SPRINGBOARD

DATA SCIENCE CAREER TRACK

COVID-19 Diagnosis using Raman Spectroscopy

Author:
Isaac Ghebregziabher

Capstone 2 submitted as requirement for Certificate in Data Science at Springboard

May 2021

Contents

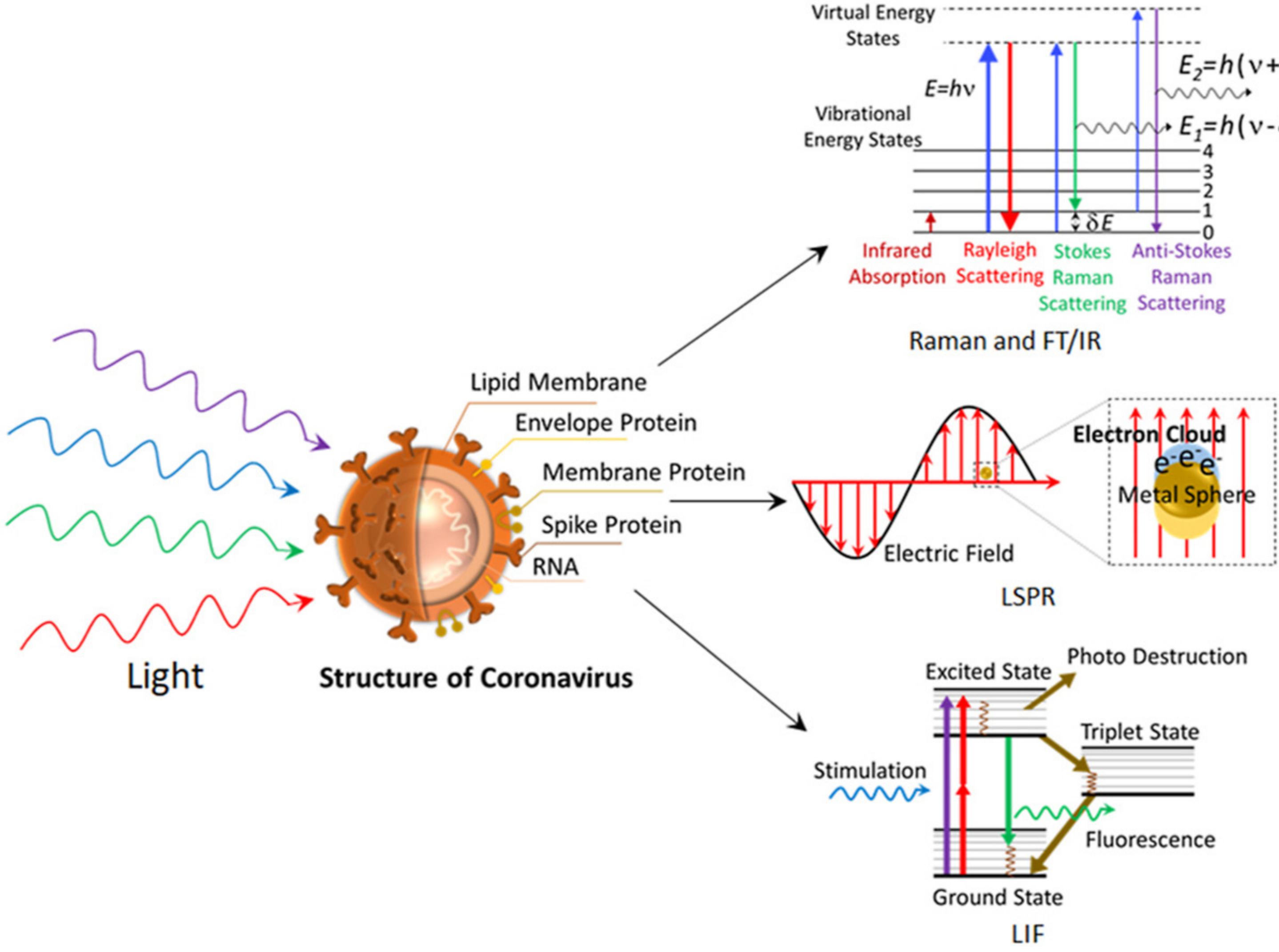
Chapter 1

Introduction and objective

Since its first reported case in Wuhan in December 2019, COVID-19 has rapidly spread globally and is now a global pandemic. At the time of this writing, 163 million active cases with 3.4 million deaths have been reported. The diagnosis tools available so far have been based on a) viral gene detection, b) human antibody detection, and c) viral antigen detection, among which the viral gene detection by real time polymerase chain reaction (RT-PCR) has been found as the most reliable technique. However, it has been reported that COVID-19 diagnosis using RT-PCR is not without limitations. RT-PCR though very sensitive, testing has to be conducted in a sophisticated and controlled laboratory environment and increases the turn around time. Though other diagnostic have been developed for fast bedside tests, they are not as accurate as they detect antigens instead of the virus itself. COVID-19 detection employing RT-PCR using Serum obtained from respiratory tract has been reported to have false negative results due to insufficient amount of the virus in the serum.

At the moment a fast and reliable diagnostics technique is needed to combat this rapidly progressing pandemic outbreak. We propose Raman spectroscopy as a safe and efficient method for diagnosing COVID-19.

Using dataset consisting of Raman spectra of serum samples collected from 150 patients and 150 healthy control individuals, we develop a machine learning model based on Logistic regression, decision trees and Random Forest to seamlessly detect COVID-19., A schematic diagram depicting the principle of Raman spectroscopy for virus detection is depicted below (image adapted from [1].


Virtual Energy States
E=hν
Vibrational Energy States
$E_2=h(\nu+$
$E_1=h(\nu-$
4
3
2
1
0
δE
Infrared Absorption
Rayleigh Scattering
Stokes Raman Scattering
Anti-Stokes Raman Scattering
Raman and FT/IR
Lipid Membrane
Envelope Protein
Membrane Protein
Spike Protein
RNA
Light
Structure of Coronavirus
Electron Cloud
e⁻e⁻e⁻
Metal Sphere
Electric Field
LSPR
Excited State
Photo Destruction
Triplet State
Stimulation
Fluorescence
Ground State
LIF

Chapter 2

Target stakeholders and why

My target stakeholders include medical device developer companies, hospitals, and food and drug administration (FDA). In my opinion it is clear that fast and reliable diagnostic medical device is needed to combat effectively the current pandemic and any future virus outbreaks. The development of a fast and reliable COVID-19 and any future virus outbreak would allow medical and government entities to comprehend the extent and exact number of cases on time and take informed appropriate actions. Moreover, the current high-rate of false negatives could be reduced with the use of this technique and hence eliminating unwanted spread of the virus by the predictted false negatives.

Chapter 3

Dataset acquisition

The dataset was obtained from Kaggle. The dataset from kaggle was already prepossessed including smoothing, baseline correction and normalized by total spectral area. The dataset is in CSV format and consists of Raman spectra data of 150 healthy cases and 158 Covid cases. The first row of the dataset consists of raman spectral shift of wavenumbers and consists of wavenumbers from 400 to 2112 (total columns = 900). One additional last column is designated for the target variable named ’diagnostic’. The values of diagnostic are either ’Healthy’ or ’SARS-Cov2’. The remaining rows of the dataset (row number 2 to row number 310) correspond to Raman intensity values for each of the Raman wavenumbers.

initial dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Columns: 901 entries, 400 to diagnostic
dtypes: float64(900), object(1)
memory usage: 2.1+ MB
```

It looks like all features (900 columns) of our dataset are of type float while the target variable **diagnostic** is of type string (object).

Chapter 4

Data Wrangling

The following steps were necessary to cleanse and modify the dataset into a format suitable for analysis.

- Check and count missing values in our dataset.
- Handle all zero value feature columns.
- Check unique values of target variable **diagnostic**.
- Check whether our data is balanced by looking at target distribution.

4.1. Count missing values

To cleanse and modify our dataset, we first checked whether we have any missing values and if there are how many? The following code snippet was used to check for missing values.

```
[1]: # Count missing values, if any.
     missing = pd.concat([covid_data.isnull().sum(), 100 * covid_data.isnull().
       ↪mean()], axis=1)
     missing.columns=['count', '%']
     grouped_sum = missing.groupby(by=['count']).count()
     grouped_sum
```

```
[1]:          %
     count
     0      901
```

The number of missing values in our dataset is 0. Our dataset is clean and there is no need to drop or impute missing values.

4.2. Columns with all zero values

In general missing values are represented by ’nan/NaN’ values. As shown above there exist no ’nan/NaN’ values in our dataset. However, it is also common to represent missing values with other numbers such as ’-999/0’. Also if sampling is done properly, the values of all feature columns should be normally distributed. In our dataset we observed that some feature columns have constant values. Specifically, some columns have all their values equal to zero, which indicates the values of the feature column might be missing during data collection. The following code snippet is what we used to identify and count how many feature columns we have with all their values equal to 0.

[6]:
```
# Get columns with all zero values
zero_covid_data = covid_data.loc[:, ~(covid_data != 0).any(axis=0)]
zero_covid_data.shape
```

[6]:
```
(309, 9)
```

We have 9 feature columns with 0 values and we dropped them. A code snippet to drop the columns and its corresponding output is shown below. **final dataset:**

[7]:
```
# drop columns with all values zero: zeroFree_Covid_data
zeroFree_covid_data = covid_data.loc[:, (covid_data != 0).any(axis=0)]
zeroFree_covid_data.head()
zeroFree_covid_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Columns: 892 entries, 402 to diagnostic
dtypes: float64(891), object(1)
memory usage: 2.1+ MB
```

Our final dataset containing 891 features with float values and 1 categorical/string valued target column was saved for farther analysis.

4.3. Check unique values of target variable diagnostic

Since our objective is to develop a supervised classification machine learning model, it is prudent to check how many unique classes we have in our target variable. Using the following code snippet, we were able to see the number of unique values of target ’diagnostic’.

[8]:
```
covid_data['diagnostic'].value_counts().head()
```

[8]:
```
SARS-CoV-2    159
Healthy       150
Name: diagnostic, dtype: int64
```

Chapter 5

Exploratory data analysis

We performed data prepossessing and analysis before performing a more rigorous predictive data analysis using supervised machine learning models. The sequence of data prepossessing we conducted are listed below.

- Check whether our data is balanced by looking at target distribution.
- Visualize and contrast the average Raman spectra for COVID and Healthy cases.
- Feature/dimensional reduction using principal component analysis (PCA).

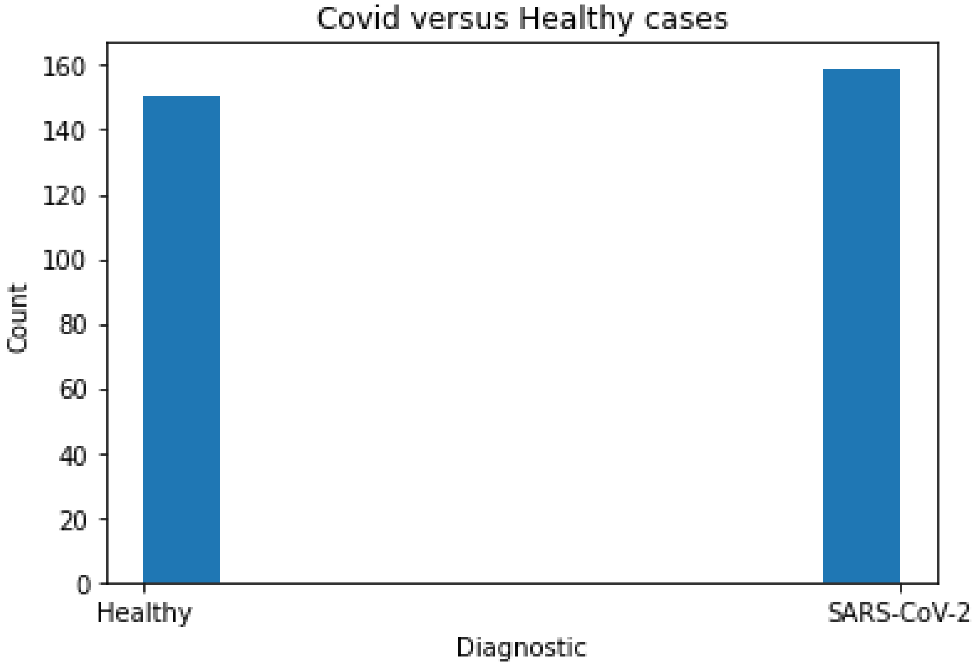
5.1. Check data for imbalance

It is well known that degree of data imbalance is one of the parameters that one needs to consider in choosing what type of Model Metrics needs to employ in assessing model performance. For instance use of ’accuracy’ metric is not recommended when data is highly imbalanced. Depending on the problem at hand, ’recall’ or ’precision’ or a combination are among the possible appropriate metrics to assess model performance for highly imbalanced data. For balanced dataset, though the metric of appropriate choice might depend on the nature of the problem the model sets out to solve, the use of ’accuracy’ metric is justified as a preliminary metric of performance assessment.

A histogram plot of the ’diagnostic’ target feature as shown in 5.1 shows our data is well balanced with almost a split in half between the two classes. Hence, the use of the ’accuracy’ metric during our Model development is justified.

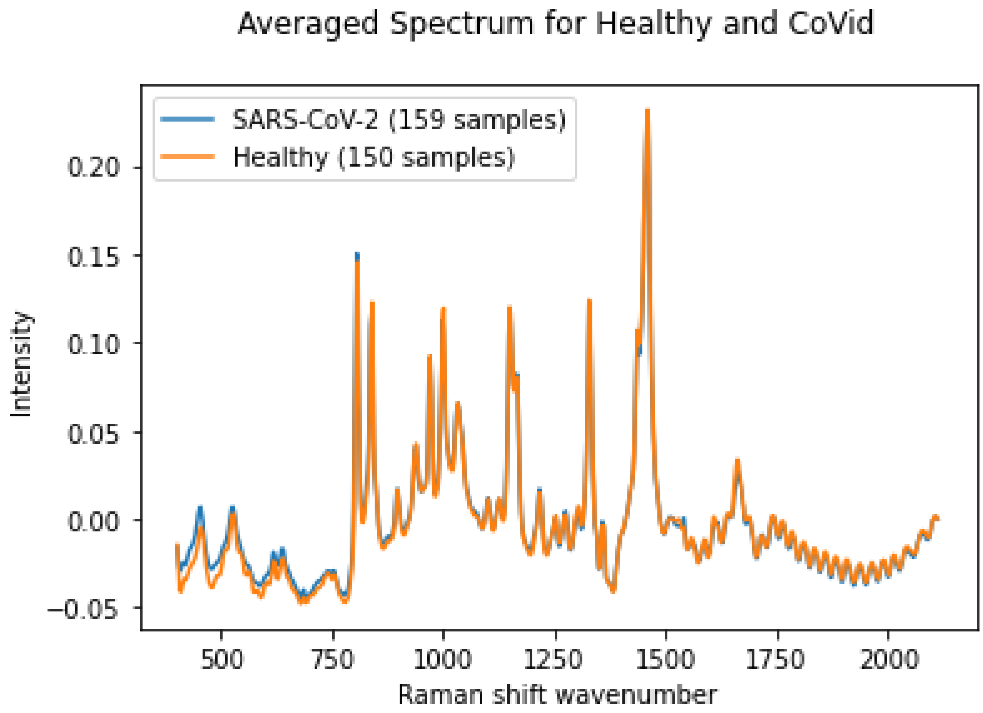
5.2. Raman spectra for COVID and Healthy cases

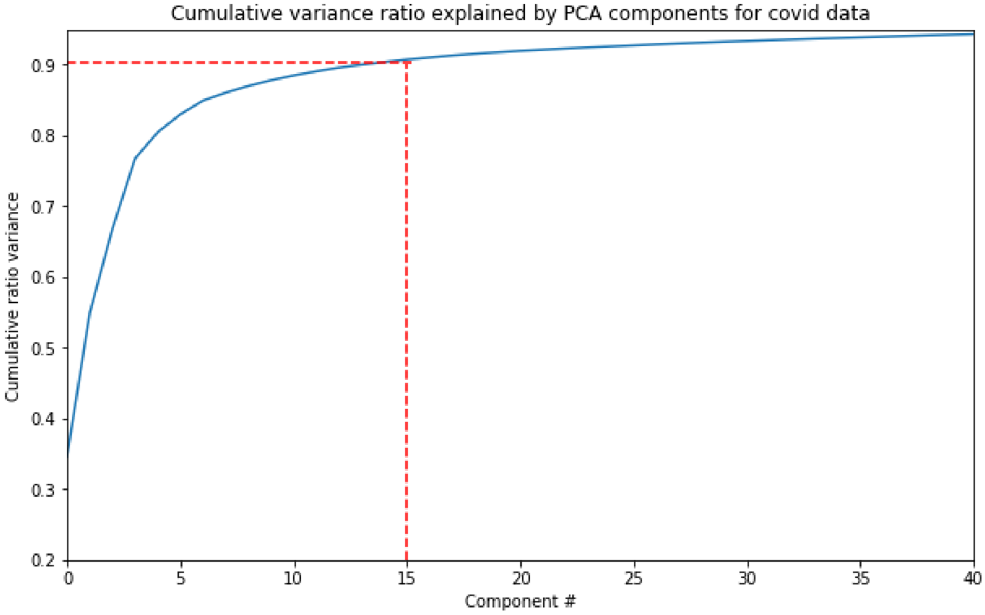
Figure 5.2 shows the average Raman spectra for COVID cases and Healthy individuals. As can be seen from the plot, the difference in the spectra for healthy and COVID cases is very small and occurs for some discrete number of Raman shift wave-numbers. visually indiscernible. This suggests that there is room to reduce the number of features that needs to be considered when developing our model. The principal component analysis will be employed in the following section to reduce the number of features in our dataset.



5.3. PCA for feature reduction

After scaling the numerical feature of our dataset, we employed PCA to reduce the number of features from 894 to 15. The explained variance of the dataset versus the number of principal components is shown in Figure 5.3. As can be seen from the figure, over 90 percent of the variance in the dataset is explained by only 15 principal components.


Averaged Spectrum for Healthy and CoVid
SARS-CoV-2 (159 samples)
Healthy (150 samples)
Intensity
0.20
0.15
0.10
0.05
0.00
−0.05
500
750
1000
1250
1500
1750
2000
Raman shift wavenumber


Cumulative variance ratio explained by PCA components for covid data
Cumulative ratio variance
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
5
10
15
20
25
30
35
40
Component #

Chapter 6

Modeling and Machine Learning

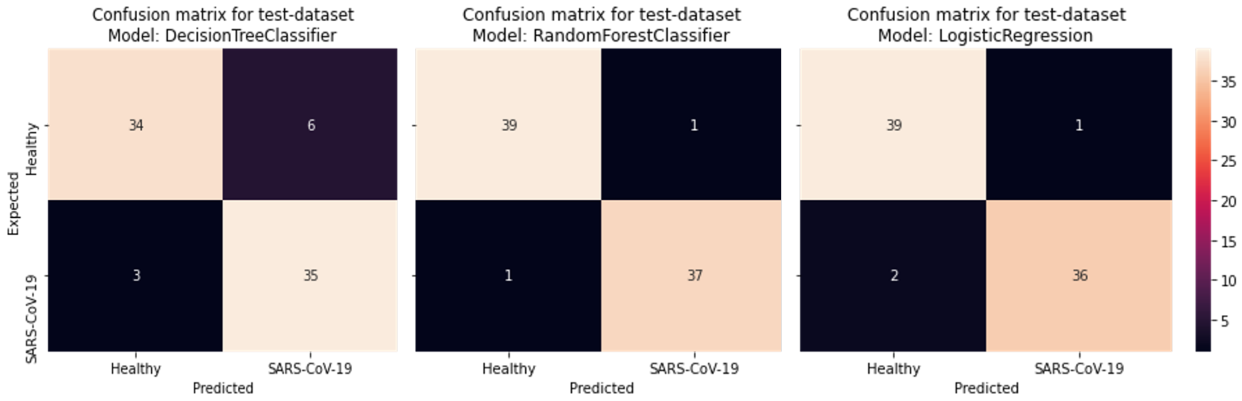
The main objective the Modeling and Machine Learning is to identify the best classifier model that enables us to accurately diagnose COVID. The model we seek to develop is a supervised machine learning classification model.

The dataset was first split into training and test set with 0.75 and 0.25 split ratios, respectively. With the training split, a decision tree, logistic regression, and random forest supervised machine learning classification algorithms were trained. All models resulted to an accuracy of 100 percent on the training data set as can be seen from a snippet of the out of our code below.

```
[0] Logistic Regression Accuracy:  1.0
[0] Decision Tree Accuracy:  1.0
[0] Random Forest Accuracy:  1.0
```

It looks like all models are over-fitting and should be tested with the test split. The heatmap plot of the confusion matrix along with the accuracy score is shown in Figure 6.1.

```
Model:  DecisionTreeClassifier
Testing accuracy:  0.8846153846153846
Model:  LogisticRegression
```



```
Testing accuracy:  0.9615384615384616
Model:  RandomForestClassifier
Testing accuracy:  0.9743589743589743
```

On the test split, decision tree performed the least with an accuracy of about 0.88. Random Forest performed the best with an accuracy of about 0.97. We are satisfied with the performance of the random forest classification model. In deployment we choose Random Forest for diagnosing COVID.

Bibliography

[1] Jijo Lukose., Santhosh Chidangil, and Sajan D. George. Optical technologies for the detection of viruses like covid-19: Progress and prospects. *Biosensers and Bioelectronics*, 178:113004. pages 2