Machine learning-based prediction of COVID-19 diagnosis based on symptoms

Introduction

The COVID-19 pandemic has presented an immense challenge to healthcare systems worldwide, highlighting the need for accurate and efficient methods of diagnosing the disease. While laboratory tests such as polymerase chain reaction (PCR) and antigen tests have been pivotal in detecting the presence of the SARS-CoV-2 virus, they often require specialized equipment, time-consuming processes, and limited testing capacity. As a result, there is a growing demand for alternative approaches that can leverage readily available data, such as symptoms, to predict COVID-19 diagnosis.

The aim of this project is to contribute to the ongoing efforts in developing a machine learning-based predictive model for COVID-19 diagnosis, focusing specifically on symptoms as the input data. By analyzing large-scale datasets encompassing diverse populations, clinical records, and symptom profiles, we intend to identify patterns and correlations that can enable accurate predictions of COVID-19 status.

Importance of the project

accurate disease prediction based on symptoms, such as the proposed machine learning-based model for COVID-19, is crucial in today's world. It allows for early identification, optimized resource management, proactive public health measures, early intervention, personalized care, and informed decision-making. By improving disease prediction accuracy, we can enhance medical treatment and contribute to better health outcomes for individuals and communities.

Impact of the project

The implementation of an effective screening tool based on accurate disease prediction can have a profound impact on the medical field, particularly in terms of screening efficiency and reducing the burden on healthcare systems. The implementation of an accurate disease prediction model for effective screening has the potential to revolutionize the medical field. It improves screening efficiency, enables early detection and intervention, optimizes resource allocation, reduces the burden on healthcare systems, and leads to cost savings. By enhancing the effectiveness and efficiency of screening processes, healthcare providers can deliver better care to patients, reduce healthcare burden, and improve overall healthcare outcomes.

Future of the project

While the proposed method focuses specifically on predicting COVID-19 diagnosis based on symptoms, it has the potential to be adapted and applied to other infectious diseases in the future. The methodology and principles employed in this project can serve as a foundation for developing predictive models for different diseases. The proposed method can address knowledge gaps by providing a framework for predicting COVID-19 diagnosis based on symptoms. This framework can be adapted and applied to other diseases by leveraging transfer learning, feature engineering, and data integration. By utilizing the knowledge gained from the development and implementation of the COVID-19 predictive model, future disease prediction efforts can be accelerated, contributing to improved diagnostic capabilities and public health preparedness.

Hypothesis

Based on the dataset provided which includes variables such as various symptoms, Corona result, Age_60_above, Gender, and Known_contact details, we can make initial hypotheses:

1. Contact with positive person will result in covid result positive
2. As I had covid in the pandemic I can tell shortness of breath, fever and cough sumptoms were main symptoms among positive patients.

Data Analysis

Observations about all features

Sex column:

1. In the sex column 3 features are there- male, female, none
2. None feature is missing value. It has 19034 values. We can't drop them as it will impact model. Even imputation wont work as it's not depend on any other feature. We will change it to 'others'.

Cough Symptoms column

1. In the cough_symptoms column 3 features are there-true, false, none
2. None feature is missing value and it has 254 observations. Number is very small so we can drop them.

Fever column

1. In the Fever column 3 features are there-true, false, none
2. None feature is missing value and it has 254 observations. Number is very small so we can drop them.

Sore throat column

1. In the Sore throat column 3 features are there-true, false, none
2. None feature is missing value and it has 254 observations. Number is very small so we can drop them.

Headache column

1. In the Headache column 3 features are there-true, false, none
2. None feature is missing value and it has 1 observation. Number is very small so we can drop them.

Corona column

1. This is our Target column
2. In the Corona column 3 features are there-true, false, other
3. Other feature is missing value and it has 3892 observations. As it is our target column we can't afford having missing values here. Number is very small so we can drop them.

Age above 60 column
1. In the Age_60_above column 3 features are there-Yes, No, none
2. None feature is missing vale and has 127320 which is very big number. we have big number of missing values We need to use method 'Develop a Model to Predict Missing Values' because by we can't delete the observations as number is big, can't use most frequent value as it will be biased opinion, these variables are important so we can't delete the variable and unsupervised machine learning technique is time consuming method.

Known contact column
1. In the Known_contact column 3 features are there-Abroad, contact with known contact, other
2. As per the data there is no missing value because other means any kind of reason can be. So we will keep other column intact. We will use one hot encoding for this feature and then drop third column as from 1st two we can get status of third column.

Conclusions about data:

For independent Variables:

After checking the missing values we got to know that from independent variables 'age above 60' and 'sex' columns have big number of missing values which. There are some methods to deal with missing values
1. Step 1: Delete the Observations, 2. Step 2: Replace Missing Values with the Most Frequent Value, 3. Step 3: Develop a Model to Predict Missing Values, 4. Step 4: Deleting the variable, Step 5: Apply unsupervised Machine learning techniques
For columns like cough, fever, soar throat, shortness of breath and headache has very minimal missing values like 1 or 252. So we have to delete those observations as they are less than 1%.
Where we have big number of missing values We need to use method 'Develop a Model to Predict Missing Values' because by we can't delete the observations as number is big, can't use most frequent value as it will be biased opinion, these variables are important so we can't delete the variable and unsupervised machine learning technique is time consuming method.

For independent Variables:

Here in column 'corona' number of missing values are 3892 which is very less so we can delete these observations.

After removing missing values from cough, fever, soar throat, shortness of breath, headache and Corona we have removed total 3892 observations in total which is less than 2% of the data. Which won't impact our model. For remaining colums we have to use predicting model.

Feature Selection:
1. After chi2 test we can see just sex column has more P_value but it's not more than 0.5 so we can not ignore it.
2.No feature seems unimportant.
3. All the features are contributing towards the detection of covid cases.

Data modelling:
Train-Test Split
Train- 11th March till 15th April as a training and validation set.
Test- From 16th April till 30th April as a test set.
Dividing training and validation set at a ratio of 4:1.

Final conclusion from all 4 models:

1. With Linear Regression we get accuracy 97.8%

2. With Decision Tree we get accuracy 98%

3. With Random Forest we get accuracy 98%

4. With K Neighbours Classifier we get accuracy 98%

All 4 models are ok to use.

SQL part of the project:

For SQL part date column format was not supported so first I changed date format in cleaned dataset and then used it for sql part.

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

In [2]: data=pd.read_csv('/Users/bididudy/Downloads/clean_data_covid_006.csv', low_memory=False)

In [3]: data['Test_date'] = pd.to_datetime(data['Test_date'],format = '%d-%m-%Y')

In [5]: data['Test_date'] = pd.to_datetime(data['Test_date'].dt.strftime('%Y-%m-%d'))

In [40]: data
         ...

In [6]: data.drop('Unnamed: 0', axis = 1, inplace = True)
        data
         ...

In [42]: data.to_csv('covid_clean_data_file.csv', header=True, index=False)
```

Here are the Queries, and output of the queries.

1. Find the number of corona patients who faced shortness of breath.

```
select count(Ind_Id) as Patients
from clean_data
where Corona = "positive" and Shortness_of_breath = "TRUE";
```

Answer- 1162 patients.

2. Find the number of negative corona patients who have fever and sore_throat.

```
select count(Ind_Id) as Patients
from clean_data
where Corona = "negative"  and Fever = "TRUE"  and Sore_throat = "TRUE";
```

Answer- 121 patients.

3. Group the data by month and rank the number of positive cases.

```
select monthname(Test_date) as Month,
count(Ind_Id) as positive,
```

```
RANK() OVER(order by count(Ind_Id) DESC) AS 'rank'
from clean_data
where Corona = "positive"
group by monthname(Test_date);
```

Answer- April month with 8863 cases.

4. Find the female negative corona patients who faced cough and headache.

```
select * from clean_data
where Sex = "female" and Corona = "negative" and Cough_symptoms = "TRUE"
and Headache = "TRUE";
```

Answer- 32 patients.

5. How many elderly corona patients have faced breathing problems?

```
select count(*)
from clean_data
where Age_60_above = "Yes" and Shortness_of_breath = "TRUE";
```

Answer :- 286 patients

6. Which three symptoms were more common among COVID positive patients?

```
select
        (select count(*) from clean_data
   where Cough_symptoms = "TRUE" and corona = "positive") as Cough,

   (select count(*) from clean_data
   where Fever = "TRUE" and corona = "positive") as Fever,

   (select count(*) from clean_data
   where Sore_throat = "TRUE" and corona = "positive") as Sore_throat,

   (select count(*) from clean_data
```

```
   where Shortness_of_breath = "TRUE" and corona = "positive") as
short_Breath,

   (select count(*) from clean_data
   where Headache = "TRUE" and corona = "positive") as headache,

   (select count(*) from clean_data
   where corona = "positive") as positive
from clean_data
limit 1;
```

Answer :- Cough_Symptoms, Fever and Headache

7. Which symptom was less common among COVID negative people?

```
select
        (select count(*) from clean_data
   where Cough_symptoms = "TRUE" and corona = "negative") as Cough_count,

   (select count(*) from clean_data
   where Fever = "TRUE" and corona = "negative") as Fever_count,

   (select count(*) from clean_data
   where Sore_throat = "TRUE" and corona = "negative") as Sore_count,

   (select count(*) from clean_data
   where Shortness_of_breath = "TRUE" and corona = "negative") as
shortBreath_count,

   (select count(*) from clean_data
   where Headache = "TRUE" and corona = "negative") as headache_count,

   (select count(*) from clean_data
   where corona = "negative") as negative_count
from clean_data
limit 1;
```

Answer- Headache, Shortness of Breath, Sore throat.

Q.8 What are the most common symptoms among COVID positive males whose known contact was abroad?

```
select
        (select count(*) from clean_data
   where Cough_symptoms = "TRUE" and corona = "positive" and
Known_contact like '%Abroad%') as Cough,

   (select count(*) from clean_data
   where Fever = "TRUE" and corona = "positive" and Known_contact like
'%Abroad%') as Fever,

   (select count(*) from clean_data
   where Sore_throat = "TRUE" and corona = "positive" and Known_contact like
'%Abroad%') as Sore,

   (select count(*) from clean_data
   where Shortness_of_breath = "TRUE" and corona = "positive" and
Known_contact like '%Abroad%') as shortBreath,

   (select count(*) from clean_data
   where Headache = "TRUE" and corona = "positive" and Known_contact like
'%Abroad%') as headache,

   (select count(*) from clean_data
   where corona = "positive"  and Known_contact like '%Abroad%') as positive
from clean_data
limit 1;
```

Answer- Cough and Fever.