=RandomForestClassifier(n_estimators=25,criterion='gini',max_depth=3)
cls.fit(x_train,y_train)
```

```
Out[34]:
```



```
In [35]: y_pred2=cls.predict(x_test)
y_pred2
```

```
Out[35]: array([1, 0, 1, ..., 1, 1, 1], dtype=int64)
```

```
In [36]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred2)
```

```
Out[36]: array([[ 316,    0],
       [    0, 1315]], dtype=int64)
```

```
In [37]: from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred2)
```

```
Out[37]: 1.0
```

DecisionTree Classification

```
In [38]: from sklearn.tree import DecisionTreeClassifier
```

```
In [39]: cls=DecisionTreeClassifier()
```

```
In [40]: cls.fit(x_train,y_train)
```

```
Out[40]: ▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [41]: y_pred3=cls.predict(x_test)
y_pred3
```

```
Out[41]: array([1, 0, 1, ..., 1, 1, 1], dtype=int64)
```

```
In [42]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred3)
```

```
Out[42]: array([[ 316,    0],
               [    0, 1315]], dtype=int64)
```

```
In [43]: from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred3)
```

```
Out[43]: 1.0
```

```
In [44]: from prettytable import PrettyTable
table=PrettyTable()
table.field_names = ["TEST_SIZE","0.30"]
table.add_row(["Logistic Regression",1.0])
table.add_row(["Random Forest classification",1.0])
table.add_row(["DecisionTree Classification",1.0])
print(table)
```

```
+-----+-----+
|          TEST_SIZE          | 0.30 |
+-----+-----+
|   Logistic Regression       | 1.0  |
| Random Forest classification | 1.0  |
| DecisionTree Classification | 1.0  |
+-----+-----+
```

Patient Risk Prediction

```
In [45]: new = x_test.iloc[0]
a = np.asarray(new)
a = a.reshape(1,-1)
p = cls.predict(a)
```

```
In [46]: if (p[0] == 1):
          print("Person is affected by Covid 19 and is at risk of dying")
        else:
          print("Great! the results are negative and you don't have to worry")
```

Person is affected by Covid 19 and is at risk of dying

```
In [47]: import tkinter as tk
          from tkinter import ttk
          import pandas as pd

          def load_data_from_csv(file_path):
              try:
                  df = pd.read_csv(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diag
                      return df
              except Exception as e:
                  print(f"Error loading CSV file: {e}")
                  return None

          def predict_risk():
              global data_df
              if data_df is None:
                  prediction_label.config(text="Error: Data not loaded.")
                  return

              input_features = {
                  'Fever': fever_var.get(),
                  'Dry Cough': dry_cough_var.get(),
                  'Sore throat': sore_throat_var.get(),
                  'Running Nose': running_nose_var.get(),
                  'Heart Disease': heart_disease_var.get(),
                  'Hyper Tension': hyper_tension_var.get(),
                  'Abroad travel': abroad_travel_var.get(),
                  'Contact with COVID Patient': covid_contact_var.get(),
                  'Attended Large Gathering': large_gathering_var.get(),
                  'Visited Public Exposed Places': exposed_places_var.get(),
                  'Family working in Public Exposed Places': family_exposed_var.get(),
                  'Wearing Masks': masks_var.get(),
                  'Sanitization from Market': sanitization_var.get()
              }

              try:
                  if input_features['Wearing Masks'] == 1 or input_features['Sanitization fro
                      predicted_risk = "Low"
                  else:
                      features_for_prediction = data_df[['Fever', 'Dry Cough', 'Sore throat',
                          'Heart Disease', 'Hyper Tension', 'Ab
                              'Contact with COVID Patient', 'Attend
                                  'Visited Public Exposed Places', 'Fam
                                      'Wearing Masks', 'Sanitization from M

                      input_df = pd.DataFrame([input_features])
                      df_concat = pd.concat([features_for_prediction, input_df], ignore_index=
                      if df_concat.iloc[-1].any():
                          predicted_risk = "High"
                      else:
                          predicted_risk = "Low"
                      prediction_label.config(text=f"Predicted Risk: {predicted_risk}")

              except Exception as e:
                  print(f"Error predicting risk: {e}")
                  prediction_label.config(text="Error predicting risk.")

          file_path = r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Ris
          data_df = load_data_from_csv(file_path)
```

```
if data_df is None:
    exit()

root = tk.Tk()
root.title("COVID-19 Risk Prediction")

fever_var = tk.IntVar()
dry_cough_var = tk.IntVar()
sore_throat_var = tk.IntVar()
running_nose_var = tk.IntVar()
heart_disease_var = tk.IntVar()
hyper_tension_var = tk.IntVar()
abroad_travel_var = tk.IntVar()
covid_contact_var = tk.IntVar()
large_gathering_var = tk.IntVar()
exposed_places_var = tk.IntVar()
family_exposed_var = tk.IntVar()
masks_var = tk.IntVar()
sanitization_var = tk.IntVar()

fever_check = ttk.Checkbutton(root, text="Fever", variable=fever_var)
fever_check.pack()

dry_cough_check = ttk.Checkbutton(root, text="Dry Cough", variable=dry_cough_var)
dry_cough_check.pack()

sore_throat_check = ttk.Checkbutton(root, text="Sore throat", variable=sore_throat_v
sore_throat_check.pack()

running_nose_check = ttk.Checkbutton(root, text="Running Nose", variable=running_nos
running_nose_check.pack()

heart_disease_check = ttk.Checkbutton(root, text="Heart Disease", variable=heart_dis
heart_disease_check.pack()

hyper_tension_check = ttk.Checkbutton(root, text="Hyper Tension", variable=hyper_ten
hyper_tension_check.pack()

abroad_travel_check = ttk.Checkbutton(root, text="Abroad travel", variable=abroad_tr
abroad_travel_check.pack()

covid_contact_check = ttk.Checkbutton(root, text="Contact with COVID Patient", varia
covid_contact_check.pack()

large_gathering_check = ttk.Checkbutton(root, text="Attended Large Gathering", varia
large_gathering_check.pack()

exposed_places_check = ttk.Checkbutton(root, text="Visited Public Exposed Places", v
exposed_places_check.pack()

family_exposed_check = ttk.Checkbutton(root, text="Family working in Public Exposed
family_exposed_check.pack()

masks_check = ttk.Checkbutton(root, text="Wearing Masks", variable=masks_var)
masks_check.pack()

sanitization_check = ttk.Checkbutton(root, text="Sanitization from Market", variable
sanitization_check.pack()

predict_button = ttk.Button(root, text="Predict Risk", command=predict_risk)
predict_button.pack()

prediction_label = ttk.Label(root, text="")
```

```
prediction_label.pack()
root.mainloop()
```

Medical Image Analysis

In [48]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
import os
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [49]:

```
for dirname, _, filenames in os.walk(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2019-novel-Coronavirus-severe-adult-respiratory-dist_2020_International-Jour-p3-89%0.png"):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2019-novel-Coronavirus-severe-adult-respiratory-dist_2020_International-Jour-p3-89%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2019-novel-Coronavirus-severe-adult-respiratory-dist_2020_International-Jour-p3-89%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2019-novel-Coronavirus-severe-adult-respiratory-dist_2020_International-Jour-p3-91.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.01.24.919183-p27-132.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.01.24.919183-p27-133.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.01.24.919183-p27-134.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.01.24.919183-p27-135.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%10.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%11.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%12.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%13.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%14.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%15.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%4.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.02.10.20021584-p6-52%5.png
```


localhost:8888/nbconvert/html/Healthcare Diagnosis and Risk Prediction.ipynb?download=false 17/53

[illegible]

[illegible]

localhost:8888/nbconvert/html/Healthcare Diagnosis and Risk Prediction.ipynb?download=false 20/53

[illegible]

[illegible]

23/53

[illegible]

vid-19 CTs\Covid\2020.03.26.20041426-p11-122.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-123.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-124.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-125.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-126.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-127.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-128.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-130.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-131.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-132.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.26.20041426-p11-133.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.30.20047985-p9-56%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.30.20047985-p9-56%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.30.20047985-p9-56%2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\2020.03.30.20047985-p9-56%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%4.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%5.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%6.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\bmj.m606.full-p4-22%7.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journa-p2-21%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journa-p2-21%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journa-p2-21%10.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journa-p2-21%11.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journa-p2-21%2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journa-p2-21%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journa-p2-21%3.png

vid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journal-p2-21%4.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journal-p2-21%5.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journal-p2-21%6.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journal-p2-21%7.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journal-p2-21%8.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Comparison-of-different-samples-for-2019-novel-cor_2020_International-Journal-p2-21%9.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\ehaa254-p0-63%2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\ehaa254-p0-63%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\impact_of_covid19_infection_on_pregnancy_outcomes_and_the_risk_of_maternal_toneonatal_intrapartum_transmission_of_covid19_during_natural_birth-p8-41%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\impact_of_covid19_infection_on_pregnancy_outcomes_and_the_risk_of_maternal_toneonatal_intrapartum_transmission_of_covid19_during_natural_birth-p8-41%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\impact_of_covid19_infection_on_pregnancy_outcomes_and_the_risk_of_maternal_toneonatal_intrapartum_transmission_of_covid19_during_natural_birth-p8-41%5.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e24-p2-8.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e24-p3-16.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e24-p5-29.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e24-p5-31.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e25-p1-10.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e25-p1-12.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e25-p1-14.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e25-p1-6.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\kjr-21-e25-p1-8.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\med-9780198814924-chapter-17-p10-515.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Ni2020_Article_CombinationOfWesternMedicineAn-p3-10%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Ni2020_Article_CombinationOfWesternMedicineAn-p3-10%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Ni2020_Article_CombinationOfWesternMedicineAn-p4-13%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Ni2020_Article_CombinationOfWesternMedicineAn-p4-13%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620301549_0%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620301549_0%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620301549_1.png

C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620301549_2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620302117_1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620302117_2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%4.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%5.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%6.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%7.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\PIIS0140673620303603%8.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Recurrence-of-positive-SARS-CoV-2-RNA-in-C_2020_International-Journal-of-Inf-p1-21%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Recurrence-of-positive-SARS-CoV-2-RNA-in-C_2020_International-Journal-of-Inf-p1-21%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Talaromyces-marneffeii-infection-relapse-presenting-as-ost_2020_International-p1-12%0.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Talaromyces-marneffeii-infection-relapse-presenting-as-ost_2020_International-p1-12%1.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Talaromyces-marneffeii-infection-relapse-presenting-as-ost_2020_International-p1-12%2.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\Covid\Talaromyces-marneffeii-infection-relapse-presenting-as-ost_2020_International-p1-12%3.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\0.jpg
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\1%0.jpg
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\1%1.jpg
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\1%2.jpg
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\10%0.jpg
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\10%1.jpg
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\10%2.jpg
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\102.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\1029.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\103.png
C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\1030.png

28/53

[illegible]

[illegible]

31/53

32/53

33/53

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\996.png

C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk Prediction\Covid-19 CTs\NonCovid\997.png

```
In [52]: import os
data_dir =(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis and Risk
categories = ['Covid', 'NonCovid']
data = []
labels = []
for category in categories:
    path = os.path.join(data_dir, category)
    class_num = categories.index(category)
    for img in os.listdir(path):
        try:
            img_array = cv2.imread(os.path.join(path, img), cv2.IMREAD_GRAYSCALE)
            resized_array = cv2.resize(img_array, (224, 224))
            data.append(resized_array)
            labels.append(class_num)
        except Exception as e:
            pass
```

```
In [53]: print(f"Number of images loaded: {len(data)}")
print(f"Number of labels loaded: {len(labels)}")
```

Number of images loaded: 746
Number of labels loaded: 746

```
In [54]: data = np.array(data).reshape(-1, 224, 224, 1)
data = data / 255.0
labels = np.array(labels)
```

```
In [55]: x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, ran
```

```
In [56]: model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 1)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')
])
```

```
In [57]: model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['ac
```

```
In [58]: datagen = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
```



```

        fill_mode="nearest"
    )
    datagen.fit(x_train)

```

In [59]:

```
history = model.fit(datagen.flow(x_train, y_train, batch_size=32), epochs=20, valida
```

```

Epoch 1/20
19/19 ————— 21s 902ms/step - accuracy: 0.5104 - loss: 1.4690 - val_acc
uracy: 0.5333 - val_loss: 0.7181
Epoch 2/20
19/19 ————— 16s 789ms/step - accuracy: 0.5499 - loss: 0.7072 - val_acc
uracy: 0.5333 - val_loss: 0.6398
Epoch 3/20
19/19 ————— 18s 892ms/step - accuracy: 0.5568 - loss: 0.6767 - val_acc
uracy: 0.5600 - val_loss: 0.6663
Epoch 4/20
19/19 ————— 17s 818ms/step - accuracy: 0.5970 - loss: 0.6666 - val_acc
uracy: 0.6333 - val_loss: 0.6215
Epoch 5/20
19/19 ————— 17s 833ms/step - accuracy: 0.5397 - loss: 0.6776 - val_acc
uracy: 0.5600 - val_loss: 0.6648
Epoch 6/20
19/19 ————— 16s 801ms/step - accuracy: 0.5536 - loss: 0.6874 - val_acc
uracy: 0.5400 - val_loss: 0.6304
Epoch 7/20
19/19 ————— 17s 837ms/step - accuracy: 0.5638 - loss: 0.6567 - val_acc
uracy: 0.6733 - val_loss: 0.6201
Epoch 8/20
19/19 ————— 18s 905ms/step - accuracy: 0.6123 - loss: 0.6531 - val_acc
uracy: 0.5733 - val_loss: 0.6236
Epoch 9/20
19/19 ————— 17s 834ms/step - accuracy: 0.5330 - loss: 0.6759 - val_acc
uracy: 0.6400 - val_loss: 0.6188
Epoch 10/20
19/19 ————— 17s 832ms/step - accuracy: 0.5906 - loss: 0.6711 - val_acc
uracy: 0.6533 - val_loss: 0.6228
Epoch 11/20
19/19 ————— 17s 833ms/step - accuracy: 0.5970 - loss: 0.6570 - val_acc
uracy: 0.7200 - val_loss: 0.5972
Epoch 12/20
19/19 ————— 22s 1s/step - accuracy: 0.6388 - loss: 0.6482 - val_accura
cy: 0.6600 - val_loss: 0.6086
Epoch 13/20
19/19 ————— 26s 1s/step - accuracy: 0.6120 - loss: 0.6440 - val_accura
cy: 0.5667 - val_loss: 0.6524
Epoch 14/20
19/19 ————— 21s 1s/step - accuracy: 0.5899 - loss: 0.6700 - val_accura
cy: 0.7067 - val_loss: 0.5993
Epoch 15/20
19/19 ————— 19s 917ms/step - accuracy: 0.6421 - loss: 0.6498 - val_acc
uracy: 0.6667 - val_loss: 0.5882
Epoch 16/20
19/19 ————— 20s 908ms/step - accuracy: 0.6443 - loss: 0.6638 - val_acc
uracy: 0.6867 - val_loss: 0.6138
Epoch 17/20
19/19 ————— 19s 907ms/step - accuracy: 0.6143 - loss: 0.6385 - val_acc
uracy: 0.6933 - val_loss: 0.6129
Epoch 18/20
19/19 ————— 19s 922ms/step - accuracy: 0.5673 - loss: 0.6736 - val_acc
uracy: 0.6467 - val_loss: 0.5822
Epoch 19/20
19/19 ————— 19s 916ms/step - accuracy: 0.6111 - loss: 0.6349 - val_acc
uracy: 0.6467 - val_loss: 0.5758

```

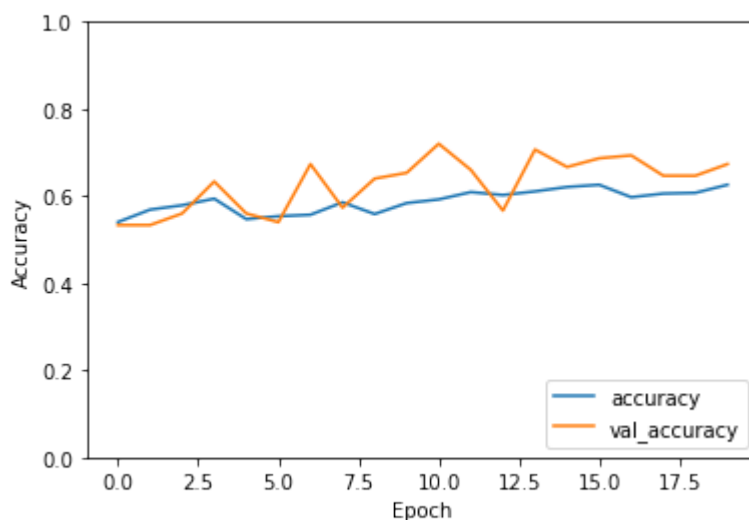
Epoch 20/20

19/19 ————— **19s** 884ms/step - accuracy: 0.6314 - loss: 0.6355 - val_accuracy: 0.6733 - val_loss: 0.5682

In [60]: `model.evaluate(x_test, y_test)`

5/5 ————— **1s** 238ms/step - accuracy: 0.6780 - loss: 0.5613
 Out[60]: [0.5682364702224731, 0.6733333468437195]

In [61]: `plt.plot(history.history['accuracy'], label='accuracy')
 plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
 plt.xlabel('Epoch')
 plt.ylabel('Accuracy')
 plt.ylim([0, 1])
 plt.legend(loc='lower right')
 plt.show()`



In [62]: `model.save('covid19_model.keras')`

In [64]: `from tensorflow.keras.models import load_model
 model = load_model('covid19_model.keras')

 # Predict on a new image
 new_image = cv2.imread(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagno
 new_image = cv2.resize(new_image, (224, 224))
 new_image = np.array(new_image).reshape(-1, 224, 224, 1)
 new_image = new_image / 255.0

 prediction = model.predict(new_image)
 print("Prediction:", categories[np.argmax(prediction)])`

1/1 ————— **0s** 238ms/step
 Prediction: NonCovid

In [65]: `from tensorflow.keras.models import load_model
 model = load_model('covid19_model.keras')

 # Predict on a new image
 new_image = cv2.imread(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagno
 new_image = cv2.resize(new_image, (224, 224))
 new_image = np.array(new_image).reshape(-1, 224, 224, 1)
 new_image = new_image / 255.0`

```
prediction = model.predict(new_image)
print("Prediction:", categories[np.argmax(prediction)])
```

1/1  0s 142ms/step

Prediction: Covid

Drug Discovery

In [66]: `drug=pd.read_csv(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnosis an
drug`

Out[66]:

	Id	Gen	Smile	Source	Score	P_value	Conne
0	1	0	COc1cccc(NC(=O)Cc2ccc(NC(=O)N3CCCC3)cc2)c1	generated	99.9	0.003320	-0.
1	2	0	C=CCNC(=O)CNc1cccc(C(=O)N(C)CCc2ccccc2)c1	generated	99.9	0.003384	-0.
2	3	0	CC(=O)Nc1ccc(S(=O)(=O)Nc2ccc(C)c(C)c2)cc1	training	99.9	0.003397	-0.
3	4	0	CCOC(=O)C1=C(C(=O)OCC)C(c2cccc(Cl)c2)NC(=O)N1	generated	99.9	0.003427	-0.
4	5	0	NC(=O)c1ccc(NC(=O)C(CC(=O)O)NC(=O)c2cc(- c3cccc...	generated	99.9	0.003468	-0.
...
1172	1173	0	CCCC1(CCc2ccccc2)CC(O)=C(C(CC)c2cccc(NS(=O) (=O...	manual	99.9	1.000000	0.
1173	1174	0	O=C1Nc2ccc(Cl)cc2C(C#CC2CC2)(C(F)(F)F)O1	manual	99.9	1.000000	0.
1174	1175	0	CC(C) (C)NC(=O)C1CN(Cc2ccccc2)CCN1CC(O)CC(Cc1cc...	manual	99.9	1.000000	0.
1175	1176	0	CCOP(=O) (COc1ccc(CC(NC(=O)OC2COC3OCCC23)C(O)CN...	manual	99.9	1.000000	0.
1176	1177	0	COC(=O)NC(C(=O)NCCCC(CO)N(CC(C)C)S(=O) (=O)c1c...	manual	99.9	1.000000	0.

1177 rows × 8 columns



In [67]: `drug['P_value'].unique()`

Out[67]: `array([0.00331968, 0.00338356, 0.00339726, 0.00342658, 0.0034682 ,
0.00353087, 0.00355484, 0.00357524, 0.00358432, 0.00363285,
0.0037343 , 0.00389023, 0.00396885, 0.0042186 , 0.00450414,
0.00486791, 0.00550357, 0.00628477, 0.00643704, 0.00645932,
0.00664478, 0.00697121, 0.00697142, 0.00757783, 0.00895931,
0.00914021, 0.00915257, 0.00949567, 0.00980125, 0.01052681,
0.01165854, 0.01199205, 0.01328552, 0.01357921, 0.01416115,
0.01450921, 0.01463088, 0.01463118, 0.01520905, 0.01579844,
0.01631903, 0.01670527, 0.01700551, 0.01744755, 0.01788815,
0.01826373, 0.0196488 , 0.01981558, 0.02005149, 0.02089379,
0.02101087, 0.02143682, 0.02161217, 0.0219501 , 0.02234632,
0.02278351, 0.02366028, 0.023799 , 0.02386443, 0.02525801,
0.02753619, 0.02795078, 0.02799259, 0.02833159, 0.02851516,
0.02862323, 0.0286239 , 0.0289735 , 0.03181352, 0.03252018,
0.03329107, 0.03354188, 0.03489295, 0.03526253, 0.03633253,
0.03719375, 0.03806932, 0.0385287 , 0.03897426, 0.03898927,`

0.03990402, 0.04046838, 0.04186649, 0.0418879 , 0.04202482,
0.04278029, 0.04308126, 0.04330748, 0.04456587, 0.04456773,
0.04492331, 0.04518693, 0.04588357, 0.04598648, 0.04660964,
0.04676691, 0.04695373, 0.04730705, 0.04757408, 0.04768286,
0.04870295, 0.05051713, 0.05053752, 0.05088461, 0.05182202,
0.05275996, 0.05312042, 0.05427295, 0.05438078, 0.05514716,
0.0552889 , 0.05568718, 0.05570817, 0.05605124, 0.05631784,
0.05651632, 0.05871457, 0.06042298, 0.06051576, 0.06052005,
0.06195518, 0.06382694, 0.06484602, 0.06511162, 0.06520212,
0.06801367, 0.06811696, 0.06867639, 0.06892979, 0.06939979,
0.06946658, 0.06976797, 0.06985624, 0.0704378 , 0.07101772,
0.0715195 , 0.07235723, 0.07247133, 0.07256171, 0.07294612,
0.07348202, 0.07354344, 0.07386633, 0.07582825, 0.07832784,
0.07905461, 0.07907366, 0.07934654, 0.07973238, 0.07980567,
0.07985288, 0.08081229, 0.08163699, 0.08181505, 0.08269558,
0.08323564, 0.08490506, 0.08534807, 0.08605934, 0.08635457,
0.08788697, 0.08806246, 0.08879437, 0.08895638, 0.09034253,
0.09052705, 0.09104235, 0.09104315, 0.09104351, 0.09164863,
0.09238933, 0.09306361, 0.09512018, 0.09555039, 0.09599071,
0.09605064, 0.09713684, 0.0985899 , 0.09952477, 0.10055522,
0.10233047, 0.10413599, 0.10451931, 0.10484266, 0.1062647 ,
0.10697024, 0.10719087, 0.10858597, 0.11185228, 0.11190014,
0.11269503, 0.11280344, 0.11365458, 0.11459782, 0.11519795,
0.11654348, 0.11750316, 0.11889455, 0.11979169, 0.12018904,
0.12029071, 0.12276693, 0.12353802, 0.12390583, 0.12528238,
0.12540111, 0.12793708, 0.12933567, 0.12954606, 0.12988079,
0.13063647, 0.13096454, 0.13260898, 0.13355352, 0.1357788 ,
0.1361732 , 0.13671895, 0.13886054, 0.13927272, 0.14051 ,
0.14070866, 0.14088994, 0.14303162, 0.14806286, 0.14975012,
0.15256049, 0.15369048, 0.153924 , 0.15497247, 0.15562023,
0.15601202, 0.159757 , 0.16224638, 0.16280553, 0.16292021,
0.16302445, 0.16377314, 0.16875503, 0.1704928 , 0.17167054,
0.17234037, 0.17408346, 0.17500657, 0.17746683, 0.17785168,
0.17914667, 0.18192299, 0.18262087, 0.18344105, 0.18454918,
0.18738514, 0.18805847, 0.18832182, 0.19356944, 0.19369538,
0.19398345, 0.19751145, 0.19822424, 0.2005027 , 0.20072904,
0.20129993, 0.2031834 , 0.2045194 , 0.20949127, 0.21140648,
0.2126268 , 0.21262762, 0.21613169, 0.21780821, 0.21813464,
0.219131 , 0.21973215, 0.22184002, 0.22306153, 0.2255502 ,
0.22735973, 0.22909195, 0.22978112, 0.23050537, 0.23165731,
0.23432149, 0.23781397, 0.23836461, 0.23956356, 0.24059466,
0.2416225 , 0.24277041, 0.24295499, 0.24399913, 0.24697031,
0.25342106, 0.25381506, 0.25786067, 0.25936012, 0.2618198 ,
0.2629654 , 0.26451257, 0.26835093, 0.27131117, 0.27422719,
0.27746106, 0.28264132, 0.28696908, 0.28860334, 0.28989767,
0.29014537, 0.29041303, 0.29251472, 0.29309081, 0.29326431,
0.29479569, 0.2954055 , 0.29569829, 0.29959839, 0.30115981,
0.30311155, 0.30610146, 0.30647185, 0.31038972, 0.31091985,
0.31130012, 0.31566784, 0.31799599, 0.31969131, 0.32697298,
0.33026311, 0.33493444, 0.33584273, 0.33698531, 0.33813792,
0.33906063, 0.3397717 , 0.34077915, 0.3415142 , 0.34348328,
0.34451839, 0.34566832, 0.35022996, 0.35345171, 0.35524473,
0.35578981, 0.35606765, 0.3565432 , 0.35925243, 0.35978151,
0.36077026, 0.3611143 , 0.3645551 , 0.36466251, 0.36488603,
0.36599735, 0.36632923, 0.36732451, 0.37056154, 0.37266574,
0.37691524, 0.3793924 , 0.382819 , 0.38502115, 0.38670005,
0.38963828, 0.3904334 , 0.39046557, 0.39338427, 0.39496421,
0.39533117, 0.39597373, 0.39611395, 0.39717108, 0.40041048,
0.40283063, 0.40335082, 0.40725778, 0.40862107, 0.41547754,
0.41691513, 0.41779437, 0.41875114, 0.42050279, 0.42392585,
0.42457283, 0.43150115, 0.43405619, 0.4345 , 0.43491885,
0.43743994, 0.44159002, 0.44313879, 0.44344144, 0.44504064,
0.44840355, 0.45309878, 0.4532101 , 0.45540034, 0.45643784,
0.45777869, 0.46177079, 0.46387273, 0.46650919, 0.46773491,

```
0.47100626, 0.47891295, 0.48408713, 0.4890688 , 0.49065342,
0.49091957, 0.49403979, 0.49502733, 0.50120567, 0.51428287,
0.51527591, 0.51545886, 0.51561018, 0.51607102, 0.51700611,
0.51828149, 0.53071362, 0.53259767, 0.53337217, 0.53477984,
0.53610198, 0.53855403, 0.54437613, 0.54716412, 0.54831384,
0.54929137, 0.55035048, 0.55075521, 0.55199519, 0.55631202,
0.55880135, 0.55986261, 0.56173782, 0.56272893, 0.56286225,
0.56591826, 0.56785224, 0.56978596, 0.57093103, 0.57407658,
0.57999508, 0.58535212, 0.58550422, 0.58626813, 0.59541269,
0.59746188, 0.59822391, 0.60353854, 0.60386134, 0.60742148,
0.60751863, 0.60976589, 0.61140965, 0.62799904, 0.62815295,
0.62953156, 0.63818857, 0.64599034, 0.64846181, 0.66346141,
0.66994783, 0.67341921, 0.67359182, 0.67436709, 0.67515299,
0.67690288, 0.68476138, 0.68754834, 0.69626519, 0.70578261,
0.71110034, 0.71729496, 0.72085179, 0.72629987, 0.72695961,
0.72907016, 0.72952725, 0.73260065, 0.73553037, 0.74223549,
0.74764648, 0.74881536, 0.75298128, 0.75976866, 0.75990987,
0.76268508, 0.76287822, 0.766514 , 0.76814667, 0.77541254,
0.77667636, 0.78000203, 0.78502197, 0.78651408, 0.78684912,
0.79494148, 0.79848434, 0.80228855, 0.80485427, 0.8056745 ,
0.80638183, 0.80671752, 0.80794889, 0.80882712, 0.81621584,
0.81930391, 0.82023755, 0.82785384, 0.8282647 , 0.83005382,
0.83728053, 0.83813994, 0.83853873, 0.84430938, 0.84688197,
0.84764454, 0.84921788, 0.85083233, 0.85232688, 0.85643685,
0.86003014, 0.86004176, 0.8608426 , 0.86191448, 0.86540266,
0.87008488, 0.8782218 , 0.88538814, 0.88558038, 0.89178938,
0.89319651, 0.89734209, 0.89865245, 0.90071128 , 0.90102661,
0.90178276, 0.90324523, 0.90827515, 0.90889474, 0.90925129,
0.91212533, 0.91380576, 0.91990037, 0.92793323, 0.92866678,
0.93086134, 0.93188156, 0.9376778 , 0.94442947, 0.94475727,
0.94673358, 0.95926652, 0.96551891, 0.96554576, 0.96982546,
0.97057751, 0.97200691, 0.97305079, 0.97315653, 0.97444229,
0.97716526, 0.98104994, 0.98491947, 0.99460722, 0.99893939,
1.      ])
```

```
In [68]: drug['Source'].unique()
```

```
Out[68]: array(['generated', 'training', 'hiv', 'manual'], dtype=object)
```

```
In [69]: drug_sorted= drug.sort_values(by = 'P_value', ascending = True)
drug_sorted
```

```
Out[69]:
```

	Id	Gen	Smile	Source	Score	P_value	Cc
0	1	0	COc1cccc(NC(=O)Cc2ccc(NC(=O)N3CCCC3)cc2)c1	generated	99.9	0.003320	
1	2	0	C=CCNC(=O)CNC1cccc(C(=O)N(C)CCc2ccccc2)c1	generated	99.9	0.003384	
2	3	0	CC(=O)Nc1ccc(S(=O)(=O)Nc2ccc(C)c(C)c2)cc1	training	99.9	0.003397	
3	4	0	CCOC(=O)C1=C(C(=O)OCC)C(c2cccc(Cl)c2)NC(=O)N1	generated	99.9	0.003427	
4	5	0	NC(=O)c1ccc(NC(=O)C(CC(=O)O)NC(=O)c2cc(-c3cccc...	generated	99.9	0.003468	
...
772	773	0	COc1ccc2c(c1)OC(CNC(=O)NCc1ccco1)CC2	generated	99.9	1.000000	
773	774	0	COC(=O)C1COC(=O)N(c2ccc(NS(C)(=O)=O)cc2)C1	generated	99.9	1.000000	
774	775	0	O=C(COc1ccc2cc(OCCCN3CCOCC3)ccc2c1)NCC(c1ccccc...	generated	99.9	1.000000	
767	768	0	Cc1cccc(C)c1N(C)S(=O)(=O)c1cnn(C)c1	generated	99.9	1.000000	

	Id	Gen	Smile	Source	Score	P_value	Cc
1176	1177	0	COC(=O)NC(C(=O)NCCCC(CO)N(CC(C)C)S(=O)(=O)c1c...	manual	99.9	1.000000	

1177 rows × 8 columns

In [70]:

```
drug_sorted = drug_sorted[['Smile', 'P_value']]
drug_sorted
```

Out[70]:

	Smile	P_value
0	COc1cccc(NC(=O)Cc2ccc(NC(=O)N3CCCC3)cc2)c1	0.003320
1	C=CCNC(=O)CNc1cccc(C(=O)N(C)CCc2ccccc2)c1	0.003384
2	CC(=O)Nc1ccc(S(=O)(=O)Nc2ccc(C)c(C)c2)cc1	0.003397
3	CCOC(=O)C1=C(C(=O)OCC)C(c2cccc(Cl)c2)NC(=O)N1	0.003427
4	NC(=O)c1ccc(NC(=O)C(CC(=O)O)NC(=O)c2cc(-c3cccc...	0.003468
...
772	COc1ccc2c(c1)OC(CNC(=O)NCc1ccco1)CC2	1.000000
773	COC(=O)C1COC(=O)N(c2ccc(NS(C)(=O)=O)cc2)C1	1.000000
774	O=C(COc1ccc2cc(OCCCN3CCOCC3)ccc2c1)NCC(c1cccc...	1.000000
767	Cc1cccc(C)c1N(C)S(=O)(=O)c1cnn(C)c1	1.000000
1176	COC(=O)NC(C(=O)NCCCC(CO)N(CC(C)C)S(=O)(=O)c1c...	1.000000

1177 rows × 2 columns

In [71]:

```
drug_sorted.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1177 entries, 0 to 1176
Data columns (total 2 columns):
#   Column   Non-Null Count  Dtype
---  -
0    Smile    1177 non-null   object
1    P_value   1177 non-null   float64
dtypes: float64(1), object(1)
memory usage: 27.6+ KB
```

In [72]:

```
drug['Source']=drug['Source'].map({'generated':1,'training':0,'hiv':2,'manual':3})
```

In [73]:

```
drug1=drug.drop(['Id', 'Gen', 'Name'],axis=1)
drug1
```

Out[73]:

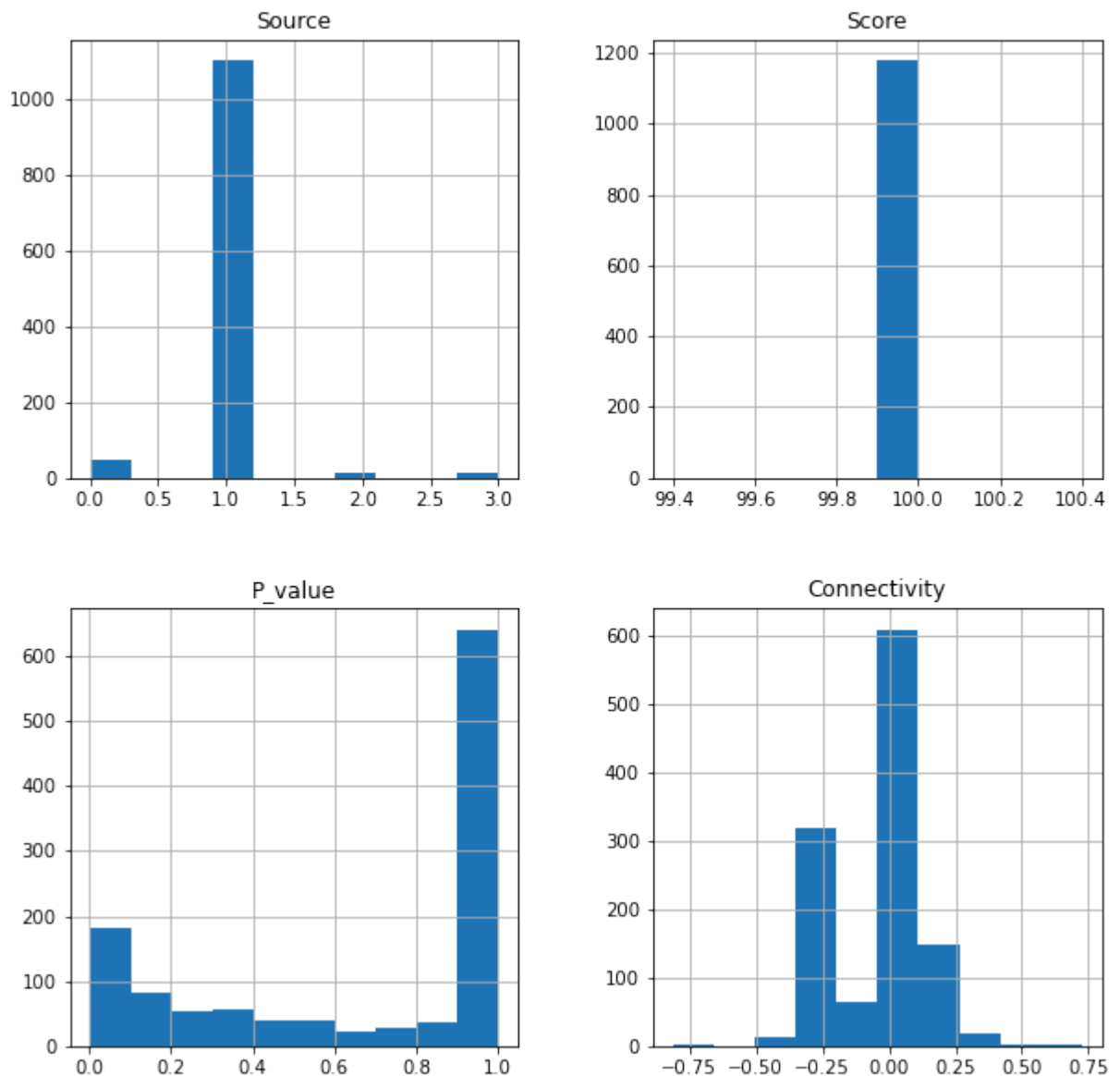
	Smile	Source	Score	P_value	Connectivity
0	COc1cccc(NC(=O)Cc2ccc(NC(=O)N3CCCC3)cc2)c1	1	99.9	0.003320	-0.376625
1	C=CCNC(=O)CNc1cccc(C(=O)N(C)CCc2ccccc2)c1	1	99.9	0.003384	-0.269090
2	CC(=O)Nc1ccc(S(=O)(=O)Nc2ccc(C)c(C)c2)cc1	0	99.9	0.003397	-0.318895

	Smile	Source	Score	P_value	Connectivity
3	<chem>CCOC(=O)C1=C(C(=O)OCC)C(c2cccc(Cl)c2)NC(=O)N1</chem>	1	99.9	0.003427	-0.329905
4	<chem>NC(=O)c1ccc(NC(=O)C(CC(=O)O)NC(=O)c2cc(-c3cccc...</chem>	1	99.9	0.003468	-0.288555
...
1172	<chem>CCCC1(CCc2ccccc2)CC(O)=C(C(CC)c2cccc(NS(=O))(=O...</chem>	3	99.9	1.000000	0.000000
1173	<chem>O=C1Nc2ccc(Cl)cc2C(C#CC2CC2)(C(F)(F)F)O1</chem>	3	99.9	1.000000	0.000000
1174	<chem>CC(C)(C)NC(=O)C1CN(Cc2cccn2)CCN1CC(O)CC(Cc1cc...</chem>	3	99.9	1.000000	0.000000
1175	<chem>CCOP(=O)(COc1ccc(CC(NC(=O)OC2COC3OCCC23)C(O)CN...</chem>	3	99.9	1.000000	0.000000
1176	<chem>COC(=O)NC(C(=O)NCCCC(CO)N(CC(C)C)S(=O)(=O)c1c...</chem>	3	99.9	1.000000	0.000000

1177 rows × 5 columns

```
In [74]: drug1.hist(figsize=(10,10))
```

```
Out[74]: array([[<AxesSubplot:title={'center':'Source'}>,  
      <AxesSubplot:title={'center':'Score'}>],  
      [<AxesSubplot:title={'center':'P_value'}>,  
      <AxesSubplot:title={'center':'Connectivity'}>]], dtype=object)
```



In [75]: `drug1.isna().sum()`

Out[75]:

Smile	0
Source	0
Score	0
P_value	0
Connectivity	0
dtype:	int64

In [76]: `x=drug1.drop(['Smile', 'P_value'],axis=1)`
`y=drug1['P_value']`

In [77]: `from sklearn.model_selection import train_test_split`
`x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)`

In [78]: `from sklearn.model_selection import GridSearchCV`
`from sklearn.ensemble import RandomForestRegressor`
`reg=RandomForestRegressor()`
`n_estimators=[25,50,75,100,125,150,175,200]`
`criterion=['squared_error']`
`max_depth=[3,5,10]`
`parameters={'n_estimators': n_estimators, 'criterion': criterion, 'max_depth': max_depth}`
`rfc_reg = GridSearchCV(reg, parameters)`
`rfc_reg.fit(x_train,y_train)`

Out[78]:

```

GridSearchCV
  estimator: RandomForestRegressor
    RandomForestRegressor

```

In [79]: `rfc_reg.best_params_`

Out[79]: `{'criterion': 'squared_error', 'max_depth': 3, 'n_estimators': 125}`

In [80]: `reg=RandomForestRegressor(n_estimators=125,criterion='squared_error',max_depth=3)`

In [81]: `reg.fit(x_train,y_train)`

Out[81]:

```

RandomForestRegressor
RandomForestRegressor(max_depth=3, n_estimators=125)

```

In [82]: `y_pred4=reg.predict(x_test)`
`y_pred4`

Out[82]:

```

array([0.11394094, 0.98344282, 0.08022616, 0.98344282, 0.98344282,
        0.98344282, 0.98344282, 0.98344282, 0.08460217, 0.67503556,
        0.97781285, 0.18298636, 0.98344282, 0.98344282, 0.08376053,
        0.23771604, 0.97433455, 0.98344282, 0.23901554, 0.98344282,
        0.11394094, 0.36115587, 0.36115587, 0.98344282, 0.08022616,
        0.46363525, 0.23771604, 0.23901554, 0.24020433, 0.39011664,

```


0.08376053, 0.98344282, 0.98344282, 0.08022616, 0.97433455,
0.98344282, 0.98344282, 0.67503556, 0.17284582, 0.98344282,
0.98344282, 0.98344282, 0.24020433, 0.98344282, 0.98344282,
0.35677375, 0.98344282, 0.98344282, 0.98344282, 0.55170252,
0.98344282, 0.16092847, 0.18298636, 0.23771604, 0.98344282,
0.98344282, 0.98344282, 0.98344282, 0.98344282, 0.98344282,
0.98344282, 0.98344282, 0.39011664, 0.98344282, 0.98344282,
0.98344282, 0.98344282, 0.98155505, 0.98344282, 0.47006357,
0.98155505, 0.98344282, 0.98344282, 0.08376053, 0.98344282,
0.98344282, 0.30806659, 0.25676267, 0.98344282, 0.98344282,
0.23901554, 0.46363525, 0.14565726, 0.30035269, 0.23771604,
0.2487313 , 0.21265106, 0.98344282, 0.98344282, 0.98344282,
0.47006357, 0.98344282, 0.98344282, 0.98344282, 0.98344282,
0.39011664, 0.67775977, 0.23771604, 0.98344282, 0.98344282,
0.98344282, 0.08022616, 0.98344282, 0.98344282, 0.98344282,
0.08376053, 0.98344282, 0.08376053, 0.98344282, 0.98344282,
0.29844605, 0.08022616, 0.98344282, 0.35677375, 0.98344282,
0.46805783, 0.67503556, 0.39011664, 0.98344282, 0.24758739,
0.98344282, 0.98344282, 0.14565726, 0.23771604, 0.48621922,
0.16092847, 0.23771604, 0.11394094, 0.98344282, 0.98155505,
0.14565726, 0.98344282, 0.98344282, 0.67503556, 0.23771604,
0.98344282, 0.98344282, 0.97433455, 0.14565726, 0.11394094,
0.08600972, 0.98344282, 0.39011664, 0.23771604, 0.98344282,
0.08022616, 0.98344282, 0.23901554, 0.98344282, 0.98344282,
0.24020433, 0.16092847, 0.98344282, 0.97433455, 0.14565726,
0.98344282, 0.08376053, 0.98344282, 0.98344282, 0.98344282,
0.98344282, 0.98344282, 0.98344282, 0.14814182, 0.91341502,
0.98344282, 0.98344282, 0.98344282, 0.98344282, 0.08022616,
0.67503556, 0.98344282, 0.98344282, 0.12057016, 0.98344282,
0.98344282, 0.98344282, 0.98344282, 0.98344282, 0.98344282,
0.84265949, 0.98344282, 0.98344282, 0.98344282, 0.66742357,
0.98344282, 0.24758739, 0.84265949, 0.98344282, 0.98344282,
0.96207513, 0.98344282, 0.23901554, 0.98344282, 0.98344282,
0.98344282, 0.98344282, 0.67503556, 0.98344282, 0.70058939,
0.98344282, 0.08022616, 0.98344282, 0.98344282, 0.98344282,
0.17284582, 0.98344282, 0.98344282, 0.39011664, 0.08376053,
0.98155505, 0.98344282, 0.98344282, 0.98344282, 0.67503556,
0.08376053, 0.98344282, 0.14814182, 0.14814182, 0.98344282,
0.98344282, 0.14565726, 0.23771604, 0.93843854, 0.67503556,
0.14814182, 0.08376053, 0.37155545, 0.98344282, 0.18298636,
0.1445022 , 0.98344282, 0.08022616, 0.98344282, 0.67271978,
0.98344282, 0.98344282, 0.23901554, 0.62053942, 0.97433455,
0.84265949, 0.09272078, 0.98344282, 0.46363525, 0.98344282,
0.23901554, 0.98344282, 0.23771604, 0.98344282, 0.23901554,
0.98344282, 0.38628721, 0.23771604, 0.95407736, 0.23901554,
0.98344282, 0.55170252, 0.98344282, 0.08460217, 0.98344282,
0.98344282, 0.18298636, 0.08022616, 0.23901554, 0.25608627,
0.47245739, 0.23771604, 0.98344282, 0.98344282, 0.08022616,
0.62053942, 0.98344282, 0.98344282, 0.98344282, 0.37155545,
0.98344282, 0.98344282, 0.98344282, 0.11035792, 0.98344282,
0.98344282, 0.40155541, 0.98344282, 0.98344282, 0.98344282,
0.98344282, 0.14565726, 0.98344282, 0.98344282, 0.98344282,
0.37155545, 0.23901554, 0.08376053, 0.98344282, 0.08600972,
0.08022616, 0.98344282, 0.98344282, 0.98344282, 0.98344282,
0.98344282, 0.98344282, 0.98344282, 0.67503556, 0.98344282,
0.98344282, 0.71355391, 0.98344282, 0.97433455, 0.98344282,
0.27886675, 0.98344282, 0.08022616, 0.98155505, 0.08022616,
0.11421769, 0.18298636, 0.98344282, 0.24020433, 0.98344282,
0.18298636, 0.96560003, 0.08022616, 0.98344282, 0.46363525,
0.39153281, 0.98344282, 0.98344282, 0.11035792, 0.98344282,
0.98344282, 0.98344282, 0.42106301, 0.36642945, 0.36103774,
0.08022616, 0.46363525, 0.23901554, 0.24020433, 0.23901554,
0.08376053, 0.98344282, 0.11421769, 0.14565726, 0.98344282,

0.98344282, 0.98344282, 0.14565726, 0.98344282, 0.98344282,
0.98344282, 0.98344282, 0.98344282, 0.98344282])

```
In [83]: from sklearn.metrics import r2_score
r2_score(y_test,y_pred4)
```

Out[83]: 0.9382886492776896

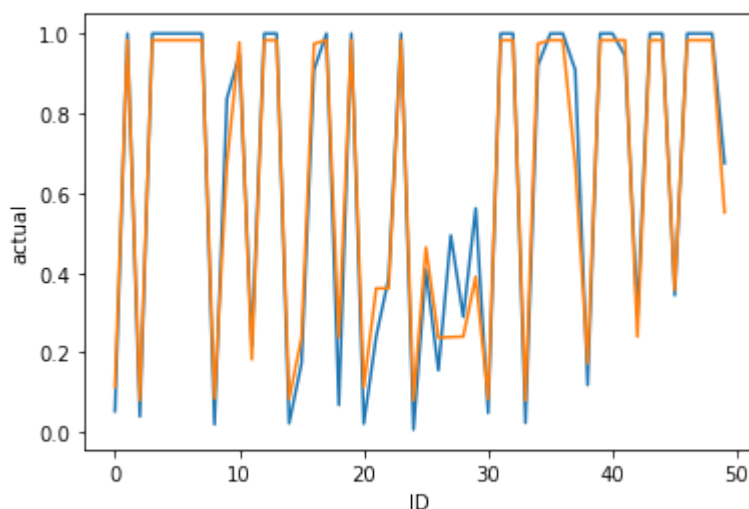
```
In [84]: results=pd.DataFrame(columns=['actual','predicted'])
results['actual']=y_test
results['predicted']=y_pred4
results=results.reset_index()
results['ID']=results.index
results.head(5)
```

Out[84]:

	index	actual	predicted	ID
0	107	0.053120	0.113941	0
1	774	1.000000	0.983443	1
2	81	0.039904	0.080226	2
3	787	1.000000	0.983443	3
4	665	1.000000	0.983443	4

```
In [85]: import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='actual',data=results.head(50))
sns.lineplot(x='ID',y='predicted',data=results.head(50))
plt.plot()
```

Out[85]: []



Personalized Treatment Recommendations

```
In [86]: import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import warnings
import os
```

```
import importlib
warnings.filterwarnings('ignore')
sns.set_theme(color_codes=True)
```

```
In [87]: def check_and_install_library(library_name):
        try:
            importlib.import_module(library_name)
            print(f"{library_name} is already installed.")
        except ImportError:
            print(f"{library_name} is not installed. Installing...")
            try:
                import pip
                pip.main(['install', library_name])
            except:
                print("Error: Failed to install the library. Please install it manually.")
```

```
In [88]: !pip show scikit-surprise
```

```
Name: scikit-surprise
Version: 1.1.3
Summary: An easy-to-use library for recommender systems.
Home-page: https://surpriselib.com
Author: Nicolas Hug
Author-email: contact@nicolas-hug.com
License: GPLv3+
Location: c:\users\reshma_koduri\anaconda3\lib\site-packages
Requires: joblib, numpy, scipy
Required-by:
```

```
In [89]: import sys
        print(sys.path)
        import sys
        sys.path.append('/path/to/directory')

['C:\\Users\\reshma_koduri', 'C:\\Users\\reshma_koduri\\anaconda3\\python39.zip',
'C:\\Users\\reshma_koduri\\anaconda3\\DLLs', 'C:\\Users\\reshma_koduri\\anaconda3\\li
b', 'C:\\Users\\reshma_koduri\\anaconda3', '', 'C:\\Users\\reshma_koduri\\anaconda3
\\lib\\site-packages', 'C:\\Users\\reshma_koduri\\anaconda3\\lib\\site-packages\\lock
et-0.2.1-py3.9.egg', 'C:\\Users\\reshma_koduri\\anaconda3\\lib\\site-packages\\win3
2', 'C:\\Users\\reshma_koduri\\anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\Users
\\reshma_koduri\\anaconda3\\lib\\site-packages\\Pythonwin', 'C:\\Users\\reshma_koduri
\\anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\reshma_koduri\\.ip
ython']
```

```
In [90]: treatment=pd.read_csv(r"C:\Users\reshma_koduri\OneDrive\Documents\Healthcare Diagnos
treatment
```

```
Out[90]:
```

	User_id	Age	Sex	Source	Haemoglobins	Erythrocyte	Leucocyte	Haematocrit	Treatment
0	1	68	F	out	11.8	4.65	6.3	35.1	Oxygen Therapy
1	2	70	F	out	14.8	5.39	12.7	43.5	Convalescent Plasma
2	3	64	F	out	11.3	4.74	13.2	33.5	COVID-19 Vaccine
3	4	73	F	out	13.7	4.98	10.5	39.1	Tocilizumab
4	5	66	M	out	9.9	4.23	22.1	30.9	Dexamethasone

	User_id	Age	Sex	Source	Haemoglobins	Erythrocyte	Leucocyte	Haematocrit	Treatment
...	
1184	1185	90	F	out	12.5	4.41	10.9	37.1	Remdesivir
1185	1186	90	F	out	10.6	4.44	9.4	33.7	Baricitinib
1186	1187	90	F	in	11.2	4.06	8.4	35.4	Anticoagulants
1187	1188	25	M	in	13.8	4.88	3.5	40.0	Molnupiravir
1188	1189	31	M	in	13.8	4.86	2.3	39.4	Casirivimab and imdevimab

1189 rows × 11 columns

```
In [91]: treatment.groupby(['Condition']).count()
```

```
Out[91]:
```

	User_id	Age	Sex	Source	Haemoglobins	Erythrocyte	Leucocyte	Haematocrit	Treatment
Condition									
improved	414	414	414	414	414	414	414	414	414
stable	418	418	418	418	418	418	418	418	418
worse	357	357	357	357	357	357	357	357	357

```
In [92]: treatment['Condition'].unique()
```

```
Out[92]: array(['worse', 'improved', 'stable'], dtype=object)
```

```
In [93]: treatment['Source'].unique()
```

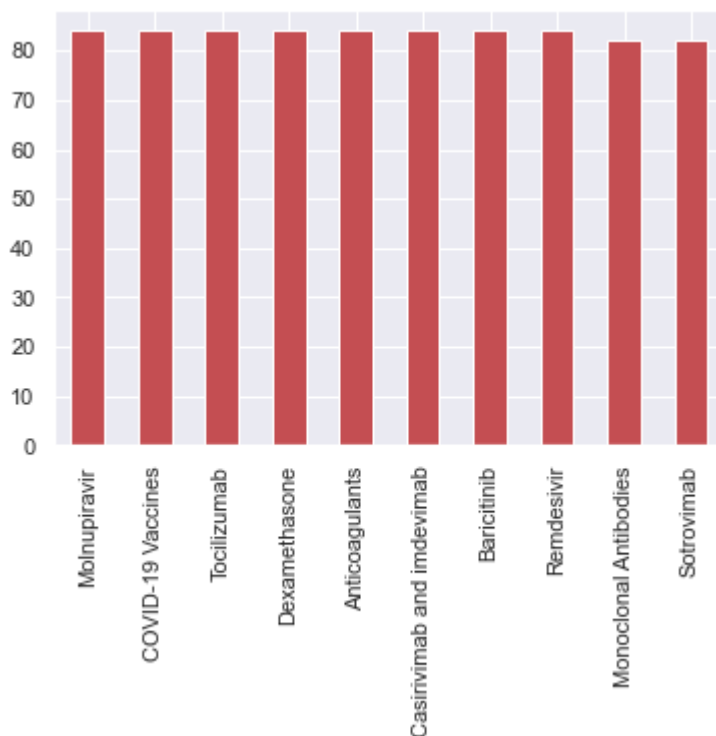
```
Out[93]: array(['out', 'in'], dtype=object)
```

```
In [94]: treatment['Treatment'].unique()
```

```
Out[94]: array(['Oxygen Therapy', 'Convalescent Plasma', 'COVID-19 Vaccines',
        'Tocilizumab', 'Dexamethasone',
        'Non-invasive Ventilation (CPAP/BiPAP)',
        'Casirivimab and imdevimab', 'Baricitinib', 'Remdesivir',
        'Monoclonal Antibodies', 'Etesevimab', 'Mechanical Ventilation',
        'Molnupiravir', 'Bamlanivimab', 'Sotrovimab', 'Anticoagulants',
        'Methylprednisolone', 'Heparin'], dtype=object)
```

```
In [95]: import seaborn as sns
import matplotlib.pyplot as plt
treatment['Treatment'].value_counts().head(10).plot(kind='bar', color='r')
```

```
Out[95]: <AxesSubplot:>
```



```
In [96]: treatment['Sex']=treatment['Sex'].map({'F':1,'M':0})
treatment['Condition']=treatment['Condition'].map({'worse':1,'stable':2,'improved':3})
treatment['Source']=treatment['Source'].map({'in':1,'out':0})
```

```
In [97]: from surprise import Dataset, Reader
from surprise import KNNBasic, accuracy, SVD
from surprise.model_selection import train_test_split
```

```
In [98]: reader = Reader(rating_scale=(1,3))
surprise_data = Dataset.load_from_df(treatment[['User_id', 'Treatment', 'Ratings']],
```

```
In [99]: trainset, testset = train_test_split(surprise_data, test_size=0.3, random_state=42)
```

```
In [100... model = SVD()
model.fit(trainset)
```

```
Out[100... <surprise.prediction_algorithms.matrix_factorization.SVD at 0x1ad91ee5df0>
```

```
In [101... predictions = model.test(testset)
```

```
In [102... from surprise.accuracy import rmse
```

```
In [103... accuracy = rmse(predictions)
print(f'RMSE: {accuracy}')
```

```
RMSE: 0.1045
```

```
RMSE: 0.10451349669382987
```