Covid19_Capstone_Project.py

May 5, 2023

0.1 Introduction

This Covid data set given whole information about covid patient. COVID-19 started in the Chinese province of Hubei's Wuhan in December 2019.As the whole world was striving to combat the coronavirus disease (COVID-19), healthcare and health monitoring systems were struggling to confront the virus. Many cases had been observed where COVID-19 could not be identified at a specific time. #### The aim of this project is to make a predictive model which will predict the trajectory of the outbreak of the covid-19 virus in the upcoming days. Covid-19 is an infectious disease that is affecting a huge number of people all around the world.

The impact on healthcare systems is often reduced by effective screening, which enables early and accurate

1 Data Preprocessing :

1.0.1 1) In above dataset consists of two steps, i.e., Data Collection and Data Preprocessing. Data can be referred to as the raw material.

1.0.2 2)Data set contains 11 columns, including 8 features that were noted in 278848

1.0.3 3) The dataset's attributes include Ind_ID,Test_date , fever, shortness of breath, headache,Cough_symptoms,Sore_throat ,corona , age 60 and above, Sex,known_contact

1.0.4 4) The corona result tells whether or not people may have the coronavirus in their bodies.

1.0.5 5) Outcome variable is covid result test positive or negative.

2 The following attributes describe each of the datasets features used by the model:

1) Ind_id :It consist of indivul id of each patient

2) Test_date : We have data from 11th March 2020 till 30th April 2020.

3) Cough_symptoms :This feature is assessed into two categories, presence and absence.0 indicates the absence of cough and 1 indicates the presence of cough. After data pre-processing 0 is replaced with 0.0 and 1 is replaced with 1.0.

4) Fever–This feature is assessed into two categories, presence and absence of fever .0 indicates the absence of fever and 1 indicates the presence of fever. After data pre-processing, 0 is

replaced with 0.0 and 1 is replaced with 1.0.

5) Sore_throat: This feature is assessed into two categories, presence and absence of sore_throat .0 indicate the absence of sore_throat and 1 indicates the presence of sore_throat. After data pre-processing, 0 is replaced with 0.0 and 1 is replaced with 1.0.

6) head_ache :This feature is assessed into two categories, presence and absence of head_ache.0 indicates the absence of head_ache and 1 indicates the presence of head_ache. After data preprocessing, 0 is replaced with 0.0 and 1 is replaced with 1.0

7) Shortness_of_breath :This feature is assessed into two categories, presence and absence of shortness_of_breath.0 indicates the absence of it and 1 indicates presence. After data preprocessing, 0 is replaced with 0.0 and 1 is replaced with 1.0.

8) Age_60_and_above :This feature is assessed into three categories -'None', 'No','Yes'. No indicates that the age is below 60 and Yes indicates that the age is above 60.

9) Known_contact: This feature is assessed into three categories-Other, Abroad, Contact with Confirmed. This indicates whether an individual has come into contact with covid positive, he/she has come from abroad or there's any other reason for existing symptoms

10) Corona : This feature is assessed into three categories -Positive, Negative, and Other. Positive indicates that the individual is covid positive, negative indicates that the individual isn't covid positive, other indicates that there's no surety about the result it can be some other allergy also.

3 Data preprocessing can be defined as a process of preparing the raw data and making it suitable for a machine learning model. it's the first and crucial step while creating a machine learning model.

3.0.1 Steps that were followed during data pre-processing are:

Data preprocessing is an important process in development of machine learning model

• Imputation of missing values - In our data, missing values have been handled by using simple imputer from sklearn python package. The missing values are

• Encoding Categorical Data - We used the package of OneHotEncoder in python, this package handles categorical data by one-hot or dummy encoding scheme.

!)Importing the libraries 2)Importing the Datasets 3)Handling Missing data 4)ENCODING CATEGORICAL DATA 5)Splitting the Dataset into the Training set and Test set

3.0.2 Import the libraries

Various predefined python libraries were used for data pre- processing. Some of the libraries used are Numpy, Pandas, Seaborn, Sklearn

[25]:

```
import pandas as pd
```

```
[26]: import numpy as np
```

```
[27]: import seaborn as sns
```

```
[28]: import matplotlib.pyplot as plt
```

```
[29]: import plotly.express as px
```

```
[30]: from sklearn import metrics #Importing scikit-learn metrics module for accuracy␣
        ↪calculation
```

```
[31]: from sklearn import tree
```

3.0.3 Importing the Datasets

For performing on datasets collected for machine learning models, the present directory was set to the working directory. Then the datasets were imported. To import the dataset, the read_csv() function of the pandas library was used, which may read a CSV file

```
[2]: data = pd.read_csv("corona_tested_006.csv")
```

```
C:\Users\payal\AppData\Local\Temp\ipykernel_16724\201642362.py:1: DtypeWarning:
Columns (2,3,4,5,6) have mixed types. Specify dtype option on import or set
low_memory=False.
  data = pd.read_csv("corona_tested_006.csv")
```

```
[48]: data
```

```
[48]:         Ind_ID   Test_date Cough_symptoms  Fever Sore_throat  \
       0           1  11-03-2020           TRUE  FALSE        TRUE
       1           2  11-03-2020          FALSE   TRUE       FALSE
       2           3  11-03-2020          FALSE   TRUE       FALSE
       3           4  11-03-2020           TRUE  FALSE       FALSE
       4           5  11-03-2020           TRUE  FALSE       FALSE
       …           …         …             …      …       …
       278843  278844  30-04-2020          False  False       False
       278844  278845  30-04-2020          False  False       False
       278845  278846  30-04-2020          False  False       False
       278846  278847  30-04-2020          False  False       False
       278847  278848  30-04-2020          False  False       False

             Shortness_of_breath Headache    Corona Age_60_above     Sex  \
       0                   FALSE    FALSE  negative         None    None
       1                   FALSE    FALSE  positive         None    None
       2                   FALSE    FALSE  positive         None    None
       3                   FALSE    FALSE  negative         None    None
       4                   FALSE    FALSE  negative         None    None
       …                     …       …       …           …      …
```

```
278843               False    False  positive         None    male
278844               False    False  negative         None  female
278845               False    False  negative         None    male
278846               False    False  negative         None    male
278847               False    False  negative         None  female

                 Known_contact
0                       Abroad
1                       Abroad
2                       Abroad
3                       Abroad
4       Contact with confirmed
…                        …
278843                   Other
278844                   Other
278845                   Other
278846                   Other
278847                   Other

[278848 rows x 11 columns]
```

[3]:

```
#here we replace the sring in lower case

data = pd.DataFrame(data).replace('FALSE','False')
data
```

[3]:

```
        Ind_ID   Test_date Cough_symptoms  Fever Sore_throat  \
0            1  11-03-2020           TRUE  False        TRUE
1            2  11-03-2020          False   TRUE       False
2            3  11-03-2020          False   TRUE       False
3            4  11-03-2020           TRUE  False       False
4            5  11-03-2020           TRUE  False       False
…        …         …            …   …      …
278843  278844  30-04-2020          False  False       False
278844  278845  30-04-2020          False  False       False
278845  278846  30-04-2020          False  False       False
278846  278847  30-04-2020          False  False       False
278847  278848  30-04-2020          False  False       False

       Shortness_of_breath Headache    Corona Age_60_above     Sex  \
0                    False    False  negative         None    None
1                    False    False  positive         None    None
2                    False    False  positive         None    None
3                    False    False  negative         None    None
4                    False    False  negative         None    None
…                    …      …       …          …    …
278843               False    False  positive         None    male
```

```
278844               False    False  negative         None  female
278845               False    False  negative         None    male
278846               False    False  negative         None    male
278847               False    False  negative         None  female

                 Known_contact
0                       Abroad
1                       Abroad
2                       Abroad
3                       Abroad
4        Contact with confirmed
…                        …
278843                   Other
278844                   Other
278845                   Other
278846                   Other
278847                   Other

[278848 rows x 11 columns]
```

[4]:

```
data=pd.DataFrame(data).replace('TRUE','True')
data
```

[4]:

```
        Ind_ID   Test_date Cough_symptoms  Fever Sore_throat  \
0            1  11-03-2020           True  False        True
1            2  11-03-2020          False   True       False
2            3  11-03-2020          False   True       False
3            4  11-03-2020           True  False       False
4            5  11-03-2020           True  False       False
…        …         …            …    …         …
278843  278844  30-04-2020          False  False       False
278844  278845  30-04-2020          False  False       False
278845  278846  30-04-2020          False  False       False
278846  278847  30-04-2020          False  False       False
278847  278848  30-04-2020          False  False       False

       Shortness_of_breath Headache    Corona Age_60_above     Sex  \
0                    False    False  negative         None    None
1                    False    False  positive         None    None
2                    False    False  positive         None    None
3                    False    False  negative         None    None
4                    False    False  negative         None    None
…                    …      …       …          …     …
278843               False    False  positive         None    male
278844               False    False  negative         None  female
278845               False    False  negative         None    male
278846               False    False  negative         None    male
```

```
278847               False    False  negative         None  female

                  Known_contact
0                        Abroad
1                        Abroad
2                        Abroad
3                        Abroad
4        Contact with confirmed
…                           …
278843                    Other
278844                    Other
278845                    Other
278846                    Other
278847                    Other

[278848 rows x 11 columns]
```

```
[5]: data.isnull().sum()
```

```
[5]: Ind_ID                 0
     Test_date              0
     Cough_symptoms         0
     Fever                  0
     Sore_throat            0
     Shortness_of_breath    0
     Headache               0
     Corona                 0
     Age_60_above           0
     Sex                    0
     Known_contact          0
     dtype: int64
```

```
[277]: #data['Corona'] = data['Corona'].map({'negative': 0, 'positive': 1})
       #data['Sex'] = data['Sex'].map({'female': 0, 'male': 1})
       #data['Age_60_above'] = data['Age_60_above'].map({'No': 0, 'Yes': 1})
```

```
[6]: data['Cough_symptoms'] = data['Cough_symptoms'].map({'True': 1, 'False': 0})
     data['Fever'] = data['Fever'].map({'True': 1, 'False': 0})
     data['Sore_throat'] = data['Sore_throat'].map({'True': 1, 'False': 0})
     data['Shortness_of_breath'] = data['Shortness_of_breath'].map({'True': 1,␣
      ↪'False': 0})
     data['Headache'] = data['Headache'].map({'True': 1, 'False': 0})
```

```
[7]: data
```

```
[7]:         Ind_ID   Test_date  Cough_symptoms  Fever  Sore_throat  \
     0            1  11-03-2020             1.0    0.0          1.0
```

```
1            2  11-03-2020             0.0    1.0          0.0
2            3  11-03-2020             0.0    1.0          0.0
3            4  11-03-2020             1.0    0.0          0.0
4            5  11-03-2020             1.0    0.0          0.0
…        …         …             …    …          …
278843  278844  30-04-2020             NaN    NaN          NaN
278844  278845  30-04-2020             NaN    NaN          NaN
278845  278846  30-04-2020             NaN    NaN          NaN
278846  278847  30-04-2020             NaN    NaN          NaN
278847  278848  30-04-2020             NaN    NaN          NaN

        Shortness_of_breath  Headache    Corona Age_60_above     Sex  \
0                       0.0       0.0  negative         None    None
1                       0.0       0.0  positive         None    None
2                       0.0       0.0  positive         None    None
3                       0.0       0.0  negative         None    None
4                       0.0       0.0  negative         None    None
…                     …       …       …          …     …
278843                  NaN       NaN  positive         None    male
278844                  NaN       NaN  negative         None  female
278845                  NaN       NaN  negative         None    male
278846                  NaN       NaN  negative         None    male
278847                  NaN       NaN  negative         None  female

                 Known_contact
0                       Abroad
1                       Abroad
2                       Abroad
3                       Abroad
4       Contact with confirmed
…                        …
278843                   Other
278844                   Other
278845                   Other
278846                   Other
278847                   Other

[278848 rows x 11 columns]
```

```
[7]: data.head(1000)
```

```
[7]:
     Ind_ID   Test_date  Cough_symptoms  Fever  Sore_throat  \
0         1  11-03-2020             1.0    0.0          1.0
1         2  11-03-2020             0.0    1.0          0.0
2         3  11-03-2020             0.0    1.0          0.0
3         4  11-03-2020             1.0    0.0          0.0
4         5  11-03-2020             1.0    0.0          0.0
```

```
..      …        …             …    …          …
995     996  13-03-2020             1.0    1.0          0.0
996     997  13-03-2020             1.0    0.0          0.0
997     998  13-03-2020             1.0    0.0          0.0
998     999  13-03-2020             1.0    0.0          0.0
999    1000  13-03-2020             0.0    0.0          0.0

     Shortness_of_breath  Headache    Corona Age_60_above   Sex  \
0                    0.0       0.0  negative         None  None
1                    0.0       0.0  positive         None  None
2                    0.0       0.0  positive         None  None
3                    0.0       0.0  negative         None  None
4                    0.0       0.0  negative         None  None
..                   …       …       …          …   …
995                  1.0       0.0  negative         None  None
996                  1.0       0.0  negative         None  None
997                  1.0       0.0  positive         None  None
998                  0.0       0.0  negative         None  None
999                  0.0       0.0  negative         None  None

              Known_contact
0                    Abroad
1                    Abroad
2                    Abroad
3                    Abroad
4    Contact with confirmed
..                      …
995                   Other
996  Contact with confirmed
997                  Abroad
998  Contact with confirmed
999                   Other

[1000 rows x 11 columns]
```

[53]:
```
data.isnull().sum()
```

[53]:
```
Ind_ID                      0
Test_date                   0
Cough_symptoms         148028
Fever                  148028
Sore_throat            213313
Shortness_of_breath    213313
Headache               213313
Corona                      0
Age_60_above                0
Sex                         0
```

```
Known_contact              0
dtype: int64
```

3.0.4 Handling Missing data

If the dataset contains some missing data, then it's going to create a huge problem for our machine learning model. Therefore, it's required to handle missing values present in the dataset. The next step that was followed in data pre- processing was handling the missing data.

3.0.5 ENCODING CATEGORICAL DATA

If there are categorical variables, it can cause trouble in building the model because the machine learning model completely works on mathematics and numbers. Therefore, the specific variables were encoded into numbers using replace function. The categorical variables were converted to the following numerical values

3.0.6 1) "No" = 0.0

3.0.7 2) "Yes" = 1.0

3.0.8 3) "True" = 1.0

3.0.9 4) "False" = 0.0

ML model may think that order matters. However, there is no relationship. There will be some misinterpreted correlation. Onehot encoding convert this country column into three columns, using binary vectors. Hence, there will be no numerical order.

[8]:
```
# here im using ffill method to fill missing value
df=data.fillna(method= 'ffill')
df
```

[8]:
```
        Ind_ID   Test_date  Cough_symptoms  Fever  Sore_throat  \
0            1  11-03-2020             1.0    0.0          1.0
1            2  11-03-2020             0.0    1.0          0.0
2            3  11-03-2020             0.0    1.0          0.0
3            4  11-03-2020             1.0    0.0          0.0
4            5  11-03-2020             1.0    0.0          0.0
…        …         …             …    …          …
278843  278844  30-04-2020             0.0    0.0          0.0
278844  278845  30-04-2020             0.0    0.0          0.0
278845  278846  30-04-2020             0.0    0.0          0.0
278846  278847  30-04-2020             0.0    0.0          0.0
278847  278848  30-04-2020             0.0    0.0          0.0

        Shortness_of_breath  Headache    Corona Age_60_above     Sex  \
0                       0.0       0.0  negative         None    None
1                       0.0       0.0  positive         None    None
2                       0.0       0.0  positive         None    None
3                       0.0       0.0  negative         None    None
```

```
4                      0.0       0.0  negative         None     None
…                    …       …       …          …      …
278843                 0.0       0.0  positive         None     male
278844                 0.0       0.0  negative         None   female
278845                 0.0       0.0  negative         None     male
278846                 0.0       0.0  negative         None     male
278847                 0.0       0.0  negative         None   female

                 Known_contact
0                       Abroad
1                       Abroad
2                       Abroad
3                       Abroad
4        Contact with confirmed
…                          …
278843                   Other
278844                   Other
278845                   Other
278846                   Other
278847                   Other

[278848 rows x 11 columns]
```

[55]:
```
#check any missing value present or not
df.isnull().sum()
```

[55]:
```
Ind_ID                 0
Test_date              0
Cough_symptoms         0
Fever                  0
Sore_throat            0
Shortness_of_breath    0
Headache               0
Corona                 0
Age_60_above           0
Sex                    0
Known_contact          0
dtype: int64
```

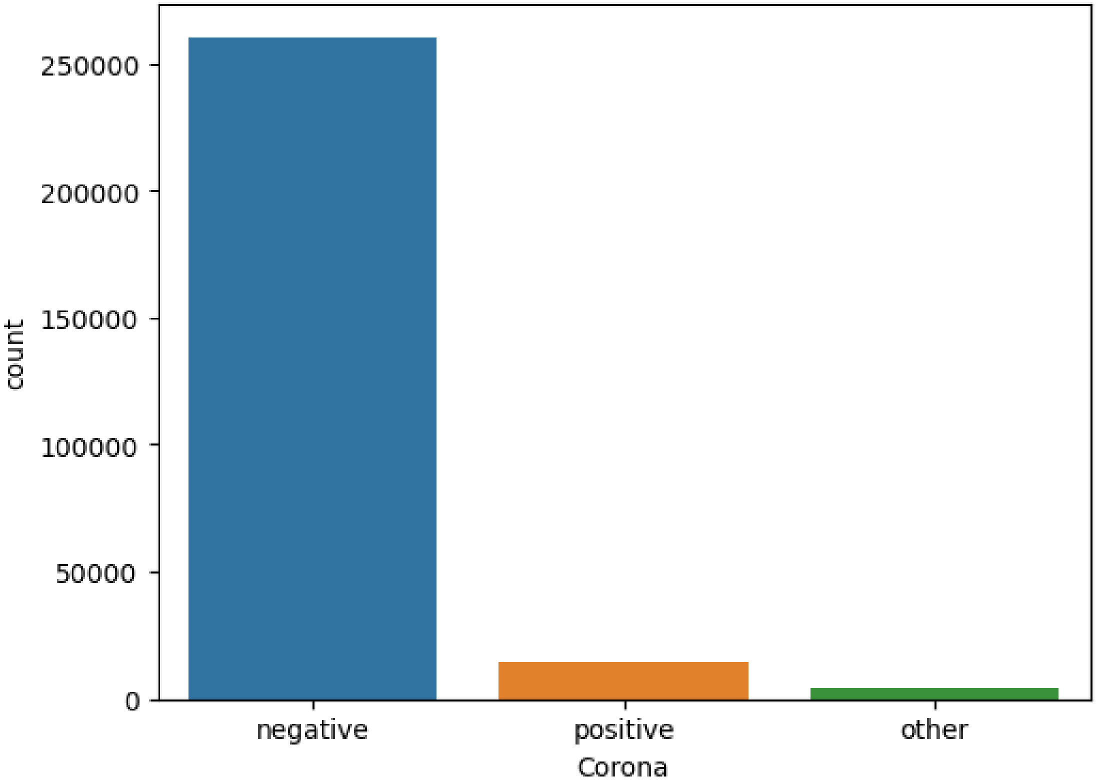
[374]:
```
# here  we use  nominal distribution  convert the value 1 and 0
#binary, only the values 0 and 1 (true/false, yes/no,) can also consider this␣
 ↪nominal/categorical. so use this.
# here i have created a dummy column of each symptoms true or false

#data=pd.
 ↪get_dummies(df,columns=["Cough_symptoms","Fever","Sore_throat","Shortness_of_breath","Heada
                  #  drop_first=True)
```

[9]:

```
# then plot grah to find relation between variables
fig = px.imshow(df.corr(), color_continuous_scale = 'rainbow_r')
fig.show()
```

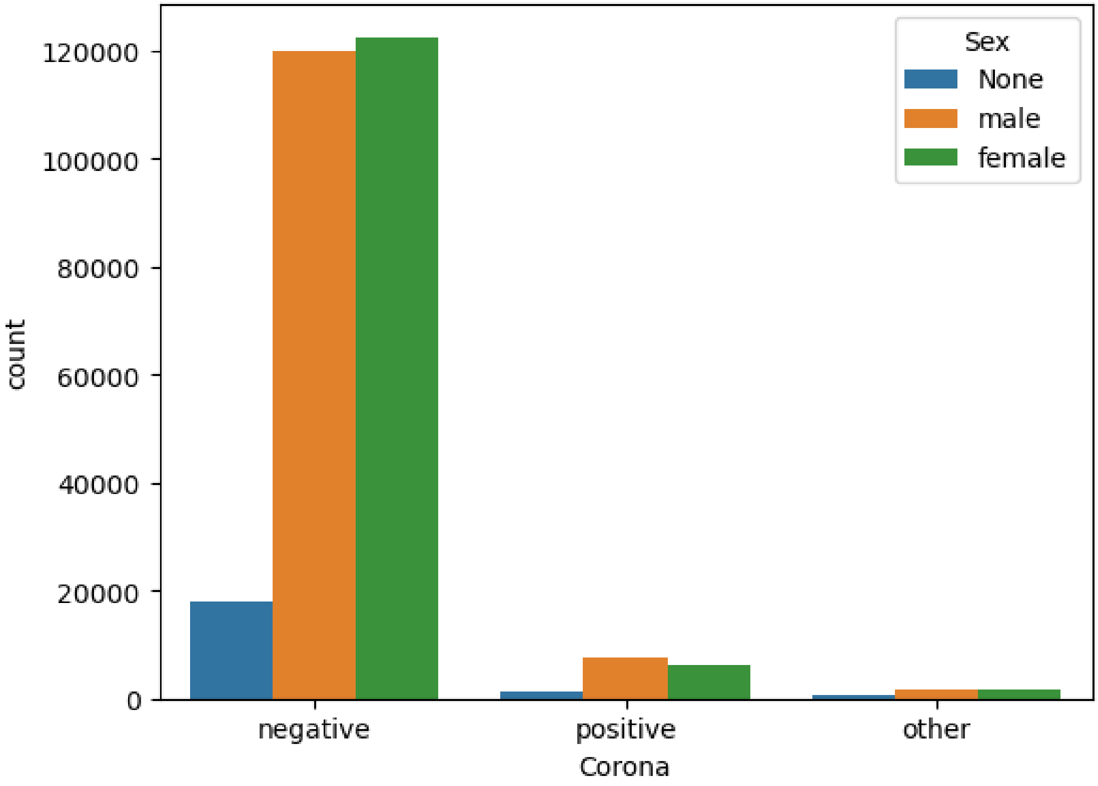
[32]:

```
# here ihave to count how many corona patient positive , negative or other
#count the number categorical data
# # x axis is 'corona' column.
sns.countplot(x='Corona',data=df)
plt.show()
```

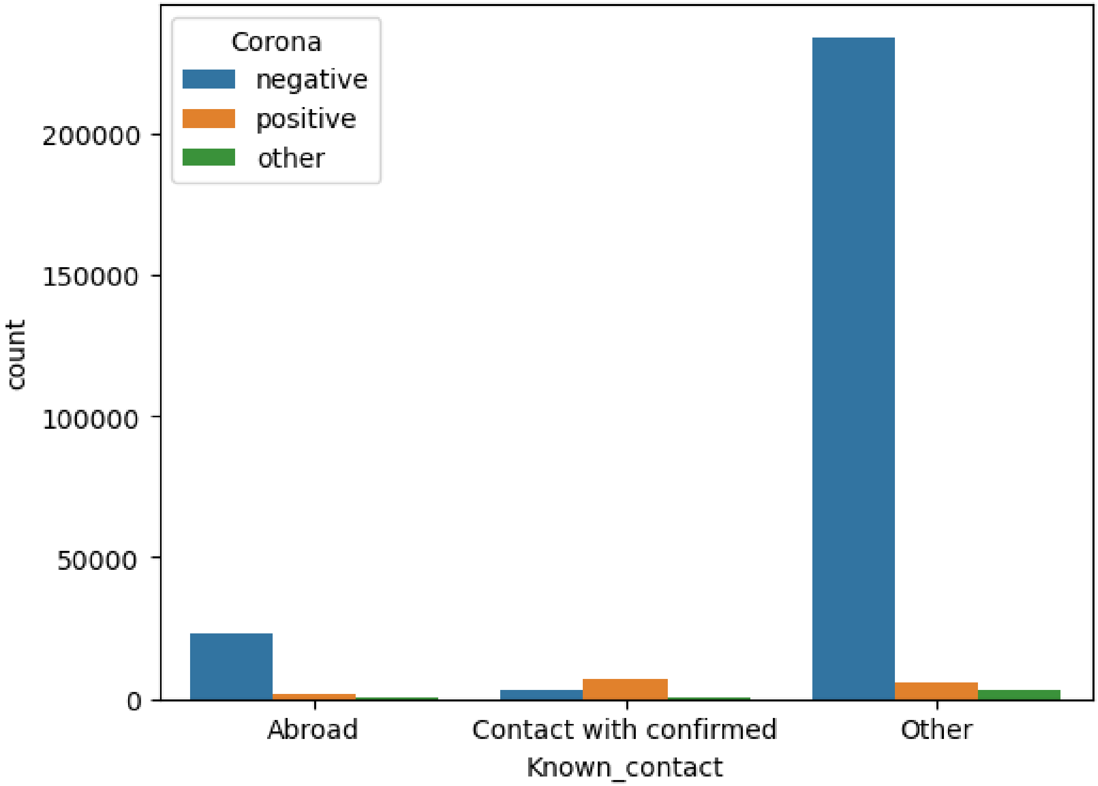


[92]:

```
sns.countplot(x='Corona',hue="Sex",data=df)  # here i have to count how many␣
 ↪male female are positve or negative
plt.show()
```

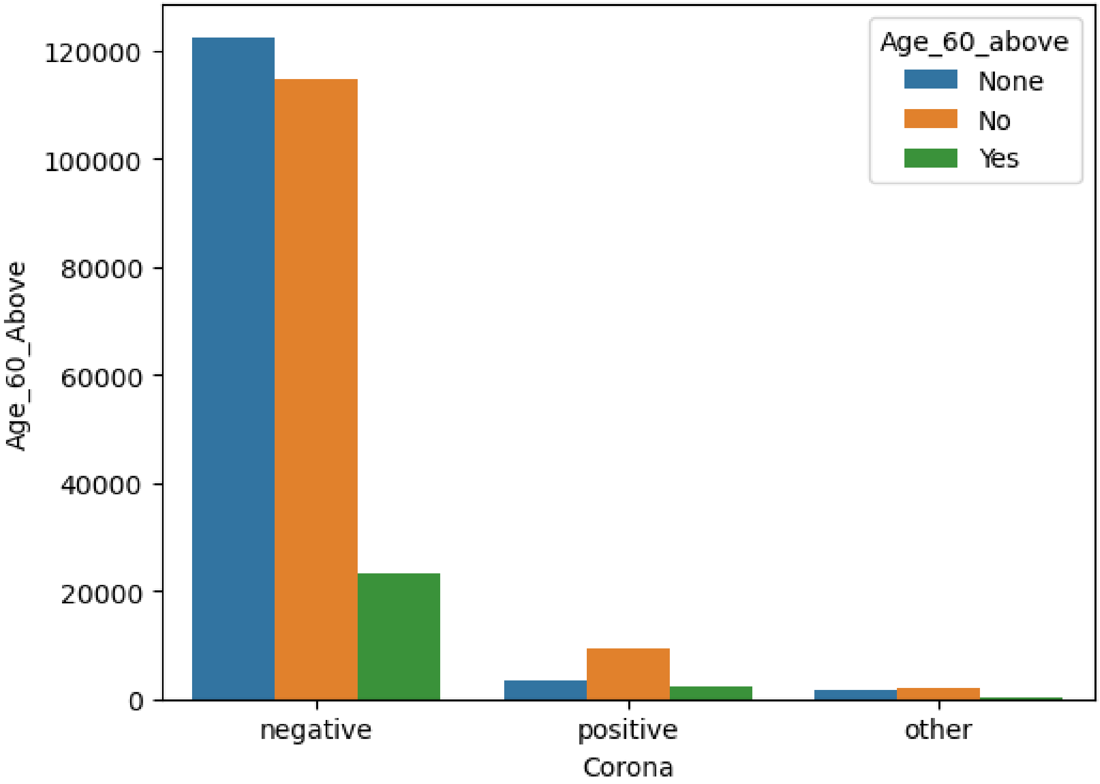


[93]:
```
#count the covid positive negative on the basic of "Known_contact"

sns.countplot(x='Known_contact', hue='Corona', data=df)
plt.show()

#so here observe that Negative patient is more in "other " categories
```



```
[94]: sns.countplot(x= "Corona", hue= "Age_60_above", data = df)
      plt.ylabel("Age_60_Above")    # count positive and negative patient  base on the␣
       ↪age
```

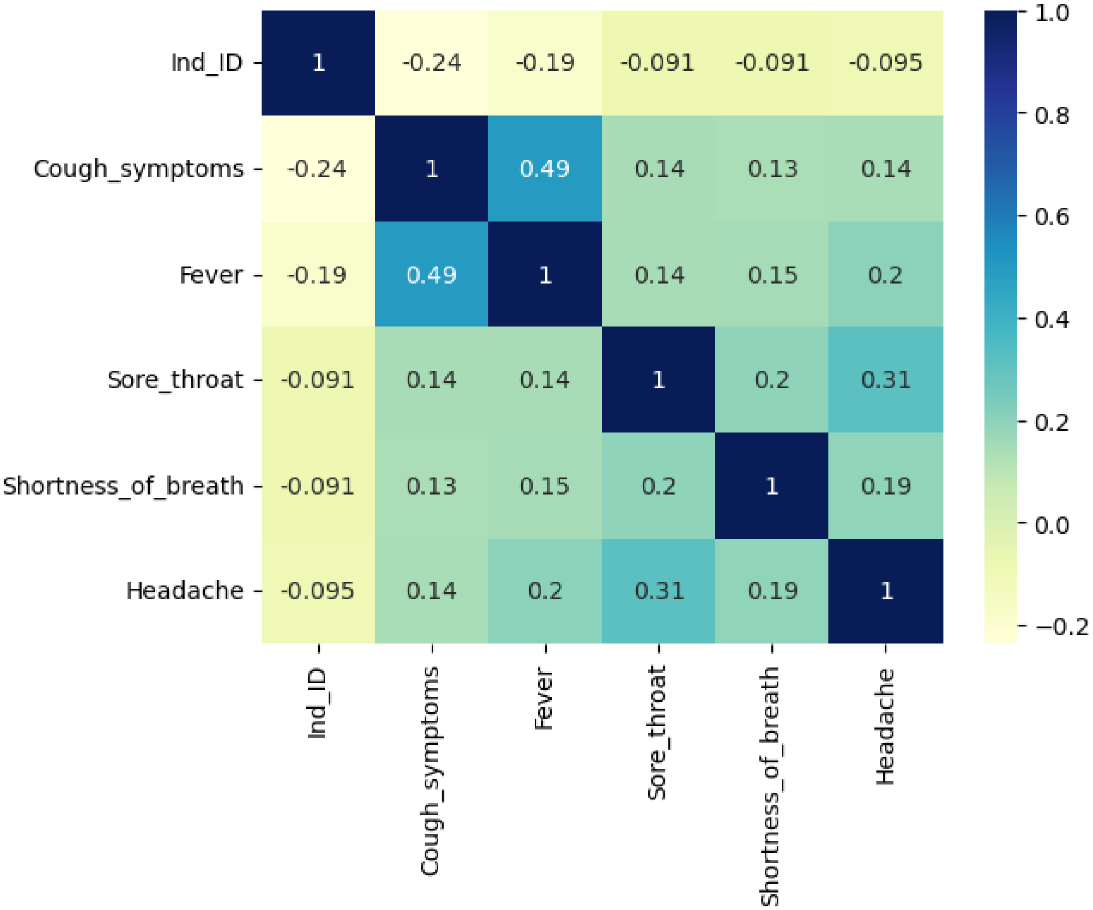
```
[94]: Text(0, 0.5, 'Age_60_Above')
```



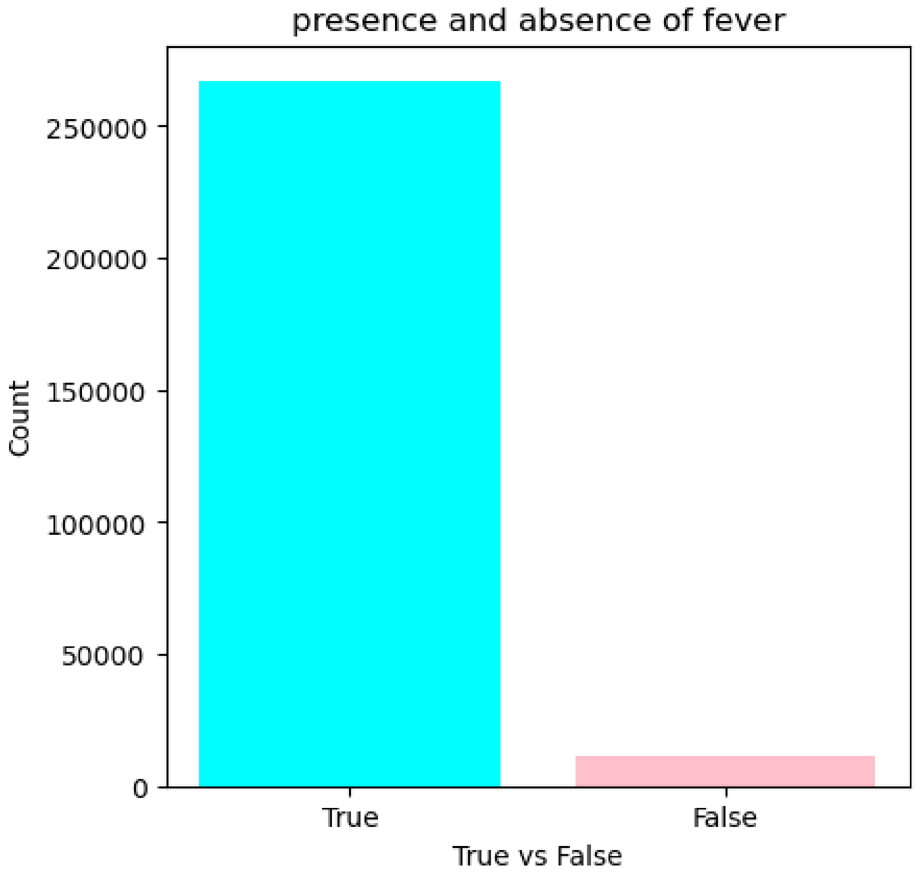
Heat map analysis is the process of reviewing and analyzing heat map data to gather insights about user interaction and behavior as they engage with your product. This data analysis can lead to improved site designs with lower bounce rates, reduced churn, fewer drop-offs, more pageviews, and better conversion rates.

```
[33]: sns.heatmap(df.corr(), cmap="YlGnBu", annot=True)

      #heat map view has a color-coded "heat map" showing which symptoms  have the␣
       ↪most (darker areas) and
      #the least (brighter areas) number of symptoms COVID-19 case
```

```
[33]: <AxesSubplot:>
```

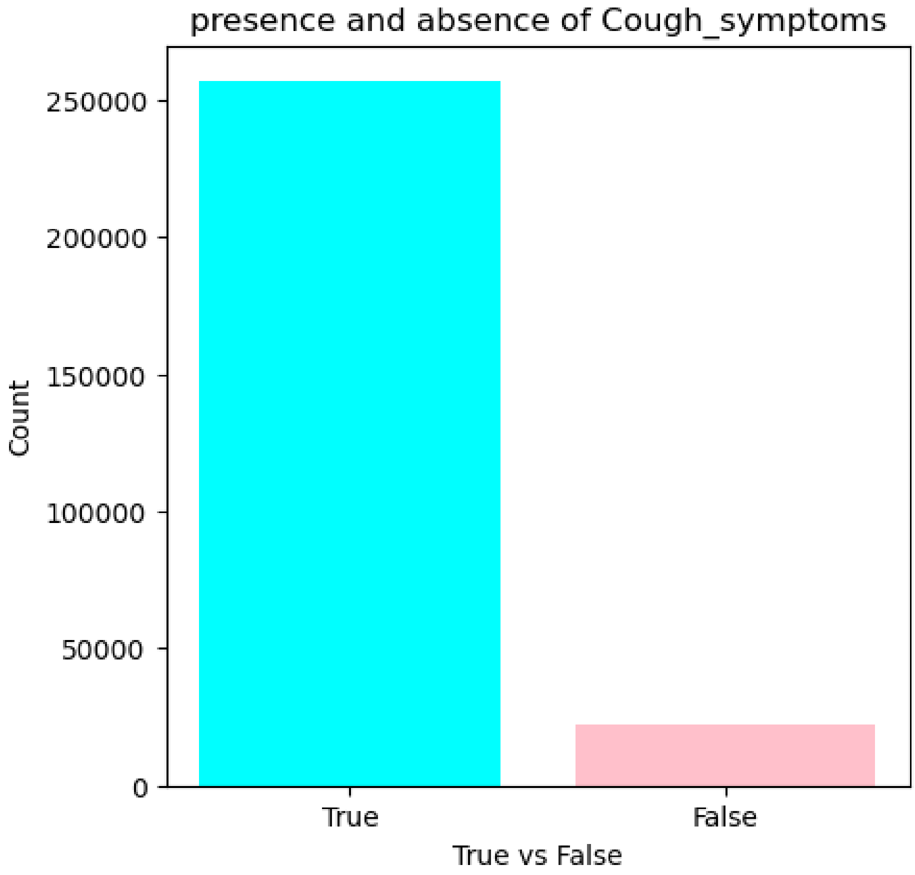


[34]:
```
count = df['Fever'].value_counts()
# Checking the numbers
count
# Creating categories based on numbers
Fever= ['True', 'False']
# Creating plot
fig = plt.figure(figsize=(5,5))
plt.bar(Fever, count, color=['cyan', 'pink'])
plt.title("presence and absence of fever")
plt.xlabel("True vs False")
plt.ylabel("Count")
# show plot
plt.show()
```



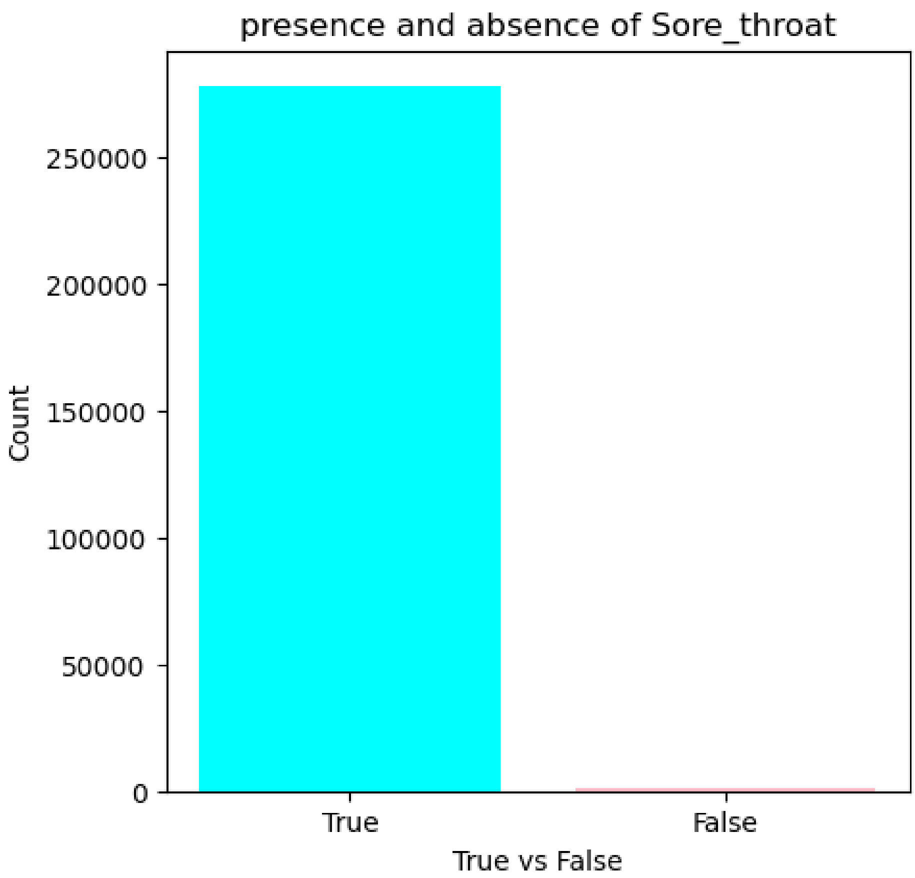
[97]:

```
count = df['Cough_symptoms'].value_counts()
# Checking the numbers
count
# Creating categories based on numbers
Cough_symptoms= ['True', 'False']
# Creating plot
fig = plt.figure(figsize=(5,5))
plt.bar(Cough_symptoms, count, color=['cyan', 'pink'])
plt.title("presence and absence of Cough_symptoms")
plt.xlabel("True vs False")
plt.ylabel("Count")
# show plot
plt.show()
```



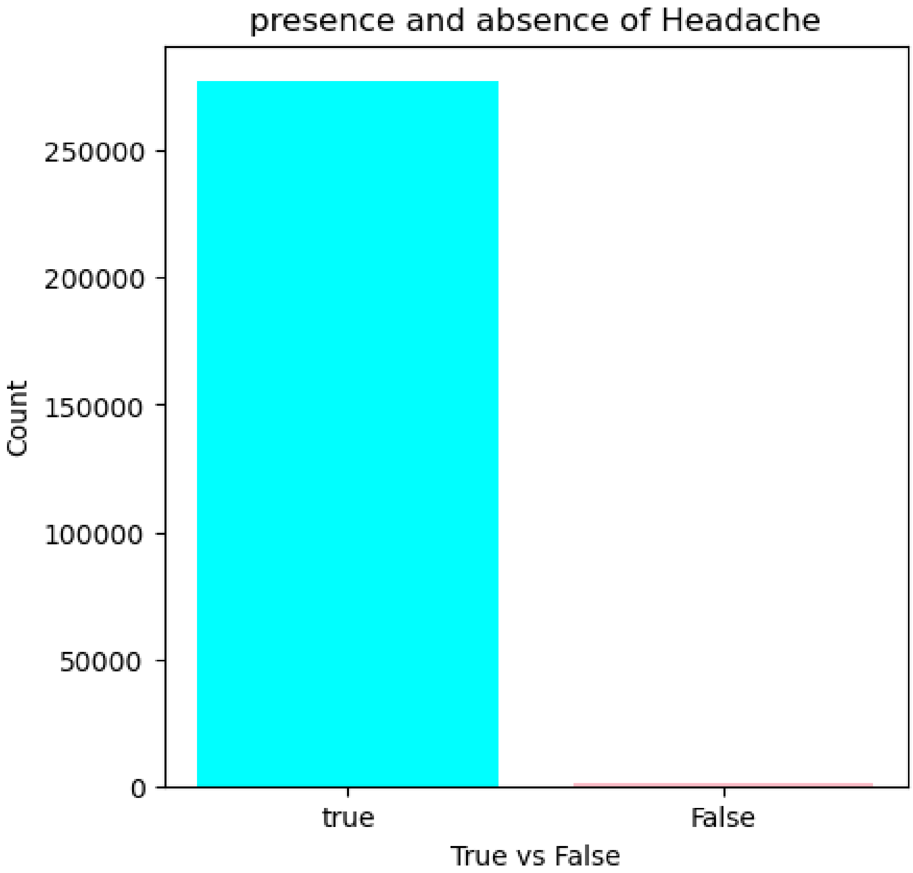
[98]:

```python
count = df['Sore_throat'].value_counts()
# Checking the numbers
count
# Creating categories based on numbers
Sore_throat= ['True', 'False']
# Creating plot
fig = plt.figure(figsize=(5,5))
plt.bar(Sore_throat, count, color=['cyan', 'pink'])
plt.title("presence and absence of Sore_throat")
plt.xlabel("True vs False")
plt.ylabel("Count")
# show plot
plt.show()
```

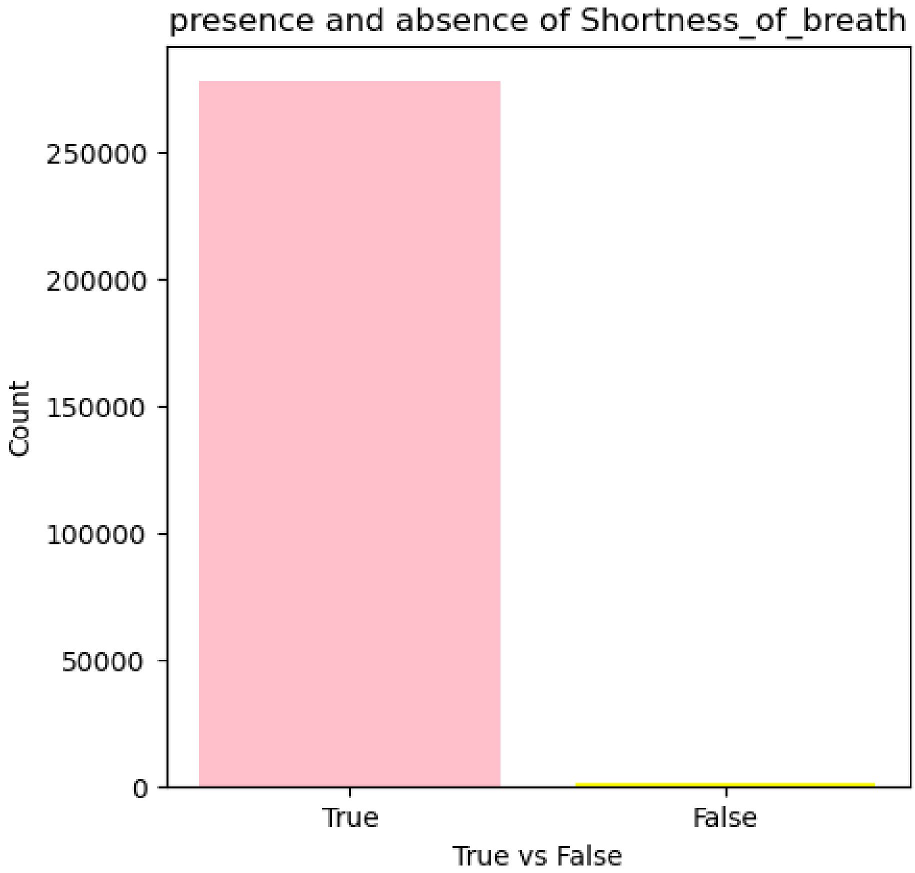


[100]:

```
count = df['Headache'].value_counts()
# Checking the numbers
count
# Creating categories based on numbers
Headache= ['true', 'False']
# Creating plot
fig = plt.figure(figsize=(5,5))
plt.bar(Headache, count, color=['cyan', 'pink'])
plt.title("presence and absence of Headache")
plt.xlabel("True vs False")
plt.ylabel("Count")
# show plot
plt.show()
```



[10]:
```
count = df['Shortness_of_breath'].value_counts()
# Checking the numbers
count
# Creating categories based on numbers
Shortness_of_breath= ['True', 'False']
# Creating plot
fig = plt.figure(figsize=(5,5))
plt.bar(Shortness_of_breath, count, color=['pink', 'yellow'])
plt.title("presence and absence of Shortness_of_breath")
plt.xlabel("True vs False")
plt.ylabel("Count")
# show plot
plt.show()
```



```
[14]: df
```

```
[14]:         Ind_ID  Test_date  Cough_symptoms  Fever  Sore_throat  \
      0            1  11-03-2020             1.0    0.0          1.0
      1            2  11-03-2020             0.0    1.0          0.0
      2            3  11-03-2020             0.0    1.0          0.0
      3            4  11-03-2020             1.0    0.0          0.0
      4            5  11-03-2020             1.0    0.0          0.0
      …          …         …               …     …            …
      278843  278844  30-04-2020             0.0    0.0          0.0
      278844  278845  30-04-2020             0.0    0.0          0.0
      278845  278846  30-04-2020             0.0    0.0          0.0
      278846  278847  30-04-2020             0.0    0.0          0.0
      278847  278848  30-04-2020             0.0    0.0          0.0

              Shortness_of_breath  Headache    Corona Age_60_above   Sex  \
      0                       0.0       0.0  negative         None  None
      1                       0.0       0.0  positive         None  None
      2                       0.0       0.0  positive         None  None
```

```
3                          0.0       0.0  negative         None    None
4                          0.0       0.0  negative         None    None
…                        …       …       …          …      …
278843                     0.0       0.0  positive         None    male
278844                     0.0       0.0  negative         None  female
278845                     0.0       0.0  negative         None    male
278846                     0.0       0.0  negative         None    male
278847                     0.0       0.0  negative         None  female

                 Known_contact
0                       Abroad
1                       Abroad
2                       Abroad
3                       Abroad
4        Contact with confirmed
…                            …
278843                   Other
278844                   Other
278845                   Other
278846                   Other
278847                   Other

[278848 rows x 11 columns]
```

*When should we do feature scaling? Is it before or after splitting the dataset?

Feature scaling should be done after splitting the dataset. For example, if we do before mean and standard deviation will be from all the values including the one's from test set. Test set is not supposed to have information from training set. If we use all values for feature scaling, it would lead to information leakage on the test set. Test set is supposed to be new data or new observation.

```
[10]: #### Selecting Independent variables,,,,,,inp
      X = df.iloc[:,2 :7].values
      X
```

```
[10]: array([[1., 0., 1., 0., 0.],
             [0., 1., 0., 0., 0.],
             [0., 1., 0., 0., 0.],
             …,
             [0., 0., 0., 0., 0.],
             [0., 0., 0., 0., 0.],
             [0., 0., 0., 0., 0.]])
```

```
[11]: #### Selecting Dependent variable..output variable
      y = df.iloc[:, -4].values
      y
```

```
[11]: array(['negative', 'positive', 'positive', …, 'negative', 'negative',
              'negative'], dtype=object)
```

```
[17]: y.shape
```

```
[17]: (278848,)
```

4 Splitting the Dataset into the Training set and Test set

The covid 19 dataset was divided into 80% of the dataset as the training set and 20% as the test set. Training Set can be described as a subset of dataset to coach the machine learning model, and we already know the output. Test set can be defined as a subset of the dataset to check the machine learning model, and by using the test set, the model predicts the output.

```
[12]: from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,␣
       ↪random_state = 2)
```

```
[13]: X_train.shape
```

```
[13]: (209136, 5)
```

```
[14]: X_train
```

```
[14]: array([[0., 0., 0., 0., 0.],
             [0., 0., 0., 0., 0.],
             [0., 0., 0., 0., 0.],
             …,
             [0., 0., 0., 0., 0.],
             [0., 0., 0., 0., 0.],
             [0., 0., 0., 0., 0.]])
```

```
[15]: y_train
```

```
[15]: array(['negative', 'negative', 'negative', …, 'negative', 'negative',
              'positive'], dtype=object)
```

```
[16]: print(X.shape,X_train.shape,X_test.shape)

      (278848, 5) (209136, 5) (69712, 5)
```

```
[17]: print(y.shape,y_train.shape,y_test.shape)

      (278848,) (209136,) (69712,)
```

5 ## Feature Scaling

6 Development of the machine learning models

six models that are trained using six machine learning algorithms which are as follows:

7 Random forest Algorithm

Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning Thus, the future development of COVID-19 can be predicted by selecting epidemic parameters with similar trends in past periods.

```
[20]: from sklearn.ensemble import RandomForestClassifier
      classifier= RandomForestClassifier(n_estimators= 10, criterion="entropy")
      classifier.fit(X_train, y_train)
```

```
[20]: RandomForestClassifier(criterion='entropy', n_estimators=10)
```

```
[22]: y_pred= classifier.predict(X_test)
```

```
[23]: print(y_pred)
```

```
['negative' 'negative' 'negative' … 'positive' 'negative' 'negative']
```

```
[24]: print(f"RandomForestClassifier training set accuracy: {format(classifier.
        ↪score(X_train, y_train), '.4f')} ")
      print(f"RandomForestClassifier testing set accuracy: {format(classifier.
        ↪score(X_test, y_test), '.4f')} ")
```

```
RandomForestClassifier training set accuracy: 0.9388
RandomForestClassifier testing set accuracy: 0.9403
```

8 Decision Tree Algorithm

One of the widely used supervised type machine learning methods for classification and regression is the decision tree algorithm. * It also known as classification and regression tree (CART). * According to predetermined principles, data is constantly divided in this algorithm at each row till the final result is obtained. * Decision trees classify the results into groups until no more similarity is left. * Decision tree is non-parametric approach and does not depend on any probability distribution assumptions. * Decision tree is non-parametric approach and does not depend on any probability distribution assumptions.

```
[33]: from sklearn.tree import DecisionTreeClassifier # Importing Decision Tree␣
        ↪Classifier
```

```
[34]: # Create Decision Tree classifer object
      classification = DecisionTreeClassifier()
```

```

# Train Decision Tree Classifer
classification = classification.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = classification.predict(X_test)
```

```
[35]: print(f"Decision tree training set accuracy: {format(classification.
       ↪score(X_train, y_train), '.4f')} ")
      print(f"Decision tree testing set accuracy: {format(classification.
       ↪score(X_test, y_test), '.4f')} ")
```

```
Decision tree training set accuracy: 0.9388
Decision tree testing set accuracy: 0.9403
```

Advantages**:

Easy to understand and create. Can be applicable for both regression and classification. A robust model with excellent outcomes. Handle large data efficiently. Handle training data well with less effort.

Disadvantages:

Instability: Decision tree works well if the information is precise and accurate. A slight change in input may change the tree drastically.

9 Conclusion:

The COVID-19 pandemic has impacted millions of lives worldwide as a significant public health concern. Therefore, we all need to work together to end it and return to normality. In this study, a COVID-19 database was created and used for Data Mining on some of those data. The COVID-19 predictive model obtained good accuracy in the classification tests with all the algorithms used. The best algorithm was RandomForestClassifier training set accuracy: 0.9388 and Decision tree testing set accuracy: 0.9403