In [76]:

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
covid=pd.read_csv('Covid Dataset.csv')
```

In [77]:

```
covid
```

Out[77]:

| | Breathing Problem | Fever | Dry Cough | Sore throat | Running Nose | Asthma | Chronic Lung Disease | Headache | Heart Disease | Diabet |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Yes | Yes | Yes | Yes | Yes | No | No | No | No | Y |
| **1** | Yes | Yes | Yes | Yes | No | Yes | Yes | Yes | No | N |
| **2** | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No | Y |
| **3** | Yes | Yes | Yes | No | No | Yes | No | No | Yes | Y |
| **4** | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes | Yes | Y |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **5429** | Yes | Yes | No | Yes | Yes | Yes | Yes | No | No | N |
| **5430** | Yes | Yes | Yes | No | Yes | Yes | No | Yes | No | Y |
| **5431** | Yes | Yes | Yes | No | No | No | No | No | Yes | N |
| **5432** | Yes | Yes | Yes | No | Yes | No | No | Yes | Yes | N |
| **5433** | Yes | Yes | Yes | No | Yes | Yes | No | Yes | No | Y |

5434 rows × 21 columns

In [78]:

```
covid.isnull().sum()
```

Out[78]:

```
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
dtype: int64
```

In [79]:

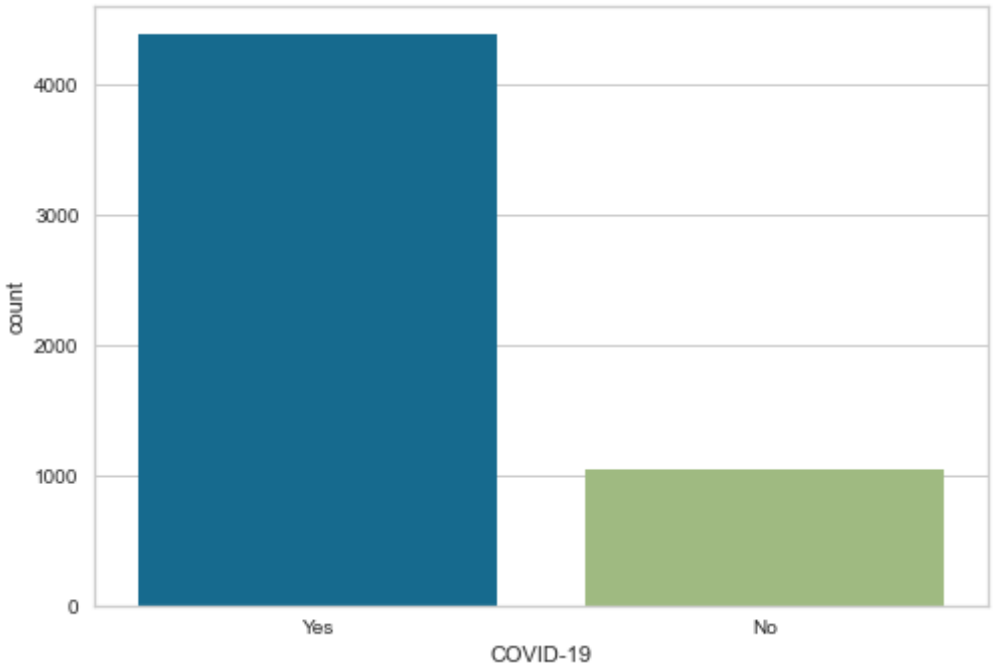
```
import seaborn as sns
```

In [80]:

```
sns.countplot(x='COVID-19',data=covid)
```

Out[80]:

```
<AxesSubplot:xlabel='COVID-19', ylabel='count'>
```




In [81]:

```
from sklearn.preprocessing import LabelEncoder
e=LabelEncoder()
```

In [82]:

```
covid['Breathing Problem']=e.fit_transform(covid['Breathing Problem'])
covid['Fever']=e.fit_transform(covid['Fever'])
covid['Dry Cough']=e.fit_transform(covid['Dry Cough'])
covid['Sore throat']=e.fit_transform(covid['Sore throat'])
covid['Running Nose']=e.fit_transform(covid['Running Nose'])
covid['Asthma']=e.fit_transform(covid['Asthma'])
covid['Chronic Lung Disease']=e.fit_transform(covid['Chronic Lung Disease'])
covid['Headache']=e.fit_transform(covid['Headache'])
covid['Heart Disease']=e.fit_transform(covid['Heart Disease'])
covid['Diabetes']=e.fit_transform(covid['Diabetes'])
covid['Hyper Tension']=e.fit_transform(covid['Hyper Tension'])
covid['Abroad travel']=e.fit_transform(covid['Abroad travel'])
covid['Contact with COVID Patient']=e.fit_transform(covid['Contact with COVID Patient'])
covid['Attended Large Gathering']=e.fit_transform(covid['Attended Large Gathering'])
covid['Visited Public Exposed Places']=e.fit_transform(covid['Visited Public Exposed Places
covid['Family working in Public Exposed Places']=e.fit_transform(covid['Family working in P
covid['Wearing Masks']=e.fit_transform(covid['Wearing Masks'])
covid['Sanitization from Market']=e.fit_transform(covid['Sanitization from Market'])
covid['COVID-19']=e.fit_transform(covid['COVID-19'])
covid['Dry Cough']=e.fit_transform(covid['Dry Cough'])
covid['Sore throat']=e.fit_transform(covid['Sore throat'])
covid['Gastrointestinal ']=e.fit_transform(covid['Gastrointestinal '])
covid['Fatigue ']=e.fit_transform(covid['Fatigue '])
covid
```

Out[82]:

| | Breathing Problem | Fever | Dry Cough | Sore throat | Running Nose | Asthma | Chronic Lung Disease | Headache | Heart Disease | Diabet |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | |
| **1** | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | |
| **2** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | |
| **3** | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | |
| **4** | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **5429** | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | |
| **5430** | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | |
| **5431** | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | |
| **5432** | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | |
| **5433** | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | |

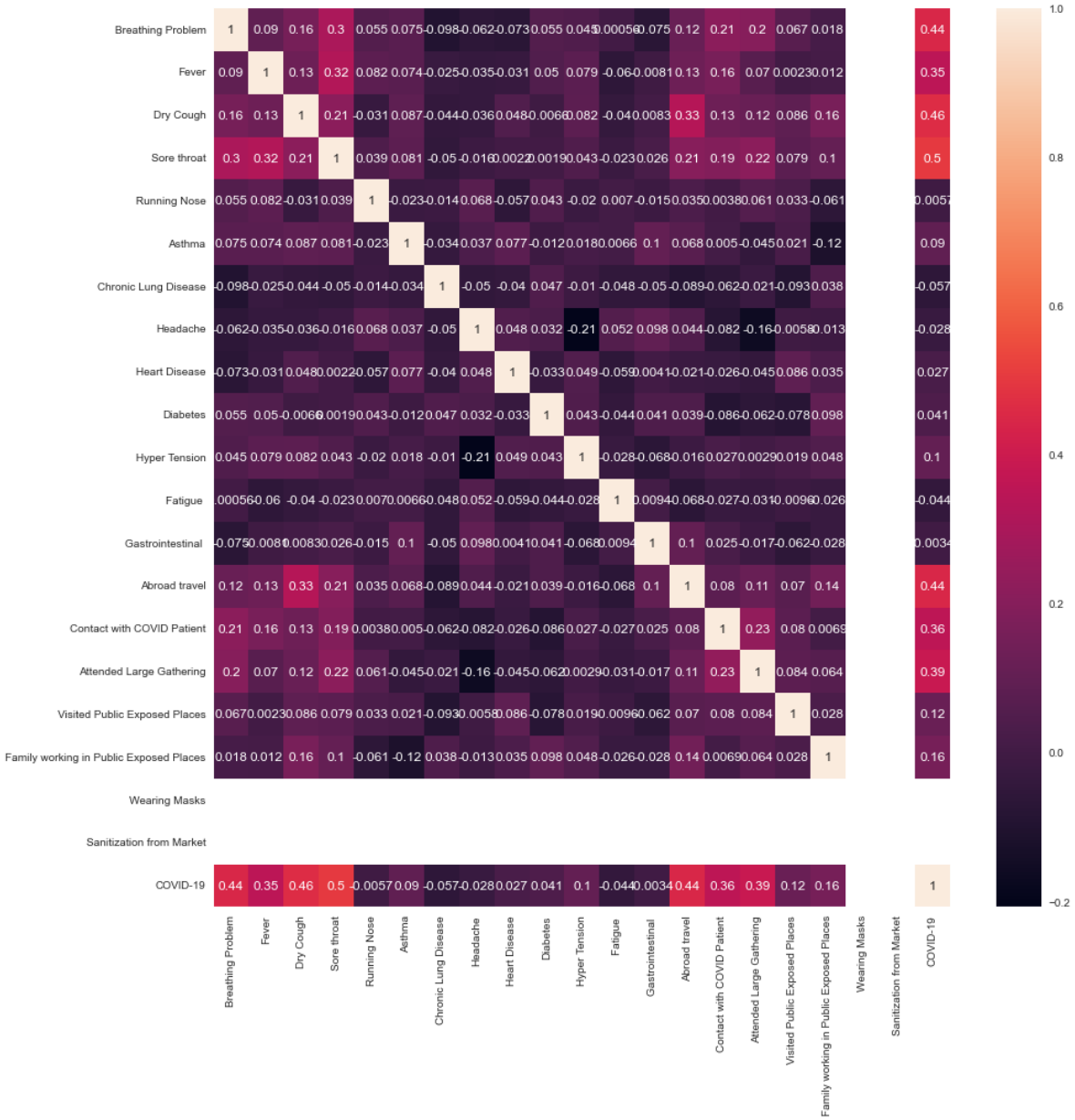
5434 rows × 21 columns

In [83]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(15,15))
sns.heatmap(covid.corr(),annot=True)
```

Out[83]:

<AxesSubplot:>



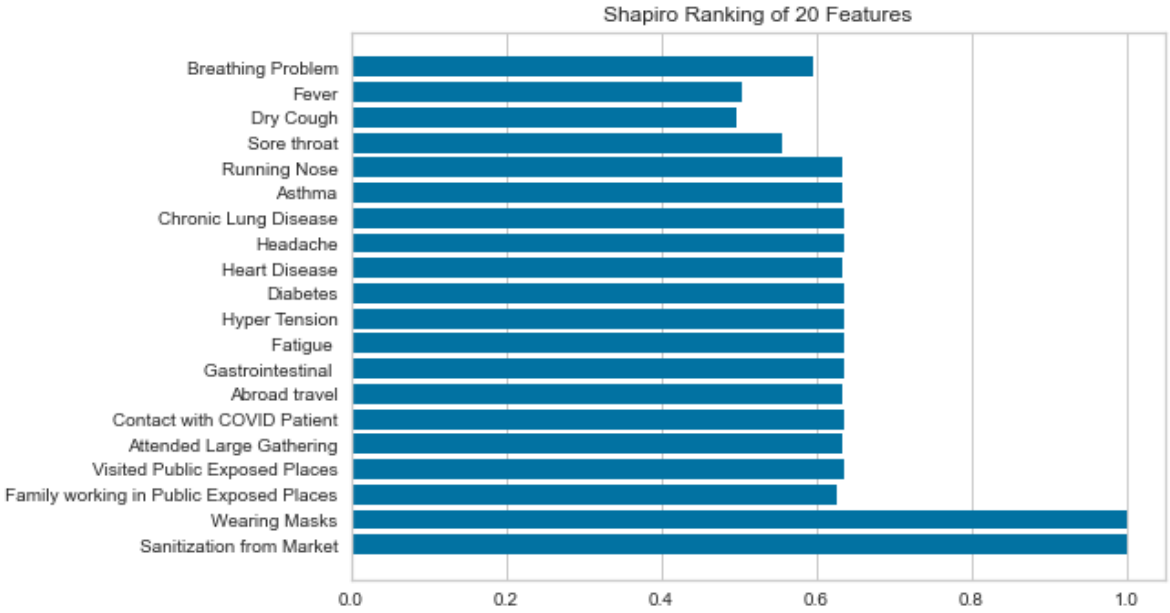
In [84]:

```
from yellowbrick.features import Rank1D


X, y = covid.drop('COVID-19',axis=1),covid['COVID-19']

# Instantiate the 1D visualizer with the Sharpiro ranking algorithm
visualizer = Rank1D(algorithm='shapiro')

visualizer.fit(X, y)           # Fit the data to the visualizer
visualizer.transform(X)        # Transform the data
visualizer.show()
```




Out[84]:

```
<AxesSubplot:title={'center':'Shapiro Ranking of 20 Features'}>
```

In [85]:

```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score
```

In [86]:

```
x=covid.drop('COVID-19',axis=1)
y=covid['COVID-19']
```

In [87]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,random_state=180)
```

# Using KneighboursClassifier

In [88]:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=20)
knn.fit(x_train, y_train)
y_pred = knn.predict(x_test)
acc_knn=(y_test==y_pred).sum()/len(y_test)
print("accuracy:",acc_knn)
```

```
accuracy: 0.9583077866339669
```

# Estimation of severity using Knn

In [89]:

```
c=pd.DataFrame({"predicted":y_pred,"actual:":y_test})
```

In [90]:

```
# Now checking whether it is seviour or moderate or easy
print(c)
c['status']='ok'
```

```
      predicted  actual:
1451          1        1
5384          1        1
827           1        1
1413          1        1
944           1        1
...         ...      ...
3082          1        1
3084          1        1
969           1        1
5136          1        1
3731          1        1

[1631 rows x 2 columns]
```

In [91]:

```
for i in c.index:
    so=0

    for j in covid.iloc[i, 0 : ]:

        if j==1:
            so+=1


    if covid['Breathing Problem'][i]==1 and (covid['Chronic Lung Disease'][i]==1 or covid['
        c['status'][i]='seviour'

    elif so>6 and covid['Fever'][i]==1:
        c['status'][i]='moderate'
    else:
        c['status'][i]='ok'
```

In [92]:

```
c['status'].value_counts()
```

Out[92]:

```
seviour     998
moderate    338
ok          295
Name: status, dtype: int64
```

In [93]:

```
c
```

Out[93]:

| | predicted | actual: | status |
|---|---|---|---|
| **1451** | 1 | 1 | seviour |
| **5384** | 1 | 1 | seviour |
| **827** | 1 | 1 | seviour |
| **1413** | 1 | 1 | seviour |
| **944** | 1 | 1 | seviour |
| **...** | ... | ... | ... |
| **3082** | 1 | 1 | seviour |
| **3084** | 1 | 1 | seviour |
| **969** | 1 | 1 | seviour |
| **5136** | 1 | 1 | moderate |
| **3731** | 1 | 1 | moderate |

1631 rows × 3 columns

# Through Decision Tree Classifier

In [94]:

```
from sklearn import tree
tree = tree.DecisionTreeClassifier()
tree.fit(x_train,y_train)
y_pre= tree.predict(x_test)
acc_decisiontree=(y_pre==y_test).sum()/len(y_pre)
print("accuracy through decision tree:",acc_decisiontree)
```

```
accuracy through decision tree: 0.9785407725321889
```

# Estimation of severity using decision tree classifier

In [95]:

```
co=pd.DataFrame({"predicted":y_pred,"actual:":y_test})
co['status']='l'
```

In [96]:

```
for i in co.index:
    so=0

    for j in covid.iloc[i, 0 : ]:

        if j==1:
            so+=1


    if covid['Breathing Problem'][i]==1 and (covid['Chronic Lung Disease'][i]==1 or covid['
        co['status'][i]='seviour'

    elif so>6 and covid['Fever'][i]==1:
        co['status'][i]='moderate'
    else:
        co['status'][i]='ok'
```

In [97]:

```
co
```

Out[97]:

| | predicted | actual: | status |
|---|---|---|---|
| **1451** | 1 | 1 | seviour |
| **5384** | 1 | 1 | seviour |
| **827** | 1 | 1 | seviour |
| **1413** | 1 | 1 | seviour |
| **944** | 1 | 1 | seviour |
| **...** | ... | ... | ... |
| **3082** | 1 | 1 | seviour |
| **3084** | 1 | 1 | seviour |
| **969** | 1 | 1 | seviour |
| **5136** | 1 | 1 | moderate |
| **3731** | 1 | 1 | moderate |

1631 rows × 3 columns

# Result comparision of both models

# Decision Tree report

In [98]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pre))
```

```
              precision    recall  f1-score   support

           0       0.91      0.98      0.95       316
           1       1.00      0.98      0.99      1315

    accuracy                           0.98      1631
   macro avg       0.95      0.98      0.97      1631
weighted avg       0.98      0.98      0.98      1631
```

# Knn Report

In [99]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.89      0.89       316
           1       0.97      0.98      0.97      1315

    accuracy                           0.96      1631
   macro avg       0.94      0.93      0.93      1631
weighted avg       0.96      0.96      0.96      1631
```

In [ ]:

```

```