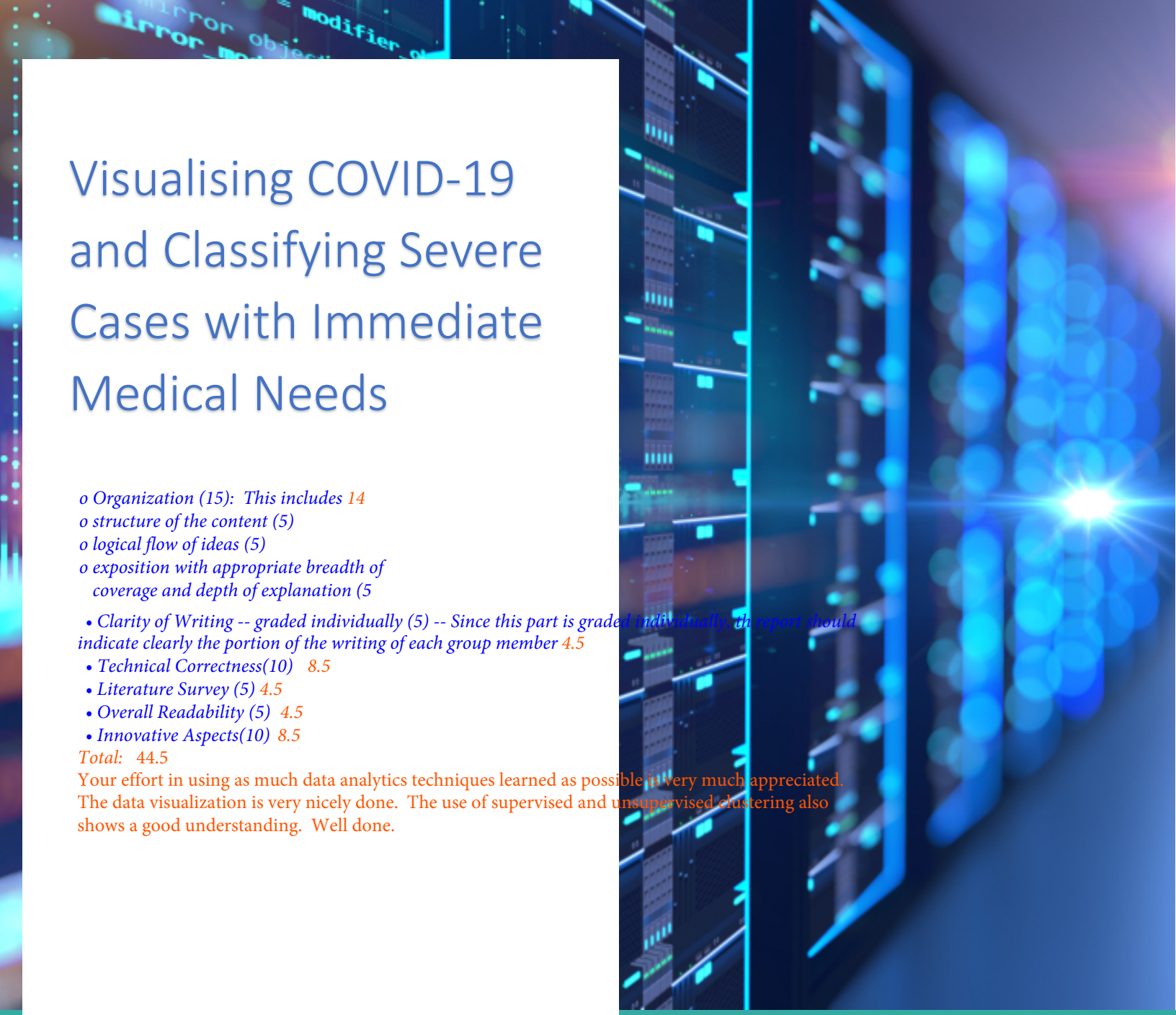

Visualising COVID-19 and Classifying Severe Cases with Immediate Medical Needs

o Organization (15): This includes 14
o structure of the content (5)
o logical flow of ideas (5)
o exposition with appropriate breadth of coverage and depth of explanation (5

- *Clarity of Writing -- graded individually (5) -- Since this part is graded individually, th report should indicate clearly the portion of the writing of each group member 4.5*
- *Technical Correctness(10) 8.5*
- *Literature Survey (5) 4.5*
- *Overall Readability (5) 4.5*
- *Innovative Aspects(10) 8.5*

Total: 44.5

Your effort in using as much data analytics techniques learned as possible is very much appreciated. The data visualization is very nicely done. The use of supervised and unsupervised clustering also shows a good understanding. Well done.

MAY 25, 2020

SEEM2460 Final Project

Authored by:

Yun Ki, LEUNG	**1155109801**
King Yeung, CHAN	**1155119394**
Tsz Chun, LAI	**1155125208**

1. Introduction

1.1. Background

We may hear “YOU STAY AT HOME FOR US” so many times because of the tremendous pressure on the medical industry. We want to give doctors a hand on making a decision based on the urgency of patients from the novel coronavirus. To understand the current circumstances quickly, we are going to visualise the overall trends and classify the issue of death from the novel coronavirus. We anticipate the outcome can assist professionals to decide a better judgement on healing the patients.

At the late end of 2019, the novel coronavirus, also known as COVID-19, has become an epidemic and communicable diseases around the world. The novel coronavirus has not only influenced the economy over the world but also put unprecedented pressure on medical personnel around the world. Due to the lack of information, a lot of medical personnel was not able to save lives from the novel coronavirus. For a general citizen without the knowledge of professional medicine, we do not have such abilities to take off their shoulders. We understand that they are fighting for us in the front line, therefore, we want to try our best to relieve their pressure in the view of making decisions. We provide essential information, such as the classification of confirmed patients with the potential death and the urgency of a patient who suffers from the novel coronavirus. We hope the results can help doctors to decide the treatment of each patient, such that they can put more resources and effort into those classified patients and save their lives. In the project, consequently, we aim to relieve the stress on medical personnel.

1.2. Data Source

Nice introduction.

To establish the model we are concerning, we are going to grab the data set *Novel Corona Virus 2019 Dataset* which published from Kaggle. Kaggle is famous for many competitions about Machine Learning and Data Science. The data set that we have chosen contains daily information on the novel coronavirus. The sources of data are organised on a daily basis from the international organisation, such as the World Health Organisation (WHO). We believe the data are reliable with such reason. The model building relies on the training data and the testing data. As a result, the outcomes of the project are biased on the data. Thus, we assume the provided data are dependable.

Should insert citations to the information sources.

2. Abstract

For visualisation, the word cloud gives us an insight into the top-ranked keywords related to the novel coronavirus. “Trump” occurred the most in the streaming data. Moreover, from the line plot, we found out that the number of confirmed, death and recovered cases are increasing as time passes. We also adopted the simple linear regression model to obtain the fact of the positive association between the confirmed and death cases.

For supervised learning, the decision tree model trained with 70% of the data and tested with the remaining data for the accuracy. The scores of Precision, Recall, and the F-measure is 0.7895, 0.9091 and 0.8451 respectively. Since the scores of evaluation close to 1, the probability of error is not significant. Thus, the model has a high precision for labelling a confirmed patient whether he or she has a high likelihood to survive in the novel coronavirus.

For unsupervised learning, we found out that gender is not a dominant factor in defending the novel coronavirus, that is no matter male or female share the same issue of surviving from the novel coronavirus. Older patients with shorter “time to death” is classified at high risk. Younger patients or those with longer “time to death” are less urgent to have immediate medical needs.

3. Visualisation

Before any further studies on the project, we want to grab some general information about the novel coronavirus. Visualisation is a robust technique to inspect the behaviour of the data, such that we can have a better understanding of the actual situation before achieving our goals. In the era of the Internet of Things, streaming data has become a target source of data to investigate the behaviour in real-time. We are going to make use of streaming data from twitter to explore the hot keywords that frequently related to the "novel coronavirus". Furthermore, we also handle the overall trends of the number of confirmed, death and recovered cases about the novel coronavirus. R software is being favoured for the section of visualisation since it is a powerful programming language to tackle a huge amount of data.

3.1. Keywords Related to the Novel Coronavirus

In the age of phubbing[1], social media became an indispensable communication channel among all the cities around the world. People would like to share their daily activities and their opinions on social media, such as Instagram and Snapchat. Such a phenomenon motivates us to study the connection between the novel coronavirus and other common keywords. For those tweets containing "the novel coronavirus" and "COVID 19" is being favoured to our purpose. We collected streaming data from Twitter on May 08, 2020, from 22:25 for 60 seconds.

Figure 1: word cloud for the related keywords

Very nice visualization.Some of the stop words can be removed.

We adopted a word cloud to identify the top-ranked keywords appeared in the tweets related to the novel coronavirus. From figure 1, it is crystal clear that the word "Trump" comes up the most in the streaming time. We know that it refers to the president of the United States. It is reasonable since the US is one of the leading countries of the world and in charge of many international policies. As we may understand that the greater power comes great responsibility, Donald Trump plays an important role in fighting with the novel coronavirus. Moreover, "death" is also a popular keyword shows up the figure above. The novel coronavirus becomes a threat to human beings, and mortality becomes a common concern to our public awareness. The attention of such an issue motivates to the study of the project.

[1] Phubbing is a term that describes the habit of snubbing someone in favor of a cellphone

3.2. Overall Trends about the Number of Cases (Worldwide)

After perceiving the related keywords of the novel coronavirus, we employed a line plot to describe the trends of the cumulative number of confirmed, deaths and recovered cases. From figure 2 on the right-hand side, we can see that number of confirmed cases is incredibly increasing from the middle of the graph. The situation became more critical since March 2020. The number of death and recovery is also increasing as time passes. Thankfully, the increment of the death rate is not as fast as the increment rate for both confirmed and recovered cases. However, there are already 233,388 patients lost their lives. It is not we want to happen during the fight with the novel coronavirus.

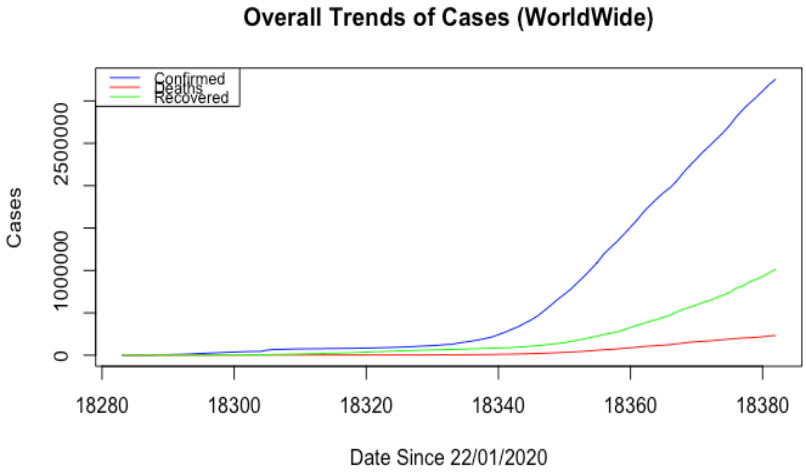

Figure 2: line plot for the overall trends

3.3. Association between Confirmed and Death Cases

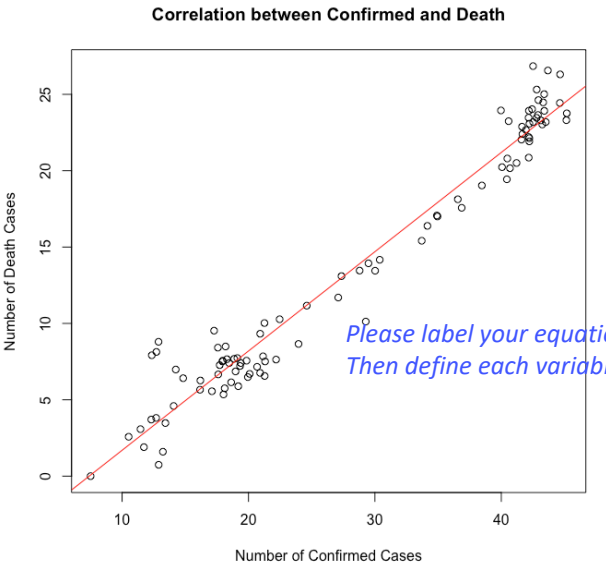

Figure 3: scatterplot with SLR line

As we may notice that even the rate of increment of the death cases is not very significant, the rise in the confirmed cases is correlated with the death cases. We attempted to use a simple linear regression (SLR) model to capture the relationship between the confirmed and death cases here. Before fitting the regression, we apply scaled-power transformation with $\lambda = 0.2$ to approximate a linear relationship among types of cases[2].

$$E(Number\ of\ Death_i) = \beta_0 + \beta_1 \times Number\ of\ Confirmed_i$$

Please label your equations , e.g. Eqn (1), and use proper variables. Then define each variable clearly.

Figure 3 shows a positive association between the confirmed and death cases. We may interrupt such a phenomenon that the number of deaths cases increase as the growth in the number of confirmed cases. From figure 2, we have already known that the number of confirmed is increasing exponentially, that means the novel coronavirus is still a threat to our lives.

Visualisation gives us a quick overview of the data as well as the current circumstance of the novel coronavirus. As we mentioned in the previous section, a lot of medical personnel work under tremendous pressure. After retrieving the data of the novel coronavirus, we see that the ongoing situation is not very pleasant. For those like us, without any knowledge of professional medicine, we are not able to take off their shoulders. Hence, we try our best to analysis the historical data and provide some suggestions to doctors to relieve their pressure in the view of making decisions.

burden

[2] Details of scaled-power transformation are omitted since it is not the focus on the project, as well as not a statistic project.

4. Supervised Learning — Decision Tree

To advice the professionals, we are going to classify confirmed patients with either a high death rate or a low death rate using supervised learning – decision tree. A decision tree is a tree-like graph to classify inputs with different labels (Brid, 2018). The tree stores split nodes and leaf nodes. Each split node consists of a test function for the incoming data. The leaf node of the model stores the final class (Criminisi et al., 2011). Figure 4 illustrates the general ideas on how the decision tree works.

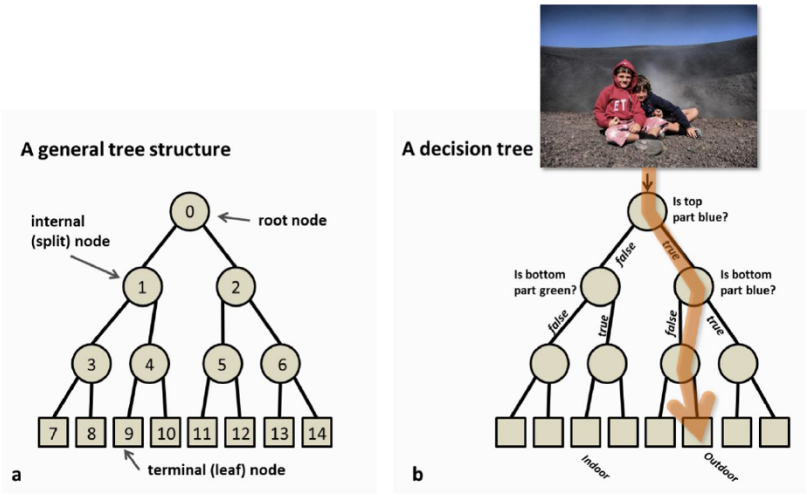

Figure 4: (a) layout for Decision Tree (b) and example

To understand the likelihood of death for a confirmed patient, we define “dead” and “alive” as the class labels of the decision model. In other words, one is classified with a higher death rate if he or she was being labelled as “dead”. For those being labelled as “alive”, they are considered with a lower chance to lose their lives. During the process of constructing the decision tree, “Gender”, “Age (>65)”, “Travel”, “Fever”, “Cough”, and “Chronic disease” are being adopted as the attributes to identify the destiny of a confirmed patient. The data type of the above variables[3] is binary, either “True” or “False”. The reason for setting the age group as “above 65 years old” based on the definition of the WHO. A person equals to or below 65 years old is still at least an adolescent (Victoria, 2018). We believe that the functionality of the body is deteriorating after the age of 65. Hence, such an age division would help us to get a clear separation to the ability to fight with the novel coronavirus.

4.1. Data Cleaning for Decision Tree *input*

Before categorising any tags, we inputted the raw data into the Integrated Development Environment – Google Colab. However, the data are not well organised. We cannot directly use the given set of data to generate the decision tree model. Therefore, we converted it into numeric type (see Figure 5 and 6), such that the dataset would be more favourable to establish the decision tree model.

	Age (>65)	Gender	Travel	Fever	Cough	Chronic Disease	Dead
0	False	male	False	False	False	False	dead
1	True	male	True	False	False	True	dead
2	False	male	False	False	False	False	dead
3	False	male	True	False	False	False	dead
4	False	female	True	False	False	False	alive

Figure 5: partial data set before transformation

	Age (>65)_n	Gender_n	Travel_n	Fever_n	Cough_n	Chronic Disease_n
0	0	1	0	0	0	0
1	1	1	1	0	0	1
2	0	1	0	0	0	0
3	0	1	1	0	0	0
4	0	0	1	0	0	0

Figure 6: partial data set after numerical transformation

[3] “Gender” classifies whether the patients are male; “Age range” classifies whether the patients are over 65 years old; “Travel” classifies whether the patients have travelled record before their confirmed infection; “Fever” classifies whether the patients have fever; “Cough” classifies whether the patients have cough; “Chronic disease” classifies whether the patients have chronic disease

4.2. Splitting Dataset

To evaluate the performance of the model, we shuffled the dataset and employed 70% of the data for training the model. The remaining data are for the testing of the model. Figure 7 shows partial training dataset.

Good.

	Age (>65)_n	Gender_n	Travel_n	Fever_n	Cough_n	Chronic Disease_n
173	1	1	1	0	0	1
101	1	1	1	0	0	0
234	0	1	1	0	1	0
144	0	1	1	0	0	1
96	0	0	1	0	0	0
...	...	...	...	...	...	...
20	0	0	1	0	0	0
188	1	1	1	0	0	1
71	0	1	1	1	0	0
106	0	1	1	0	0	0
102	1	0	1	0	0	1

186 rows × 6 columns

Figure 7: dataset for training after shuffling

4.3. Splitting Nodes

Since there are multiple attributes in the dataset, the order of nodes becomes a concern to the decision tree. The Gini index provides an appropriate splitting method based on the degree of node impurity. For those nodes with a lower degree of impurity, they will have a lower value in the Gini index and is more preferred to place at the top of the decision tree. The following equations show the necessary information for calculating the Gini index (see Equation 1 and 2).

For a given node t, where $(j|t)$ is the relative frequency of class j at node t:

$$GINI(t) = 1 - \sum_{j} [p(j|t)]^2$$

(Equation 1)

After the computation for each node, Equation 2 determines the preference of the split nodes (attributes). Noted that n_i is the number of records at child node i, n is the number records at node t.

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

(Equation 2)

4.4. Model Training and Visualisation

After the calculation, it is all set to build up the decision tree model using training data. As mentioned previously, we utilise 70% of the raw data to train up the model. The following is the fitted model (see Figure 8). The decision tree is too large to display in detail. Thus, we broke the image into pieces (see Appendixes 9.8).

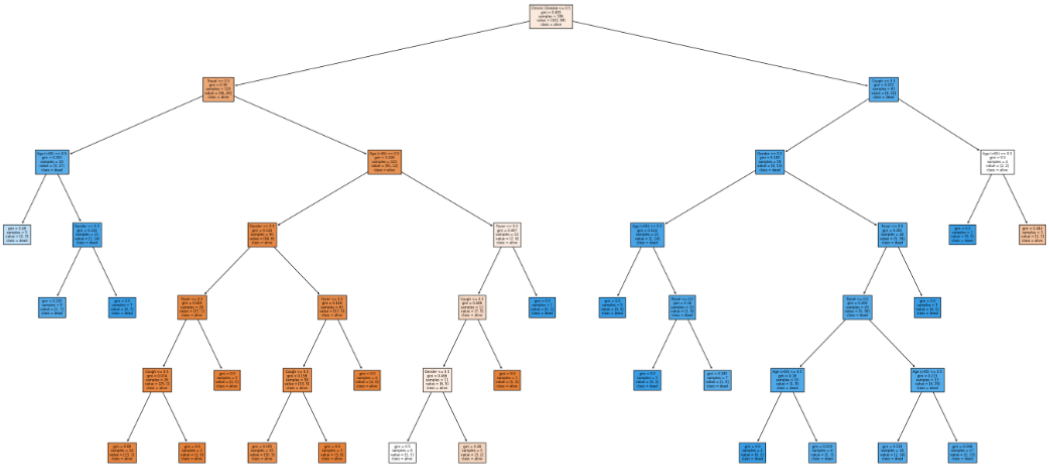

Figure 8: decision tree trained with training dataset

For your interest: we also established another decision tree model trained with the full dataset to compare with our desired model. In the coding, "model" refers to the model with the full dataset, whereas "model_training" refers to the model trained with the training dataset.

4.5. Model Evaluation

After constructing the decision tree model with training dataset, the remaining 30% of the raw data are used to evaluate the performance of the model. Besides, the scores of Precision, Recall and the F-measure are common approaches to weigh the efficiency. Their formulas are as follow:

$$Precision(p) = \frac{true\ positive}{true\ positive + false\ positive}$$

$$Recall(r) = \frac{true\ positive}{true\ positive + false\ negative}$$

$$F - measure(F) = \frac{2rp}{r+p} = \frac{2 \times true\ positive}{2 \times true\ positive + false\ negative + false\ positive}$$

From the outputs of coding in 9.7 Appendixes, the scores of Precision, Recall, and the F-measure is 0.7895, 0.9091, and 0.8451 respectively. By the definitions of the above equations, values close to 1 are known as effective since the error terms (false decision) are small. Owning the above quantities, the decision model is quite effective in predicting the new source of data. However, we figured out that there are some attributes that are unnecessary for the decision tree. For example, if the values of an observation on "Chronic Disease" (root node) is 1 and "Cough" (the split node in the next layer) is 0, the observation is classified as "dead" whatever remaining values of the split nodes. Then, the remaining attributes seem to be redundant. Here concludes that we may discard some attributes in further studies if any.

We wish the decision tree would give speedy guidance about the destiny of a confirmed patient. With such information, we believe the classification, either "alive" or "dead", would give the professionals some hints on allocating the resources on the urgent patients.

5. Unsupervised Learning — k-means Method

To infer the urgency of a confirmed patient, k-means are an affirmative technique to calculate the Euclidean distances between each patient. We focus on the historical data for those patients who have passed their lives. We believe that gender is a critical factor that would result in different immune responses to different infectious diseases (Sabra L. & Katie L., 2016). Hence, we perform the k-means method twice for both male and female to understand if sex would affect the immune responses to novel coronavirus.

5.1. Estimating the Optimal Number of Clusters

Before carrying out the k-means method, we estimated the optimal number of clusters in male and female using the Elbow Method. Recall the objective function of k-mean is

$$min_S \sum_{i=1}^{K} \sum_{x \epsilon S_i} \|\vec{x} - \vec{\mu_i}\|^2$$

Given a set of objects $(\vec{x_1}, \vec{x_2}, \dots, \vec{x_n})$, where each object is a d-dimensional real vector, we partitioned the n objects into k sets, such that $S = \{S_1, S_2, \dots, S_K\}$ and $\vec{\mu_i}$ is the mean of objects in S_i. These functions measure the compactness of the cluster. We attempt to minimise objective function to obtain the optimal number of clusters using R software. The results show that 3 is the optimal number of clusters as an elbow appears at $k = 3$. Figure 9 and 10 show the results of the Elbow Method in male and female respectively.

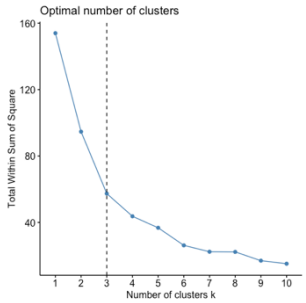

Figure 9: result of the Elbow Method in male

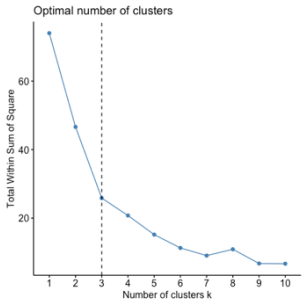

Figure 10: result of the Elbow Method in female

5.2. Data Transformation

The dataset for males consists of 78 observations with attributes of "age" and "time to death" (difference in confirmed date and death date). The dataset for females consists of the same attributes with 38 observations. Since the variability of "age" is sufficiently larger than that of "time to death", we adopted z-score standardisation to remedial the issue of huge variability. Figure 11 and 12 show part of the changes in the data after standardisation in male and female respectively.

Age <dbl>	Time to Death <dbl>
63	1
70	4
56	2
30	2
65	0
85	5

```
            Age Time to Death
[1,] -0.2729313    -0.8842778
[2,]  0.2391666     0.4681471
[3,] -0.7850293    -0.4334695
[4,] -2.6871075    -0.4334695
[5,] -0.1266176    -1.3350860
[6,]  1.3365194     0.9189553
```

Figure 11: change of the data after standardisation in male

Age <dbl>	Time to Death <dbl>
85	0
95	1
79	1
82	10
59	5
82	13

```
            Age Time to Death
[1,]  0.8852304    -1.0338540
[2,]  1.8746056    -0.7339574
[3,]  0.2916053    -0.7339574
[4,]  0.5884179     1.9651117
[5,] -1.6871450     0.4656289
[6,]  0.5884179     2.8648014
```

Figure 12: change of the data after standardisation in female

5.3. k-means Method

After data transformation, we initiated the k-means method by setting seed points randomly for partitioning. The seed points are the centroids at the beginning. We compute Euclidean distance between all objects and each centroid respectively. Recap the formula of Euclidean distance is

$$d(\vec{x}_i, \vec{\mu}_j) = \sqrt{\sum_{z=1}^{p}(x_z - \mu_z)^2}, \text{ where } p \text{ is the number of attributes}$$

While letting $k = m$, where m is the optimal number of clusters and $n \in [m, \infty)$ is the number of objects, we gain $\vec{\mu}_i = \vec{x}_i$ where $i \in [1, m]$ for initiation. There are enough information to construct a data matrix.

$$D = \begin{pmatrix} d(\vec{x}_1, \vec{\mu}_1) & \cdots & d(\vec{x}_n, \vec{\mu}_1) \\ \vdots & \ddots & \vdots \\ d(\vec{x}_1, \vec{\mu}_m) & \cdots & d(\vec{x}_n, \vec{\mu}_m) \end{pmatrix}$$

We compare the data matrix by column and assign the smallest distance in each column to the respective cluster. Then we compute the new centroids in each cluster $\vec{\mu}'_j = \frac{1}{|c|}\left(\sum_{k=1}^{[c]} \vec{x}\right)$, where c is the number of objects in cluster j. We repeat updating the data matrix with the new centroids and perform comparisons by column to compute new centroids until $\vec{\mu}'_j = \vec{\mu}_j$. The computation is done, and the following result (Figure 13 and 14) show that we used K-means Method for male and female respectively.

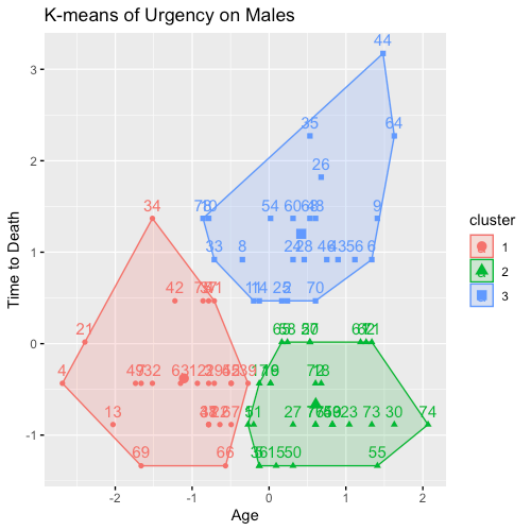

Figure 13: K-means of males

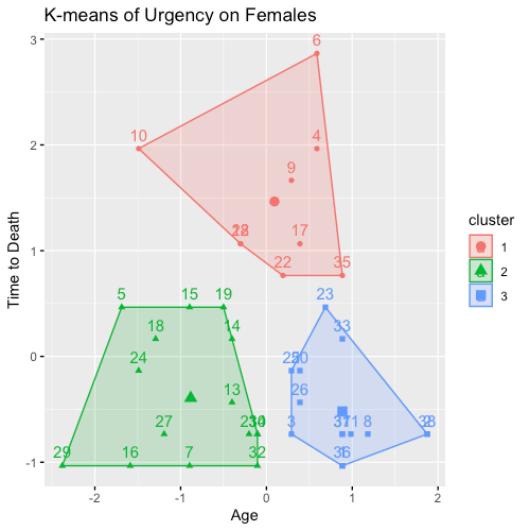

Figure 14: K-means of females

Analysing Results: for males, the rank of urgency coloured in **Green > Red > Blue**. The cluster in green has the lowest centroid, meaning patients in the cluster have the shortest time to death. For females, the order of urgency is **Blue > Green > Red**. Patients in the blue cluster have the shortest time to death. The results show that both groups have the same order of urgency so gender is not a dominant factor in the novel coronavirus.

Application of the Results: We used "Time to Death" attribute to estimate the time of a confirmed patient turned into a severe case that is likely to die. We then can determine how likely the patient would turn into a severe case by knowing his or her age and confirmed date using the k-means result. For example, if a 55 years old female confirmed the novel coronavirus for 4 days, we can quickly identify that she falls into the green cluster (normal risk) using the Euclidean distance. She is less critical to have an immediate medical need and could hold on a while for medical care. By reallocating the medical resources, doctors would spend more effort on those with higher urgency. The use of k-means helps the experts to make a decision quickly.

6. Conclusion

In reviewing the recent situation of the novel coronavirus, the overall trend does not show any good news from figure 2. The number of confirmed and death is still increasing at the moment. It is not an easy task to make things under control, where Donald Trump plays as a leader in the fight of the novel coronavirus over the world. That is why "Trump" is the keywords being so popular in the streaming data related to the novel coronavirus. The number of deaths is the main concern of the project. We understand that the association between the confirmed and death cases are positively correlated, that is an unit increase in confirmed cases would lead to an increment in the number of deaths. Having such a fact from figure 3, we want to relieve the stress on medical personnel in helping them in the view of decision making. We established the decision tree model and k-means model to classify whether a confirmed patient has a higher risk.

The performance of the decision tree model is quite well. The model trained with 70% of the raw data and tested with the remaining data for the accuracy. The score of Precision, Recall, and F-measure are 0.7895, 0.9091, and 0.8451 respectively. Owing to the above quantities, the model is significant to predict instances and a new source of data. However, we noticed that there are some unnecessary attributes in the model. If, for instance, one has a chronic disease and has cough before confirmation, he or she would be classified as "dead" no matter what other characteristics one has. Simply put, other attributes are redundant under the given set of data. For sure a better decision model may be investigated in the future while more data are available.

The outcome of the k-means method shows that gender is not a dominant factor in defending the novel coronavirus. Older patients with a shorter confirmed date are classified as urgent. Younger patients with a shorter confirmed date are classified as at normal risk. For those with a longer confirmed date are labelled a lower risk. We await the medical staff can make use to our results to detect severe cases efficiently in order to have immediate medical support. Because of the novel coronavirus, some of us have already lost their works, their relationships, and even their lives. Here we want to make a kindly remind all of us to stay safe. It is a hard time for all of us, and we believe we can pass it through the difficulties together.

7. Reference

Brid, R. S. (2018). Decision Trees - A simple way to visualize a decision. Medium. Retrieved from https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb

Criminisi, A., Shotton, J. & Konukoglu, E. (2011). Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Microsoft Research Ltd. Retrieved from https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/decisionForests_MSR_TR_2011_114.pdf

Sabra L., Flanagan L. (2016) Sex differences in immune responses. Nature Reviews Immunology. Retrieved from https://doi.org/10.1038/nri.2016.90

SRK (2020) Novel Corona Virus 2019 Dataset. Kaggle. https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset

Victoria T. (2018) 65 Years old is still young!. Indigo. Retrieved from https://www.indigo.com.hk/single-post/2018/03/13/65-Years-old-is-still-young

8. Workload Distribution

Visualisation:	King Yeung, CHAN	1155119394
Supervised learning (decision tree):	Yun Ki, LEUNG	1155109801
Unsupervised learning (k-means clustering):	Tsz Chun, LAI	1155125208

9. Appendixes

9.1. R Code for Figure 1

```r
## import libraries
library(rtweet)
library(httpuv)

# personal information omitted here
name <- "omitted"
key <- "omitted"
secret <- "omitted"

## create token named "twitter_token"

twitter_token <- create_token(app = name, consumer_key = key, consumer_secret = secret)

## keywords for filtering tweets
q <- "the novel coronavirus, covid 19, coronavirus"

# streaming time in second
streamtime <- 60

# retrieve and save the tweets
filename <- "/Users/jackchan/Downloads/tweets.json"
rt <- stream_tweets(q = q, timeout = streamtime, file_name = filename,token=twitter_token)

# clean up the tweets
clean_tweet = rt$text
clean_tweet = gsub("&amp", "", clean_tweet)
clean_tweet = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", clean_tweet)
clean_tweet = gsub("@\\w+", "", clean_tweet)
clean_tweet = gsub("[[:punct:]]", "", clean_tweet)
clean_tweet = gsub("[[:digit:]]", "", clean_tweet)
clean_tweet = gsub("http\\w+", "", clean_tweet)
clean_tweet = gsub("[ \t]{2,}", "", clean_tweet)
clean_tweet = gsub("^\\s+|\\s+$", "", clean_tweet)
clean_tweet = gsub("[^[:graph:]]", " ", clean_tweet)
clean_tweet = trimws(clean_tweet,"both")
clean_tweet = iconv(clean_tweet, "latin1", "ASCII")
clean_tweet = clean_tweet[!is.na(clean_tweet)]
clean_tweet = clean_tweet[clean_tweet!=""]

# save the tweet to a csv file
write.csv(clean_tweet,"/Users/jackchan/Downloads/Tweets.csv",row.names=FALSE)
```

9.2. R Code for Figure 2

```r
setwd("/Users/jackchan/Downloads")
data <- read.csv("Novel Coronavirus Dataset for Visualization.csv", header = TRUE)

Date <- as.Date(data$ObservationDate, format = "%m/%d/%Y")
data <- cbind(data, Date)
data <- data[, c(9, 3:4, 6:8)]

tempCum <- data.frame(as.character(data$Date), data$Confirmed, data$Deaths, data$Recovered)
names(tempCum) <- c("Date", "Confirmed", "Deaths", "Recovered")

cumulativeConfirmed <- aggregate(tempCum$Confirmed, by = list(tempCum$Date), sum)
names(cumulativeConfirmed) <- c("Date", "Confirmed")
cumulativeDeaths <- aggregate(tempCum$Deaths, by = list(tempCum$Date), sum)
names(cumulativeDeaths) <- c("Date", "Deaths")
cumulativeRecovered <- aggregate(tempCum$Recovered, by = list(tempCum$Date), sum)
names(cumulativeRecovered) <- c("Date", "Recovered")
cumulativeCases <- cbind(cumulativeConfirmed, cumulativeDeaths$Deaths, cumulativeRecovered$Recovered)
names(cumulativeCases) <- c("Date", "Confirmed", "Deaths", "Recovered")

confirmedPerDay <- c()
deathsPerDay <- c()
recoveredPerDay <- c()
confirmedPerDay[1] <- cumulativeConfirmed$Confirmed[1]
deathsPerDay[1] <- cumulativeDeaths$Deaths[1]
recoveredPerDay[1] <- cumulativeRecovered$Recovered[1]
perDayData <- data.frame(confirmedPerDay, deathsPerDay, recoveredPerDay)

for(i in 2:length(cumulativeConfirmed$Date)) {
  perDayData[i, 1] <- cumulativeConfirmed$Confirmed[i] - cumulativeConfirmed$Confirmed[i - 1]
  perDayData[i, 2] <- cumulativeDeaths$Deaths[i] - cumulativeDeaths$Deaths[i - 1]
  perDayData[i, 3] <- cumulativeRecovered$Recovered[i] - cumulativeRecovered$Recovered[i - 1]
}

# graph 1
matplot(as.Date(cumulativeConfirmed$Date), cumulativeCases[,-1], xlab = "Date Since 22/01/2020", ylab = "Cases", type = "l", col = c("blue", "red", "green"), lty = 1, main = "Overall Trends of Cases (WorldWide)")
legend("topleft", legend = c("Confirmed", "Deaths", "Recovered"), col = c("blue", "red", "green"), lty = 1, cex = 0.8)
```

9.3. R Code for Figure 3

```r
# graph 2
y_lambda0.2 <- perDayData[, 2]
x_lambda0.2 <- perDayData[, 1]

for(i in 1:length(perDayData[, 2])) {
  if(perDayData[i, 2] == 0){
    y_lambda0.2[i] <- log(perDayData[i, 2])
    x_lambda0.2[i] <- log(perDayData[i, 1])
  }
  else{
    y_lambda0.2[i] <- (y_lambda0.2[i] ^ 0.2 - 1) / 0.2
    x_lambda0.2[i] <- (x_lambda0.2[i] ^ 0.2 - 1) / 0.2
  }
}

plot(x_lambda0.2, y_lambda0.2, xlab = "Number of Confirmed Cases", ylab = "Number of Death Cases", main = "Correlation between Confirmed and Death")
reg <- lm(y_lambda0.2 ~ x_lambda0.2)
abline(reg, col = "red")
```

9.4. Python code for Figure 5 and 6

```
[ ] import pandas as pd
    df = pd.read_csv("/content/Original data (1).csv")
    df.head()

    from sklearn.preprocessing import LabelEncoder
    le_Dead = LabelEncoder()
    inputs = df
    inputs["Dead_n"] = le_Dead.fit_transform(inputs['Dead'])
    targets = inputs['Dead_n']
    inputs = inputs.drop(['Dead_n','Dead'],axis='columns')
    inputs['Age (>65)_n'] = le_Dead.fit_transform(inputs['Age (>65)'])
    inputs['Gender_n'] = le_Dead.fit_transform(inputs['Gender'])
    inputs['Travel_n'] = le_Dead.fit_transform(inputs['Travel'])
    inputs['Fever_n'] = le_Dead.fit_transform(inputs['Fever'])
    inputs['Cough_n'] = le_Dead.fit_transform(inputs['Cough'])
    inputs['Chronic Disease_n'] = le_Dead.fit_transform(inputs['Chronic Disease'])
    inputs = inputs.drop(['Age (>65)','Gender','Travel','Fever','Cough','Chronic Disease'],axis='columns')
    inputs.head()
```

9.5. Python code for Figure 7

```
[ ] from sklearn import model_selection
    # split into 70 training 30 testing
    X_train, X_test, y_train, y_test = model_selection.train_test_split(inputs, targets, test_size=0.3, random_state=42)
```

9.6. Python code for Figure 8

```
[ ] from sklearn import tree
    model = tree.DecisionTreeClassifier()
    model.fit(inputs,targets)
    model_training = tree.DecisionTreeClassifier()
    model_training.fit(X_train,y_train)
    # make two model, one for splitting data, one for whole (no split)

    # for not splitting the data
    import matplotlib.pyplot as plt
    fig, ax = plt.subplots(figsize=(50, 24))
    fn = ['Age (>65)','Gender','Travel','Fever','Cough','Chronic Disease']
    cn = ['alive','dead']
    tree.plot_tree(model,
                   fontsize=6,
                   feature_names = fn,
                   class_names = cn,
                   filled = True);
    plt.savefig('tree_whole', dpi=100)

    # for splitting the data
    fig, ax = plt.subplots(figsize=(50, 24))
    tree.plot_tree(model_training,
                   fontsize=6,
                   feature_names = fn,
                   class_names = cn,
                   filled = True);
    plt.savefig('tree_split', dpi=100)
```

9.7. Python code for the Evaluation of the Decision Tree

```
[ ] # after splitting the data, see how well the model works
    from sklearn.metrics import precision_score
    from sklearn.metrics import recall_score
    from sklearn.metrics import f1_score
    testing_result = model_training.predict(X_test)
    recall_score(y_test,testing_result)

    precision_score(y_test,testing_result)

    f1_score(y_test,testing_result)
```

9.8. Figures for the Decision Tree

Note: For the left-hand side of the node, it represents that the condition of the node is true, and vice versa. For example, in Figure 9.8.1 below, if the value of an instance on Chronic Disease is 0, which means that the instance does not have any chronic disease, it falls into the left-hand side of the split.

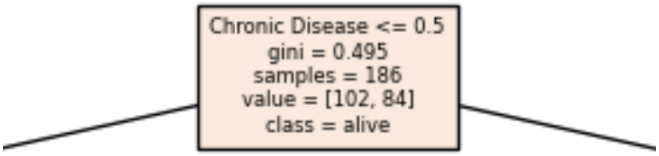

Figure 9.8.1: root node

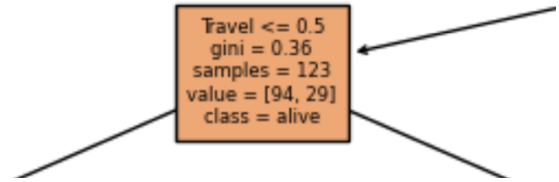

Figure 9.8.2: node after the root node if Chronic Disease is smaller than or equal to 0.5

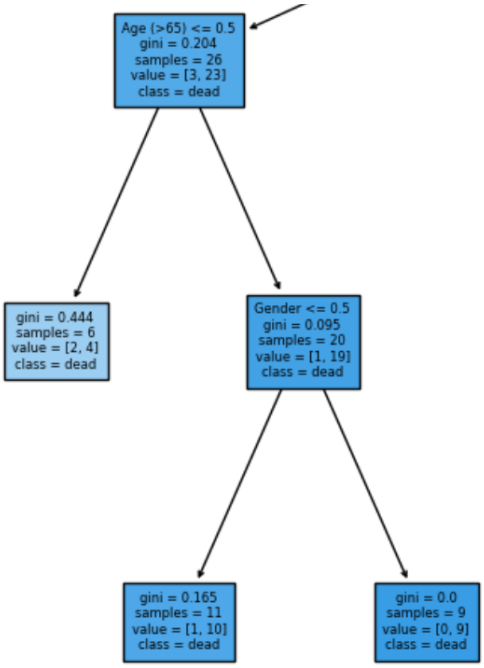

Figure 9.8.3: if the value of an instance on the node in Figure 9.8.2 is smaller than or equal to 0.5

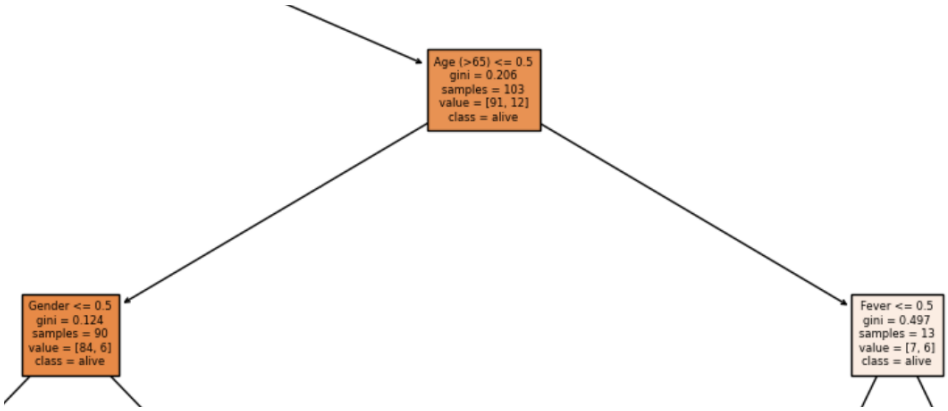

Figure 9.8.4: if the value of an instance on the node in Figure 9.8.2 is greater than 0.5

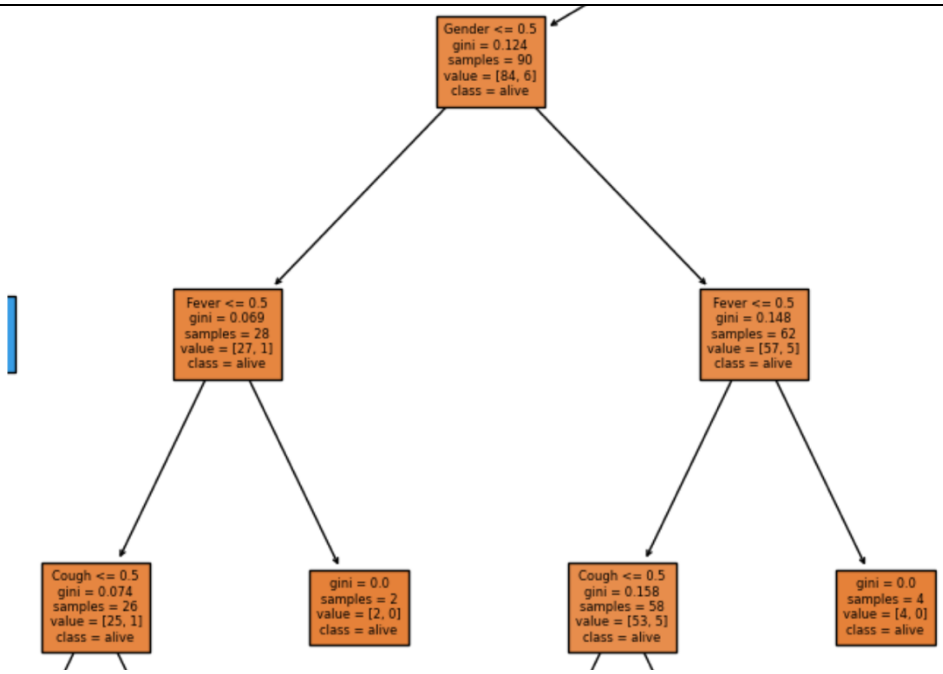

Figure 9.8.5: if the value of "Age (>65)" in Figure 9.8.4 is smaller than or equal to 0.5

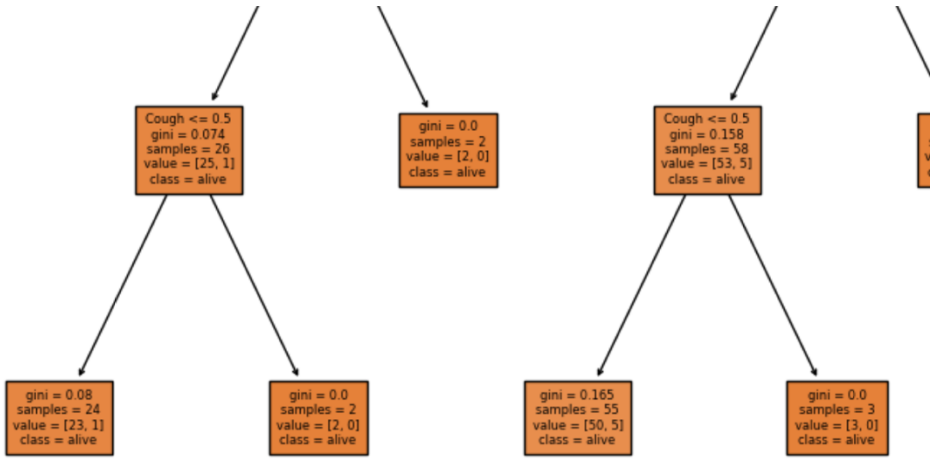

Figure 9.8.6: the remaining part for Figure 9.8.5

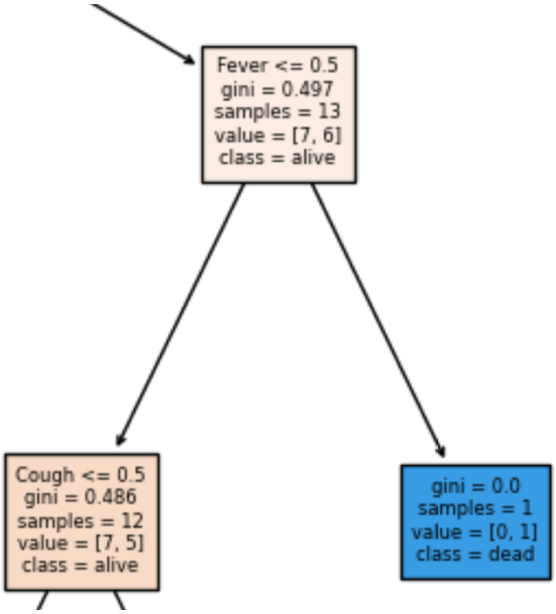

Figure 9.8.7: if the value of "Age (>65)" in Figure 9.8.4 is greater than 0.5

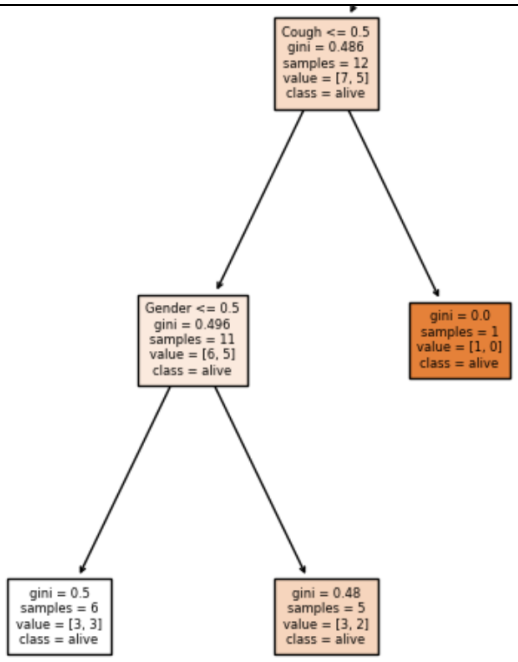

Figure 9.8.8: the remaining part for Figure 9.8.7

The above figures show all the split nodes that will be used if the value of the root node is smaller than or equal to 0.5. The following figures will show the remaining of the model.

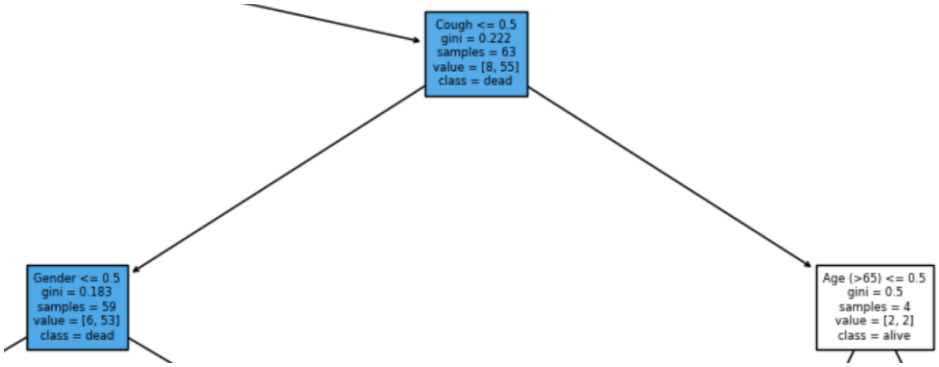

Figure 9.8.9: node after the root node if Chronic Disease is greater than 0.5

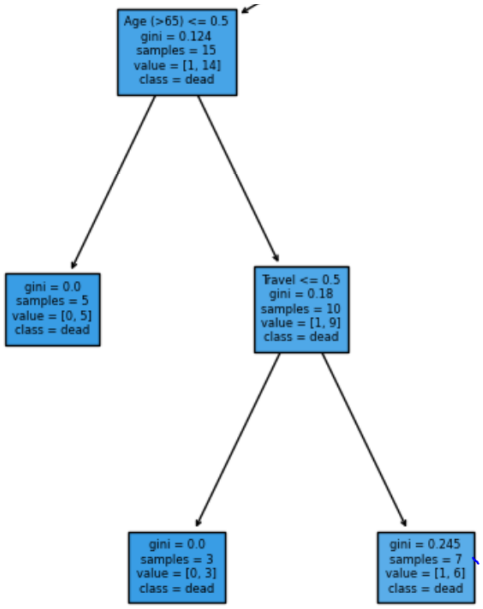

Figure 9.8.10: if the value of "Gender" in Figure 9.8.8 is smaller than or equal to 0.5

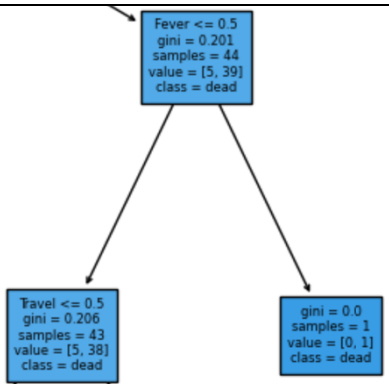

Figure 9.8.11: if the value of "Gender" in Figure 9.8.8 is greater than 0.5

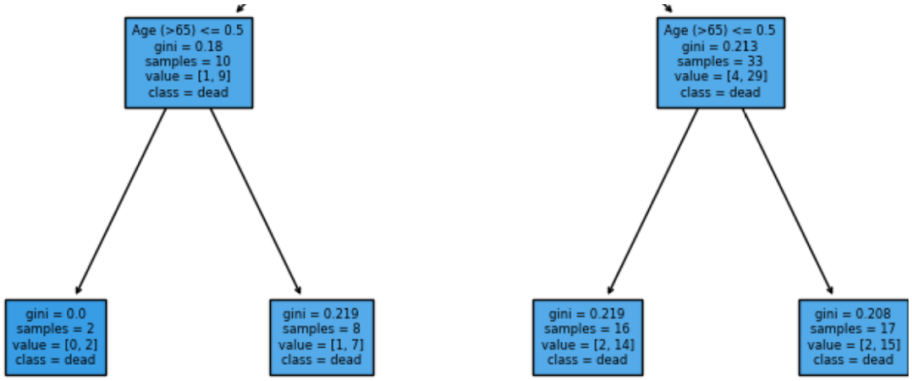

Figure 9.8.12: the remaining part of Figure 9.8.10

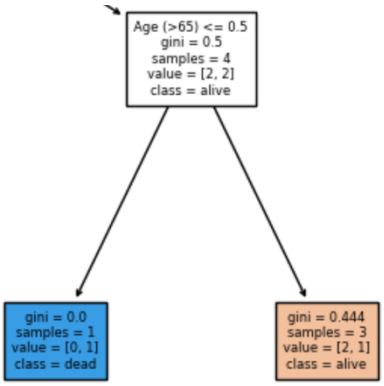

Figure 9.8.13: if the value of "Cough" in Figure 9.8.8 is greater than 0.5

9.9. R code for Unsupervised Learning

```
library(readxl)
library(factoextra)
#Loading required library
kmeansmale <- read_excel("kmeansmale.xlsx")
kmeansfemale <- read_excel("kmeansfemale.xlsx")
#Loading required dataset
stmale<-scale(kmeansmale) #Standardize the dataset
fviz_nbclust(stmale,kmeans,method = "wss")+geom_vline(xintercept = 3, linetype = 2)
set.seed(123) #Find the optimal number of clusters for male
km_res_stmale<-kmeans(stmale,3) #Use the K-means method for the dataset
km_res_stmale[["centers"]] #Show the centroids of K-means
plot_male<-fviz_cluster(km_res_stmale,stmale,main="K-means of Urgency on Males") #Plot K-means result
plot_male

stfemale<-scale(kmeansfemale) #Standardize the dataset
fviz_nbclust(stfemale,kmeans,method = "wss")+geom_vline(xintercept = 3, linetype = 2)
#Find the optimal number of clusters for female

set.seed(1234)
km_res_stfemale<-kmeans(stfemale,3) #Use the K-means method for the dataset
km_res_stfemale[["centers"]] #Show the centroids of K-means
plot_female<-fviz_cluster(km_res_stfemale,stfemale,main="K-means of Urgency on Females") #Plot K-means result
plot_female
```