Iteration 4 – BDAS

Data mining in a coronavirus dataset

Johnson Zhou
University of Auckland, New Zealand

STUDENT ID: 8279325
Name: Johnson Zhou
UPI: zzho612

1 Business/Situation Understanding

1.1 Identify the objectives of the situation

In 2015, world leaders reached a consensus on 17 global goals (formally called the Global Goals for Sustainable Development or SDGs). By eradicating poverty, eliminating inequality, improving the environment and climate, and more, these goals ultimately aim to create a better world by 2030. Guided by these goals, now all of us, governments, businesses, civil society and the public should work together to build a better future for everyone.

The SDGs recognize the interdependence between health and development, provide an ambitious and comprehensive plan of action for humanity, the planet, and prosperity, and eliminate injustices that underpin bad health and development outcomes. Promoting health and well-being is one of the 17 global goals that make up the 2030 Agenda for Sustainable Development (Goal 3). Goal 3 from SDGs committed to ensuring the health and well-being of everyone, it also aims to achieve universal health coverage and provide safe and effective medicines and vaccines for all.

However, the emergence of a novel coronavirus (nCoV) affected this goal in early 2020. Coronaviruses (CoV) are a large group of viruses that cause illnesses ranging from the common cold to more severe illnesses such as the Middle East Respiratory Syndrome (MERS-CoV) and Severe Acute Respiratory Syndrome (SARS-CoV). The nCoV is a new strain that was not discovered in humans until late 2019 to early 2020. The disease caused by the nCoV was first discovered in Wuhan, Hubei, China in December 2019, and spread to each provincial administrative region of the country on January 29, 2020. On January 30, the World Health Organization declared the disease a public health emergency of international concern, followed by an increase in the number of cases outside China. As of March 9, 2020, 119,000 cases of the disease have been confirmed in more than 110 countries and regions. Among them, mainland China (81,000 cases, about 70% of the total cases) has had pandemics in Italy, South Korea and Iran. More than 66,000 people have recovered and more than 4,200 have died (3,200 in China, about 75% of the total deaths). Coronaviruses are zoonotic viruses, meaning they spread between animals and humans. A detailed investigation found that SARS-CoV was transmitted from cats to humans, while MERS-CoV was transmitted from humans to unimodal camels. Several known coronaviruses are spreading in animals that have not yet infected humans. Common signs of infection include respiratory symptoms, fever, cough, shortness of breath, and dyspnea. In more severe cases, the infection can lead to pneumonia, severe acute respiratory syndrome, kidney failure and even death.

Because of the huge impact of the current outbreak of the nCoV, major governments and organizations including the World Health Organization have taken actions. Public health responses around the world have included travel restrictions, quarantines, curfews, and school closures. They have included the quarantine of all of Italy and the Chinese province of Hubei; various curfew measures in China and South Korea; screening methods at airports and train stations; and travel advisories regarding regions with community transmission. Schools have closed nationwide or locally in some countries, affecting more than 300 million students. Also, worldwide effects of the outbreak of the new coronavirus include social and economic instability, xenophobia and racism against people of Chinese and East Asian descent, and the online spread of misinformation and conspiracy theories about the virus.

New Zealand, a country that is currently relatively less affected, is working to protect the people of New Zealand from this new global coronavirus outbreak. However, we cannot relax because the current situation is still not optimistic. If advanced data analysis techniques can be used to predict the future of New Zealand, it will definitely help New Zealand better cope with this new coronavirus outbreak. In this study, I will focus on finding the relationship between mortality and age of people with new coronavirus infections to help people better allocate medical resources. In addition, I will focus on finding a model that could potentially predict the mortality of a patient.

In summary, data related to this new coronavirus outbreak are slowly being collected and transmitted to the Internet. Analysts hope to use the power of big data to help people better understand and respond to this new coronavirus outbreak. In this study, the following goals are proposed:

- Find the relationship between mortality and other factors in people with new coronavirus infections.
- Produce a model that have predicting power for the mortality patients of new observations.

Tentatively, the study will be judged a success if:

- Successfully determine whether the death rate of a virus infected person is related to the age group of the infected person.
- Achieve a predicting accuracy at least better than random.

1.2 Assess the situation

There hasn't been a lot of analysis produced on the new coronavirus related datasets yet. Also, the related datasets are continuously growing and very few cleaning processes are performed on these datasets. In this study, one of the main tasks is to assess the related data.

Personnel. Obviously, this research will involve very limited resources and time. I may consult with a data scientist for help as they may be able to resolve the difficulties I encountered during the study and may expand this research for future use. In this study, a basic level analysis of new coronavirus-related data will be performed and may be used by others for further research or extensions.

Data. Due to the unique nature of the new coronavirus-related data, I will not be able to create or collect more data to support my study. Depending on the circumstances, this study may involve multiple data sets. In addition, this study continues to erupt new coronaviruses, and data sources are constantly being updated. Also, the person who uploads the data set usually cleans up after the data set stops updating, so I cannot expect the quality of the original data set to be very high. Therefore, this study will involve many data pre-processing steps.

Requirements, Assumptions and Constraints.

- **Requirements:**
  The data used in this study are from public open sources, so security and legal restrictions should not be the main focus of this study. The results of this study will provide an example for analysing new coronavirus-related open-source datasets and provide readers with a better perspective and guidance on the new coronavirus outbreaks, which will help achieve the promoting good health and well-being goal.
- **Assumptions:**
  I would assume that there will not be any economic factors that might affect the study because the data and tools used in this study are open source. Any help from professionals will be gather through online channels or platforms which should not raise any economic factors to the study.

Also, the data used in this study is gathered from Kaggle, a well-known data resource that opens to the public therefore I would assume the data is reliable. However, as stated above, the dataset is still continuously updating so that I assume the quality of data will be relatively lower and needs to be pre-processed.
For the project sponsor/management team, I assume that they might focus on the result rather than to understand the models since there will be different results produced in this study and several models will be assessed.

- **Constraints:**
  Because the data comes from an open source platform and is available to the public, there is no access constraint. Anyone using the data set will refer to the original data author in order to comply with legal constraints. There are no financial restrictions on the project as both data and tools are available and open source.

Risk and Contingencies.

- **Scheduling:**
  Due to limited resources and time, the project may take longer than expected. Similarly, difficulties can arise and can be difficult to resolve. An emergency plan is to spend more time researching and trying to consult professionals through online channels.
- **Financial:**
  There should not be any financial risk in this project.
- **Data:**
  As mentioned above, data quality may be the main risk for the project. There are several new coronavirus-related datasets online, each of which may have different dimensions. This study is possible to involve multiple situation-dependent datasets. In addition, this study definitely requires good and adequate data pre-processing.
- **Results:**
  It is not surprising if the initial results were not as dramatic as expected. The theme of this research is very new, and the quality of the data can have a big impact on the results. A contingency plan will be to gather more data or adjust expectations for the study.

1.3 Determine data mining objectives

Data mining goals.

- Find the relationship between mortality and other factors in people with new coronavirus infections.
- Produce a model that have predicting power for the mortality patients of new observations.

Data mining success criteria.

- Successfully determine whether the death rate of a virus infected person is related to the age group of the infected person A proper regression model for analysing the attributes.
- Achieve a predicting accuracy at least better than random.

1.4 Produce a project plan

I will conduct this research independently, complemented by potential help from online channels. A preliminary project plan will be provided below:

Phase	Time	Resources	Risks
Business understanding (Situation assessment)	Week 1	All analysts	Economic change
Data understanding	Week 1	All analysts, Python, Spyder, PySpark	Data problems, technology problems
Data preparation	Week 2	All analysts, Python, Spyder, PySpark	Data problems, technology problems
Modelling	Week 3	All analysts, Python, Spyder, PySpark	Technology problems, inability to find adequate model
Evaluation	Week 3	All analysts, Python, Spyder, PySpark	Economic change, inability to implement results
Repeat iterations	Week 4	All analysts, Python, Spyder, PySpark	Economic change, inability to implement results
Reporting	Week 4	All analysts	Economic change, inability to implement results

Table 1 -- The project plan table

(IBM SPSS Modeler CRISP-DM Guide)

And a Gantt Chart for the project plan:

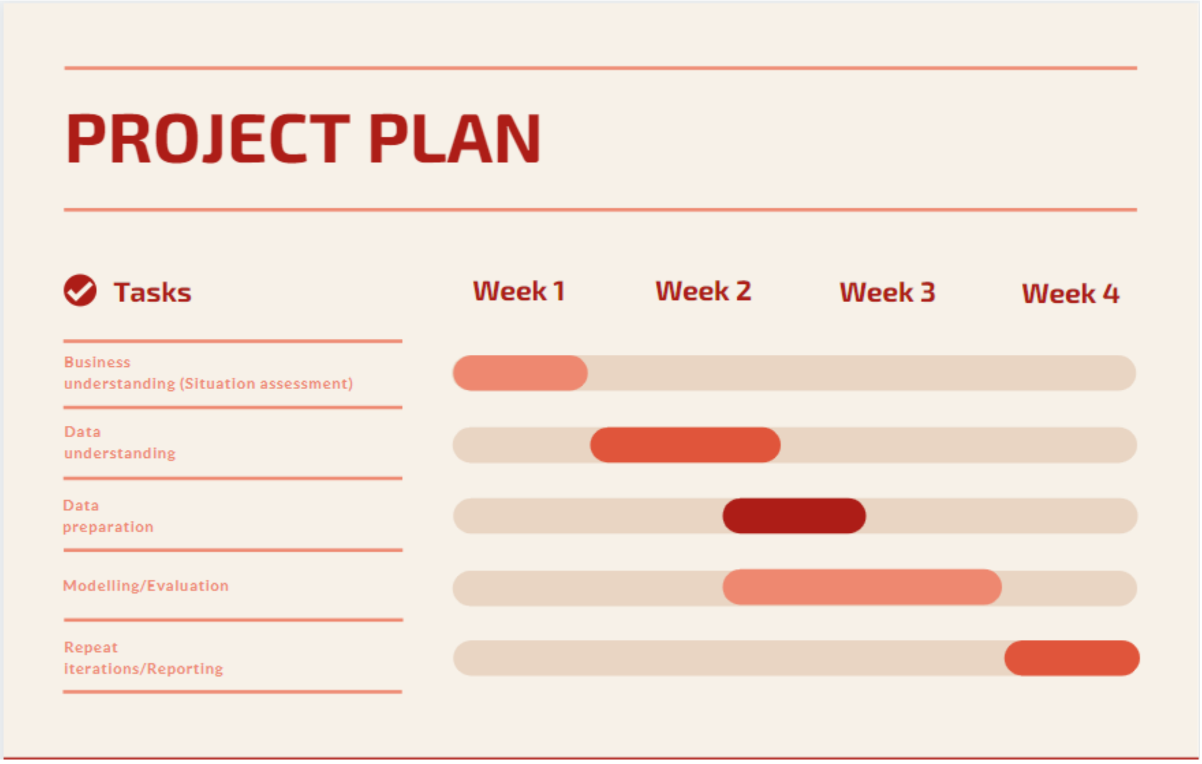


Figure 1 -- The project plan Gantt Chart

STUDENT ID: 8279325
Name: Johnson Zhou
UPI: zzho612

2 Data Understanding

2.1 Collect initial data

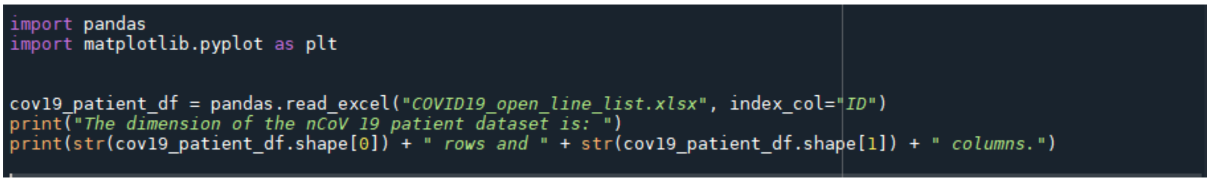
The data used in this study was downloaded from a data scientist at Kaggle named SRK on March 28, 2020 (https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset). Data was collected and summarized from the data repository for the 2019 Novel Coronavirus Visual Dashboard operated by the Johns Hopkins University Center for Systems Science and Engineering (https://github.com/CSSEGISandData/COVID-19). Depending on the circumstances, this study may involve other datasets.

A quick glance at the data set clearly shows that some data is missing in some attributes, such as age or gender. This may be human errors because the data may be collected manually, or some patients may choose to refuse or fail to answer (dyspnea is one of the symptoms of the new coronavirus infection). In this study, I will treat these missing data as miss complete at random. Since I am not doing in-depth geographic model analysis here, certain attributes may be less useful for this study, such as latitude and longitude.

At this stage, it may be too early to determine whether there is sufficient data to reach general conclusions or make accurate predictions. In addition, as new coronavirus outbreaks are still ongoing, this data set will be continuously updated. As I mentioned above, other datasets may be involved later, and I may consider merging these datasets into my original dataset.

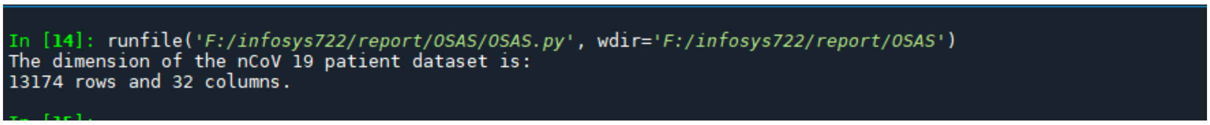
2.2 Describe the data

Amount of data. There are multiple data files of the original dataset used in this study. The data I will use to find the relationship between patients and age has over 13000 observations and 33 variables.



```
import pandas
import matplotlib.pyplot as plt


cov19_patient_df = pandas.read_excel("COVID19_open_line_list.xlsx", index_col="ID")
print("The dimension of the nCoV 19 patient dataset is: ")
print(str(cov19_patient_df.shape[0]) + " rows and " + str(cov19_patient_df.shape[1]) + " columns.")
```

Figure 2 – The initial screenshot from Python for data inspection

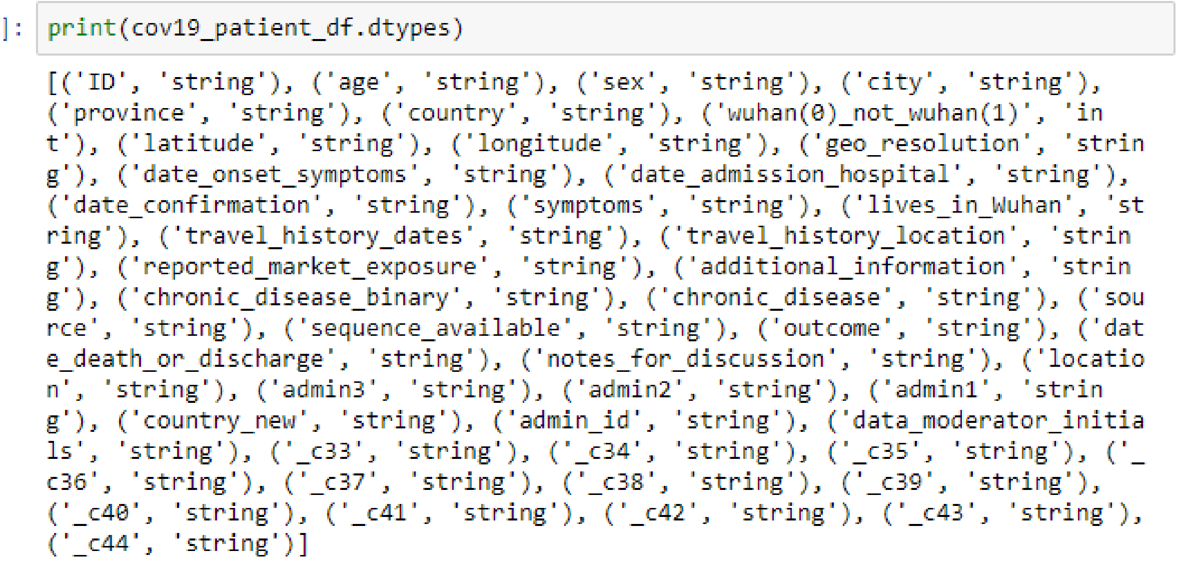


```
In [14]: runfile('F:/infosys722/report/OSAS/OSAS.py', wdir='F:/infosys722/report/OSAS')
The dimension of the nCoV 19 patient dataset is:
13174 rows and 32 columns.
```

Figure 3 – The amount of data from diagnosed patients

This dataset is relatively large, but it will be adjusted during the data pre-processing step, so the processing time here should not be a major issue.

Value types. The dataset used for this study contained several different data types, including numeric, categorical, boolean and text. There are several timestamp variables in this dataset, so time series analysis can be performed in this study.



```
]: print(cov19_patient_df.dtypes)

   [('ID', 'string'), ('age', 'string'), ('sex', 'string'), ('city', 'string'),
   ('province', 'string'), ('country', 'string'), ('wuhan(0)_not_wuhan(1)', 'in
   t'), ('latitude', 'string'), ('longitude', 'string'), ('geo_resolution', 'strin
   g'), ('date_onset_symptoms', 'string'), ('date_admission_hospital', 'string'),
   ('date_confirmation', 'string'), ('symptoms', 'string'), ('lives_in_Wuhan', 'st
   ring'), ('travel_history_dates', 'string'), ('travel_history_location', 'strin
   g'), ('reported_market_exposure', 'string'), ('additional_information', 'strin
   g'), ('chronic_disease_binary', 'string'), ('chronic_disease', 'string'), ('sou
   rce', 'string'), ('sequence_available', 'string'), ('outcome', 'string'), ('dat
   e_death_or_discharge', 'string'), ('notes_for_discussion', 'string'), ('locatio
   n', 'string'), ('admin3', 'string'), ('admin2', 'string'), ('admin1', 'strin
   g'), ('country_new', 'string'), ('admin_id', 'string'), ('data_moderator_initia
   ls', 'string'), ('_c33', 'string'), ('_c34', 'string'), ('_c35', 'string'), ('_
   c36', 'string'), ('_c37', 'string'), ('_c38', 'string'), ('_c39', 'string'),
   ('_c40', 'string'), ('_c41', 'string'), ('_c42', 'string'), ('_c43', 'string'),
   ('_c44', 'string')]
```

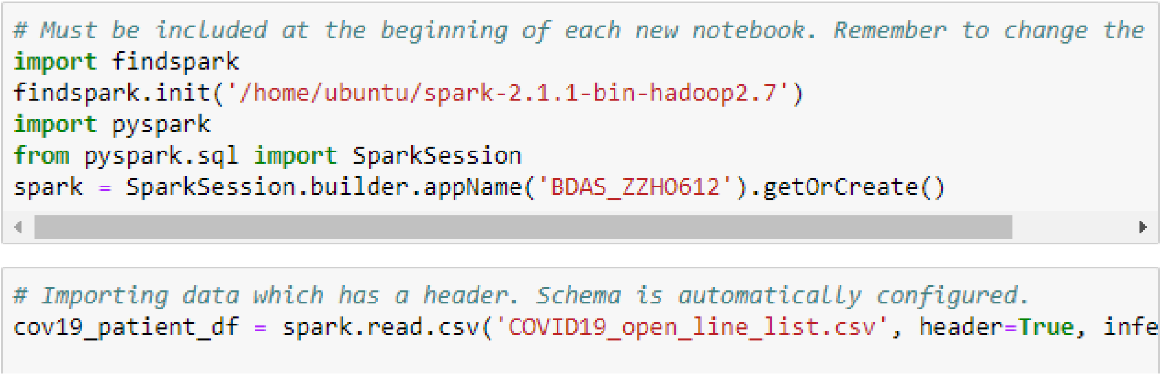
Figure 4 -- Value types of data from diagnosed patients

Coding schemes. All categorical variables in the original dataset will be transformed into numeric factors during data processing for modeling purposes in later steps.

2.3 Explore the data

- **File type conversion.**
  The original dataset was in CSV format. First, perform the file type conversion from CSV file to XLSX file (Excel workbook) to facilitate later operations.
- **Importing dataset into Python**

There is no cleaning or modifying on the dataset before importing it into Python.



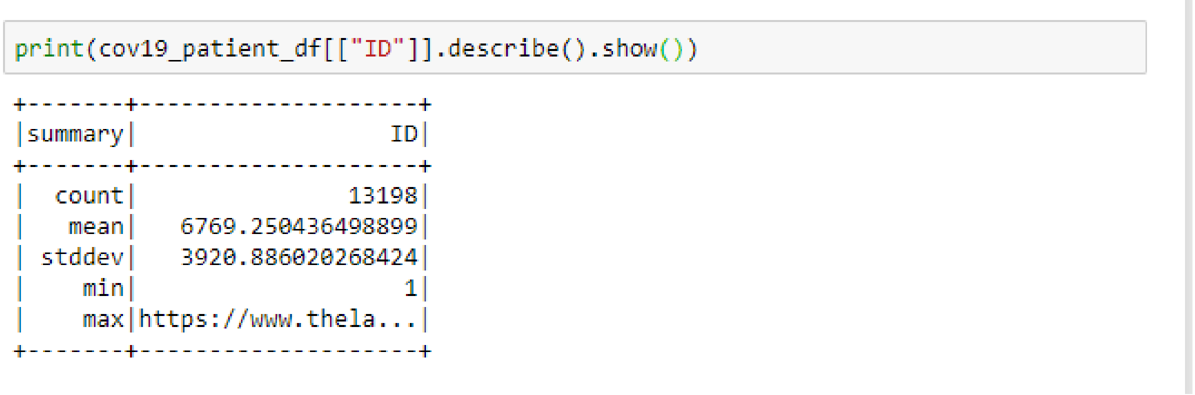
```
# Must be included at the beginning of each new notebook. Remember to change the
import findspark
findspark.init('/home/ubuntu/spark-2.1.1-bin-hadoop2.7')
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('BDAS_ZZHO612').getOrCreate()
```

```
# Importing data which has a header. Schema is automatically configured.
cov19_patient_df = spark.read.csv('COVID19_open_line_list.csv', header=True, infe
```

Figure 5 – read in data in PySpark

- **Value reading and initial type check**

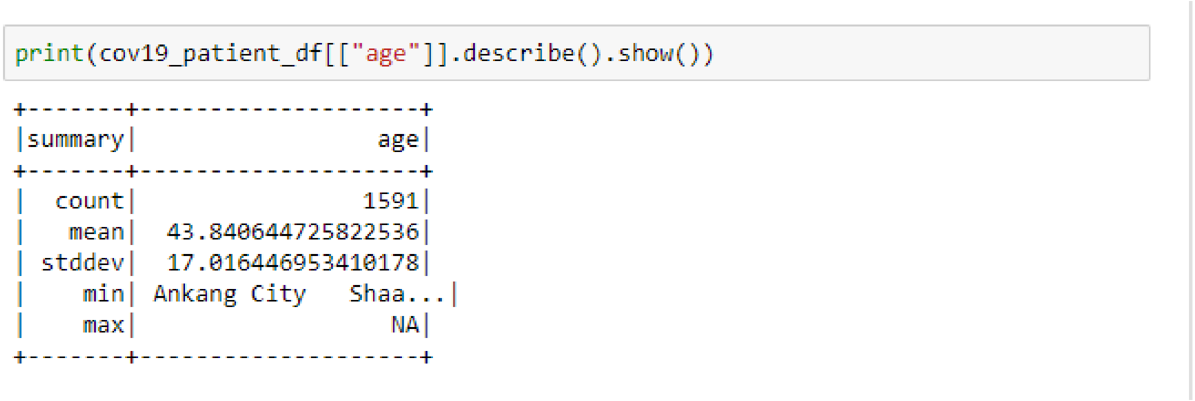
A quick insight of the data types and values can be done in Python by a function called ‘describe’ in the ‘dataframe’ class of the ‘Pandas’ package. Results comes below with some simple statistics on the values:



```
print(cov19_patient_df[["ID"]].describe().show())
```

```
+-------+--------------------+
|summary|                  ID|
+-------+--------------------+
|  count|               13198|
|   mean|   6769.250436498899|
| stddev|   3920.886020268424|
|    min|                   1|
|    max|https://www.thela...|
+-------+--------------------+
```

Figure 6 – value reading



```
print(cov19_patient_df[["age"]].describe().show())
```

```
+-------+--------------------+
|summary|                 age|
+-------+--------------------+
|  count|                1591|
|   mean|  43.840644725822536|
| stddev|  17.016446953410178|
|    min| Ankang City   Shaa...|
|    max|                  NA|
+-------+--------------------+
```

Figure 7 – value reading

Noted that the simple value statistics is pretty limited at this stage. Most of the variables in this dataset are categorical so that it needs to be further investigate on their values.

- Initial data visualisation

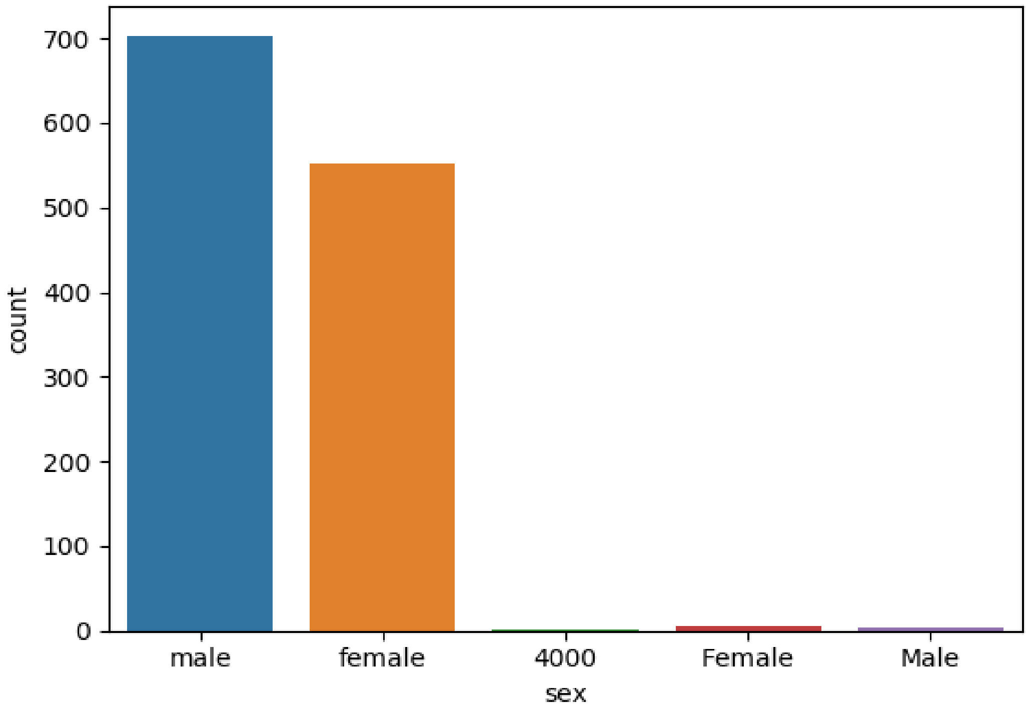


Figure 8 – initial data visualisation

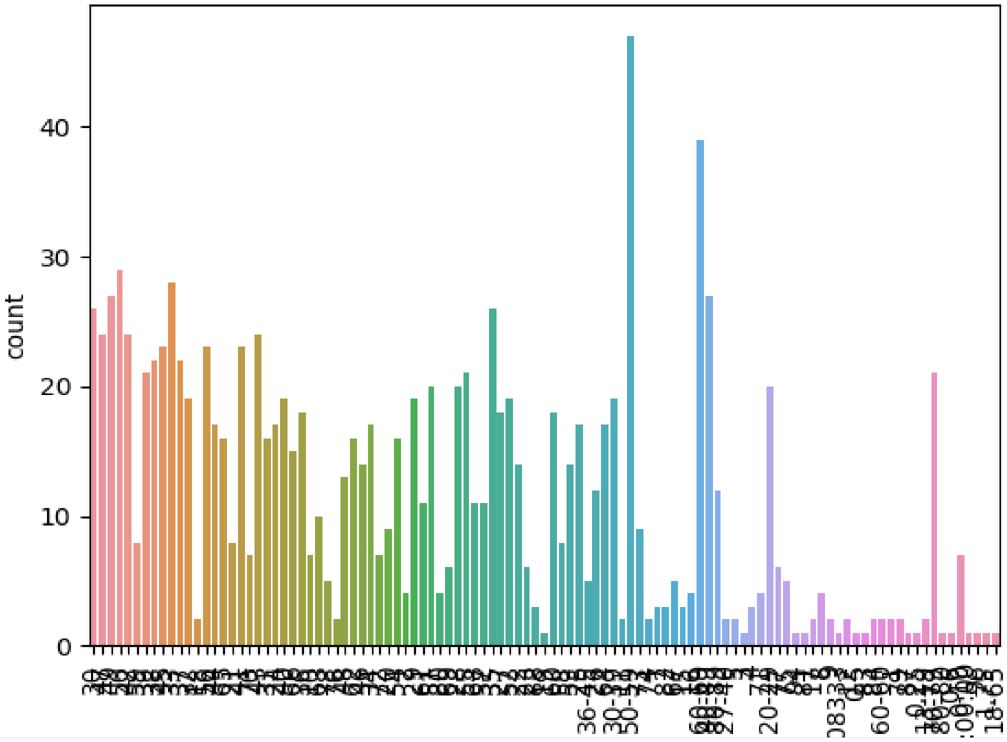


Figure 9 – initial data vaisualisation

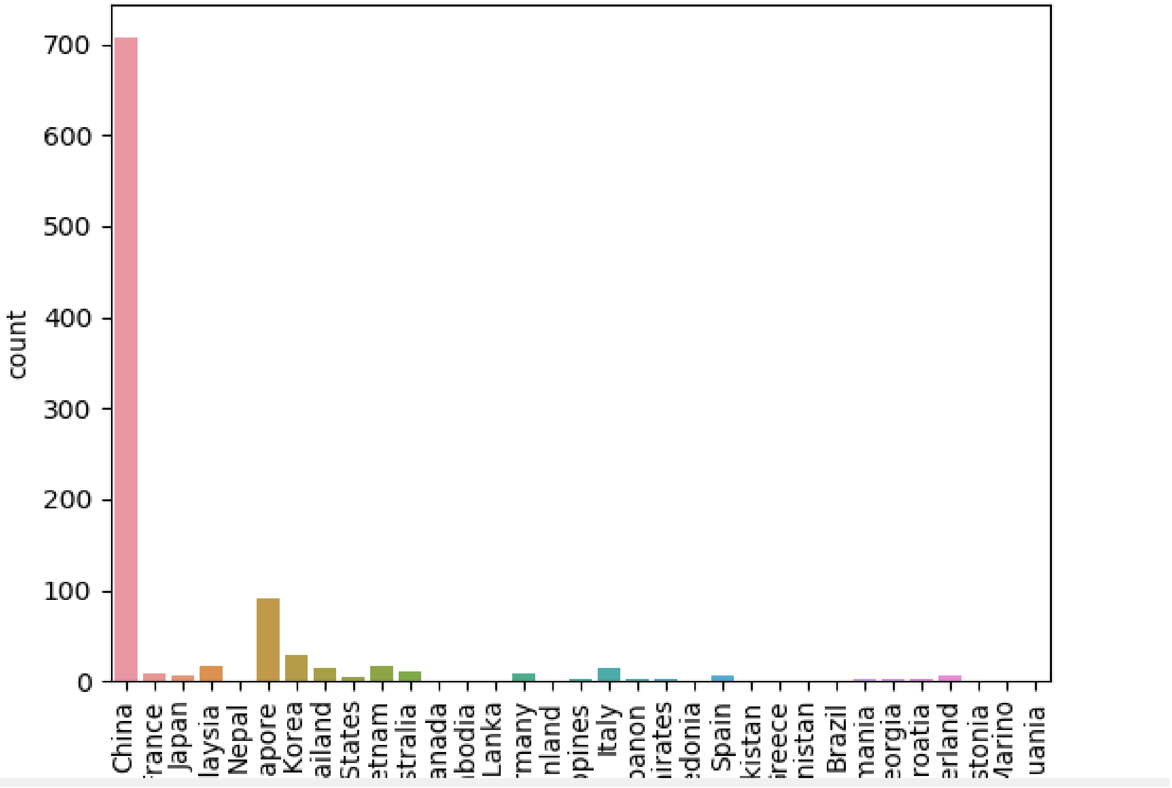
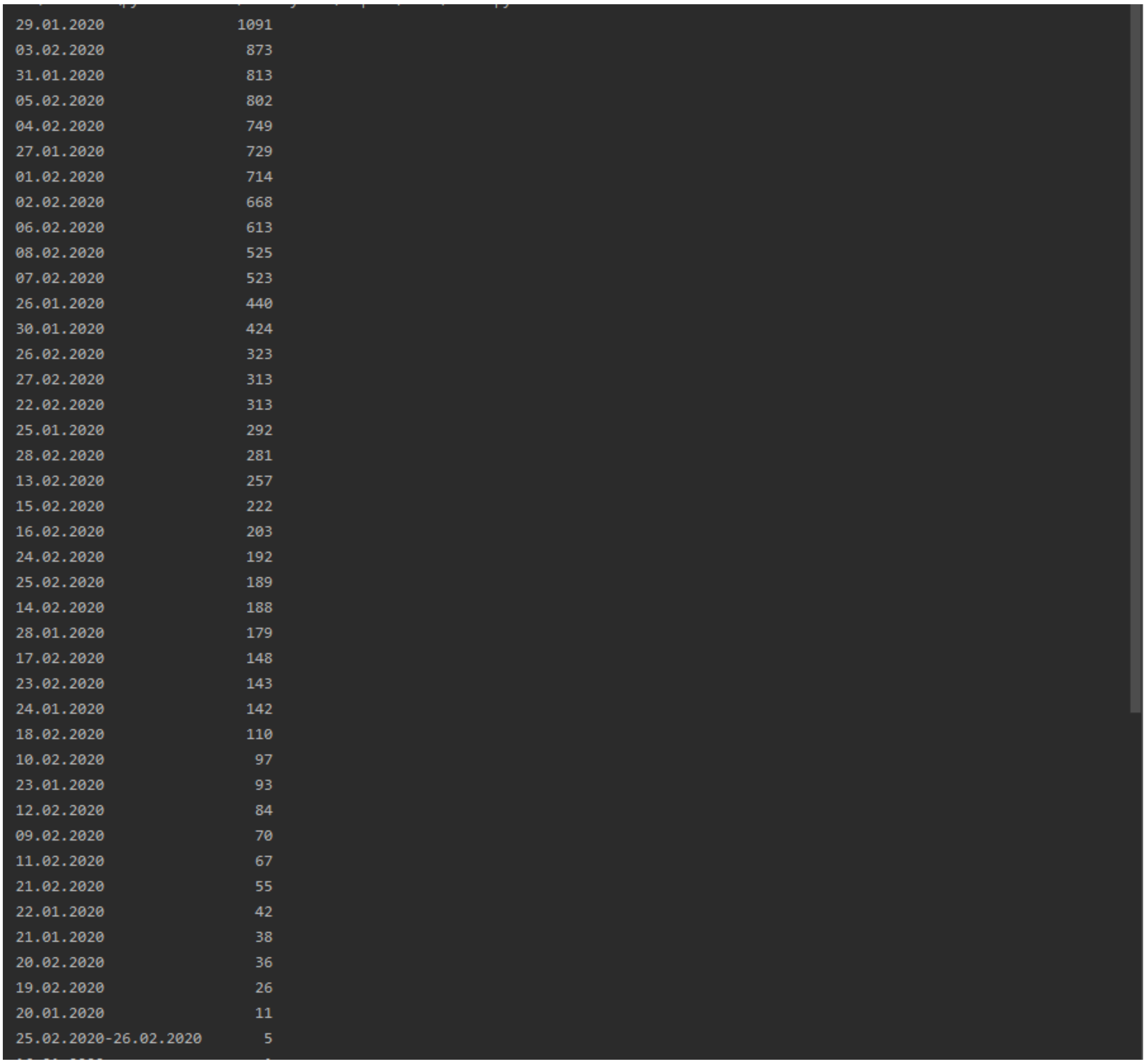


Figure 10 – initial data visualisation

The result shows that most of the examples are from China, which is not surprising, as China is considered the first country to suffer a new coronavirus outbreak. There are 10,446 samples from China in this dataset. Other countries with relatively large numbers of samples include Italy, Japan, and South Korea.

The date with the greatest number of the patient confirmed on infection of the new coronavirus:



```
29.01.2020              1091
03.02.2020               873
31.01.2020               813
05.02.2020               802
04.02.2020               749
27.01.2020               729
01.02.2020               714
02.02.2020               668
06.02.2020               613
08.02.2020               525
07.02.2020               523
26.01.2020               440
30.01.2020               424
26.02.2020               323
27.02.2020               313
22.02.2020               313
25.01.2020               292
28.02.2020               281
13.02.2020               257
15.02.2020               222
16.02.2020               203
24.02.2020               192
25.02.2020               189
14.02.2020               188
28.01.2020               179
17.02.2020               148
23.02.2020               143
24.01.2020               142
18.02.2020               110
10.02.2020                97
23.01.2020                93
12.02.2020                84
09.02.2020                70
11.02.2020                67
21.02.2020                55
22.01.2020                42
21.01.2020                38
20.02.2020                36
19.02.2020                26
20.01.2020                11
25.02.2020-26.02.2020      5
```

Figure 11 – initial data visualisation

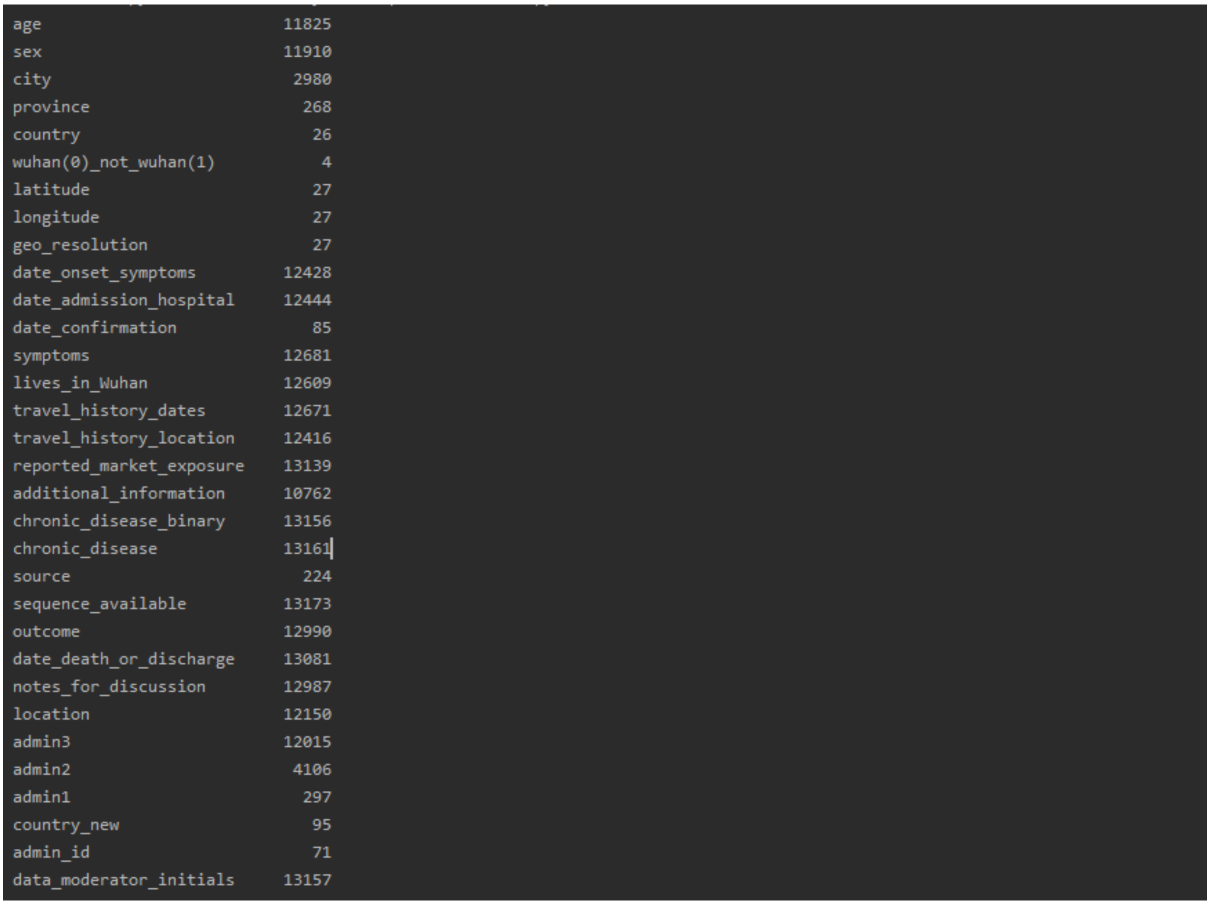
From this result, I can see that the days when the number of people diagnosed with the new coronavirus infection are high are concentrated in late January and early February 2020. This may be related to the decision of the Chinese government to block off the city of Wuhan, the earliest affected by the new coronavirus outbreak.

2.4 Verify the data quality

In data analysis, the quality of the data set is particularly important. Because the relationship between the quality of the data set and the analysis results is positively correlated. In other words, if the quality of the data set is not high, the analysis that can be done is very limited.

- **Missing data:**

Miss data include values that are blank or coded as a non-response such as null value. On the data auditing table, I can find it under the columns of Null Value, Empty String, White Space and Blank Value:

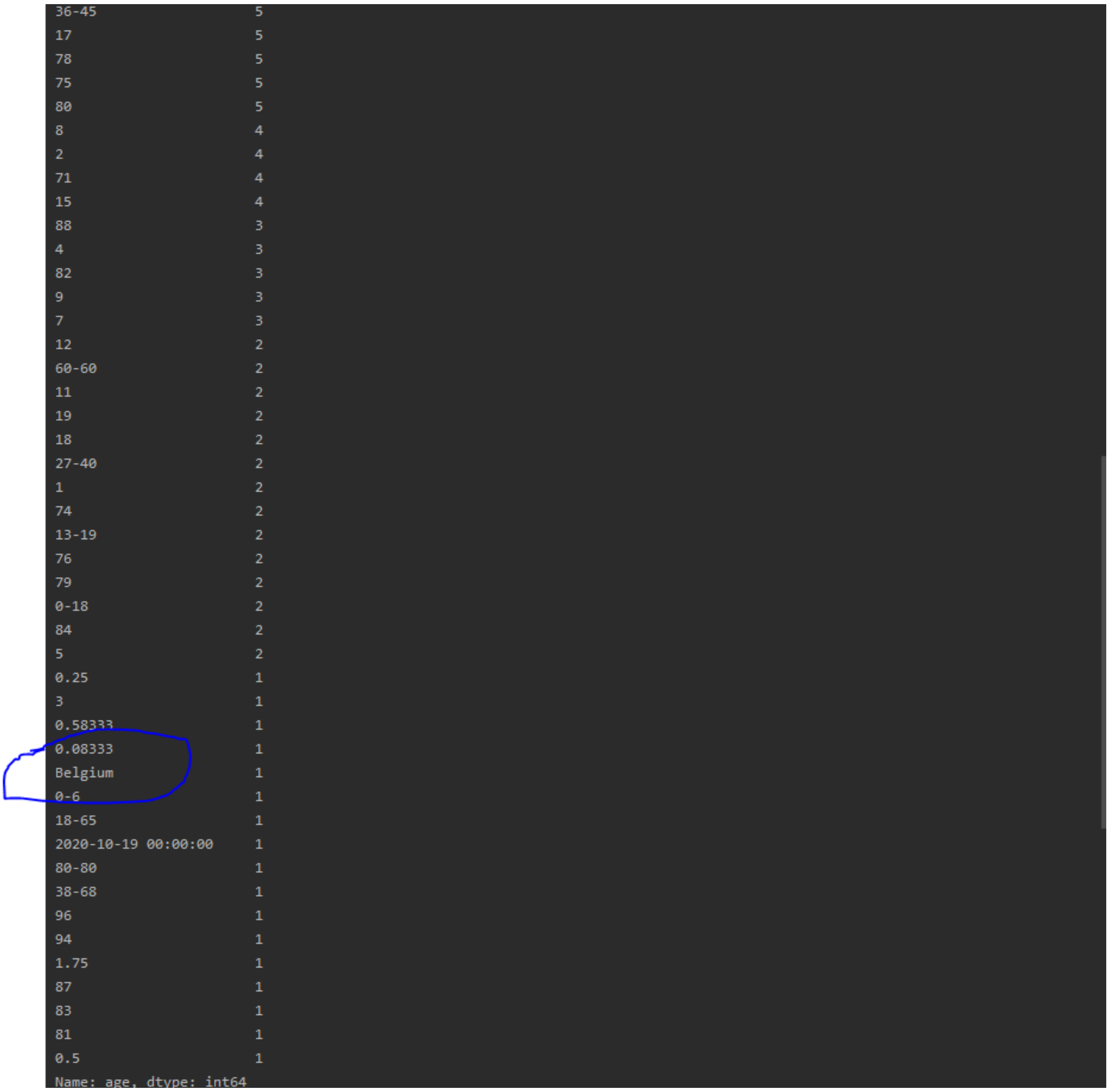


```
age                          11825
sex                          11910
city                          2980
province                       268
country                         26
wuhan(0)_not_wuhan(1)            4
latitude                        27
longitude                       27
geo_resolution                  27
date_onset_symptoms          12428
date_admission_hospital      12444
date_confirmation               85
symptoms                     12681
lives_in_Wuhan               12609
travel_history_dates         12671
travel_history_location      12416
reported_market_exposure     13139
additional_information       10762
chronic_disease_binary       13156
chronic_disease              13161
source                         224
sequence_available           13173
outcome                      12990
date_death_or_discharge      13081
notes_for_discussion         12987
location                     12150
admin3                       12015
admin2                        4106
admin1                         297
country_new                     95
admin_id                        71
data_moderator_initials      13157
```

Figure 12 -- verifying data

It shows that there are a lot of missing values in this dataset, which indicates the relatively poor quality of this dataset. However, it is tolerable since the new coronavirus outbreak is still continuing and the dataset is still constantly updating. In this study, I will use the limited valid data in this data set for analysis

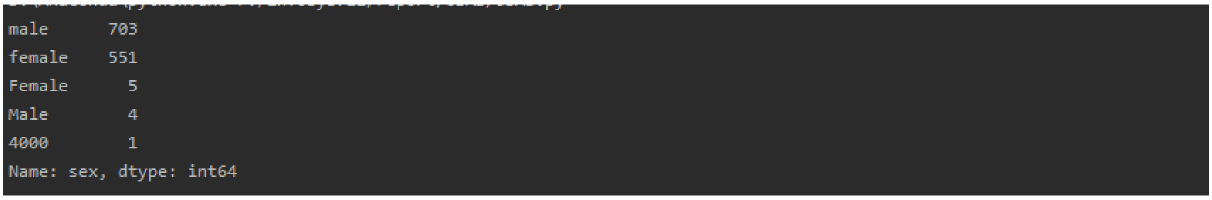
- **Data errors:**
  Data errors are usually typographical errors made in entering the data. There are some obvious errors in this dataset, such as the value "Belgium" in the age column which obviously should be in the country column.



```
36-45                  5
17                     5
78                     5
75                     5
80                     5
8                      4
2                      4
71                     4
15                     4
88                     3
4                      3
82                     3
9                      3
7                      3
12                     2
60-60                  2
11                     2
19                     2
18                     2
27-40                  2
1                      2
74                     2
13-19                  2
76                     2
79                     2
0-18                   2
84                     2
5                      2
0.25                   1
3                      1
0.58333                1
0.08333                1
Belgium                1
0-6                    1
18-65                  1
2020-10-19 00:00:00    1
80-80                  1
38-68                  1
96                     1
94                     1
1.75                   1
87                     1
83                     1
81                     1
0.5                    1
Name: age, dtype: int64
```

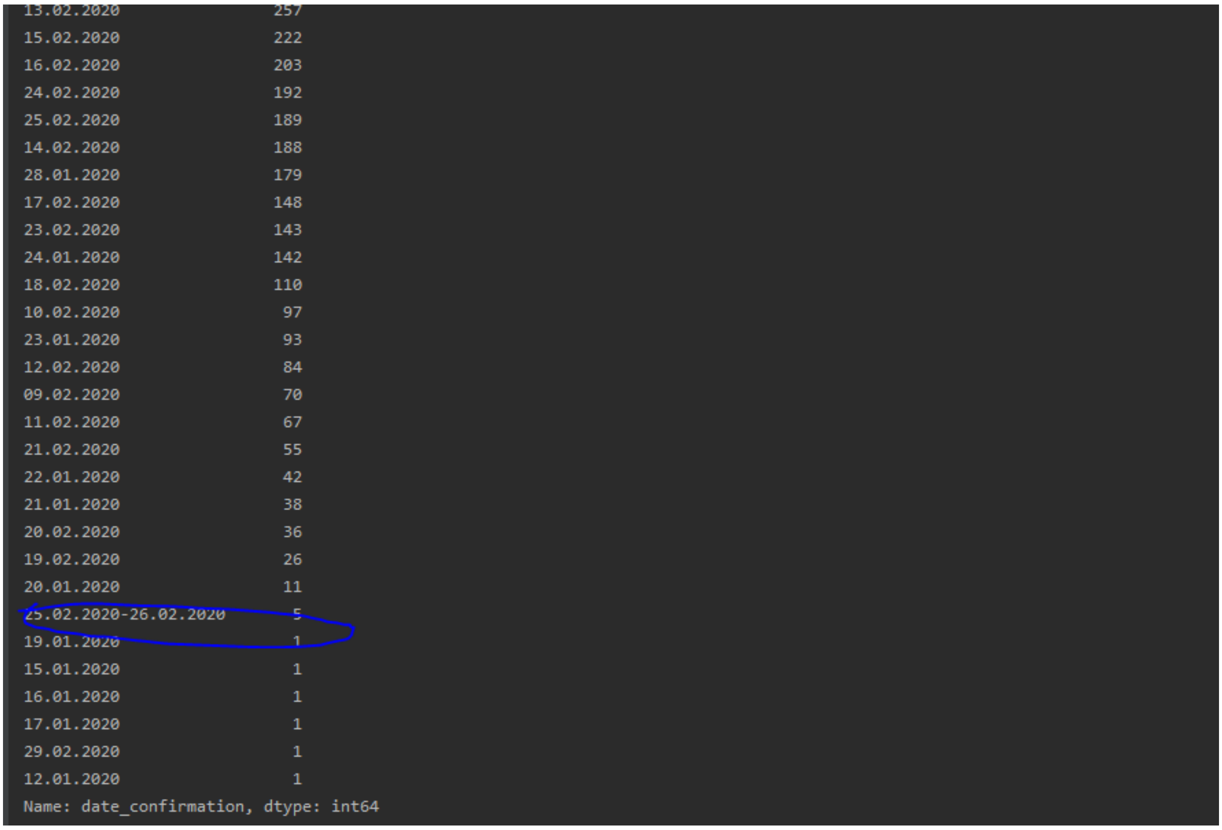
Figure 13 -- verifying data

- **Measurement errors：**
  Measurement errors include data entered correctly but based on the wrong measurement scheme. There are no related variables in this dataset, so measurement errors should not be overly concerned.
- **Coding inconsistencies.**
  Coding inconsistencies usually involve non-standard measurement units or inconsistent values. In this data set, genders were entered in lowercase and uppercase.



```
male      703
female    551
Female      5
Male        4
4000        1
Name: sex, dtype: int64
```

The same problem can also be view in other attributes such as ‘date_confirm’:

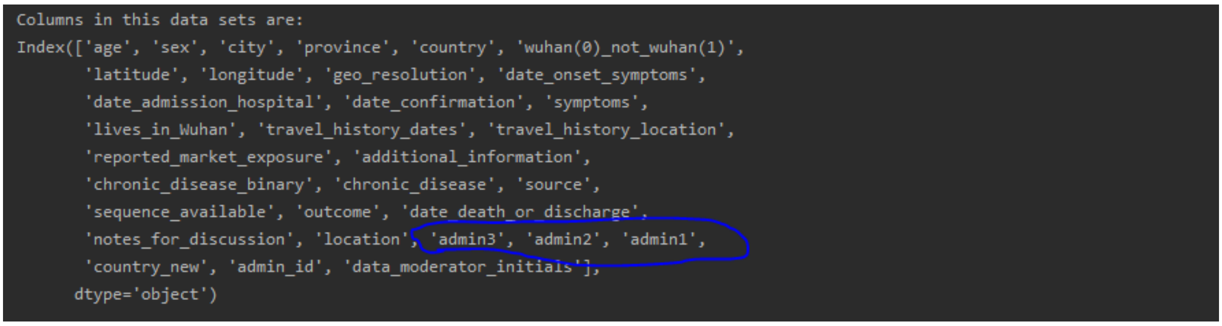


```
13.02.2020              257
15.02.2020              222
16.02.2020              203
24.02.2020              192
25.02.2020              189
14.02.2020              188
28.01.2020              179
17.02.2020              148
23.02.2020              143
24.01.2020              142
18.02.2020              110
10.02.2020               97
23.01.2020               93
12.02.2020               84
09.02.2020               70
11.02.2020               67
21.02.2020               55
22.01.2020               42
21.01.2020               38
20.02.2020               36
19.02.2020               26
20.01.2020               11
25.02.2020-26.02.2020     5
19.01.2020                1
15.01.2020                1
16.01.2020                1
17.01.2020                1
29.02.2020                1
12.01.2020                1
Name: date_confirmation, dtype: int64
```

Figure 14 -- verifying data

- **Bad metadata:**

Bad metadata include mismatches between the apparent meaning of a field and the meaning stated in a field name or definition. In this data set, there are column names such as "admin1" whose meaning cannot be directly seen, and the data source does not have detailed documentation explaining the meaning of these columns:



```
Columns in this data sets are:
Index(['age', 'sex', 'city', 'province', 'country', 'wuhan(0)_not_wuhan(1)',
       'latitude', 'longitude', 'geo_resolution', 'date_onset_symptoms',
       'date_admission_hospital', 'date_confirmation', 'symptoms',
       'lives_in_Wuhan', 'travel_history_dates', 'travel_history_location',
       'reported_market_exposure', 'additional_information',
       'chronic_disease_binary', 'chronic_disease', 'source',
       'sequence_available', 'outcome', 'date_death_or_discharge',
       'notes_for_discussion', 'location', 'admin3', 'admin2', 'admin1',
       'country_new', 'admin_id', 'data_moderator_initials'],
      dtype='object')
```

Figure 15 -- verifying data

3 Data Preparation

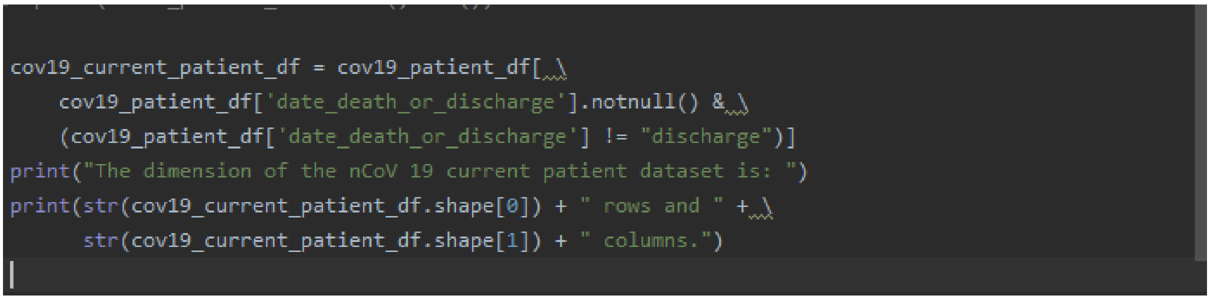
3.1 Select the data

- **Selecting Items(rows):**

There are two main goals in this study.

The goal is to find the relationship between death and other factors of patients with new coronavirus. Thus, I need to identify the dead patient observations from my patient data.

Among the patient data I have, there is a column named "date_death_or_discharge". If the patient was dead at the time this data was collected, this column will show the patient's death date. Or if the patient was discharged when this data was collected, this column will show "discharge". So I need to subset my patient data as shown below:



```
cov19_current_patient_df = cov19_patient_df[ \
    cov19_patient_df['date_death_or_discharge'].notnull() & \
    (cov19_patient_df['date_death_or_discharge'] != "discharge")]
print("The dimension of the nCoV 19 current patient dataset is: ")
print(str(cov19_current_patient_df.shape[0]) + " rows and " + \
      str(cov19_current_patient_df.shape[1]) + " columns.")
```

Figure 16 -- selecting data

And the result distribution for the column:

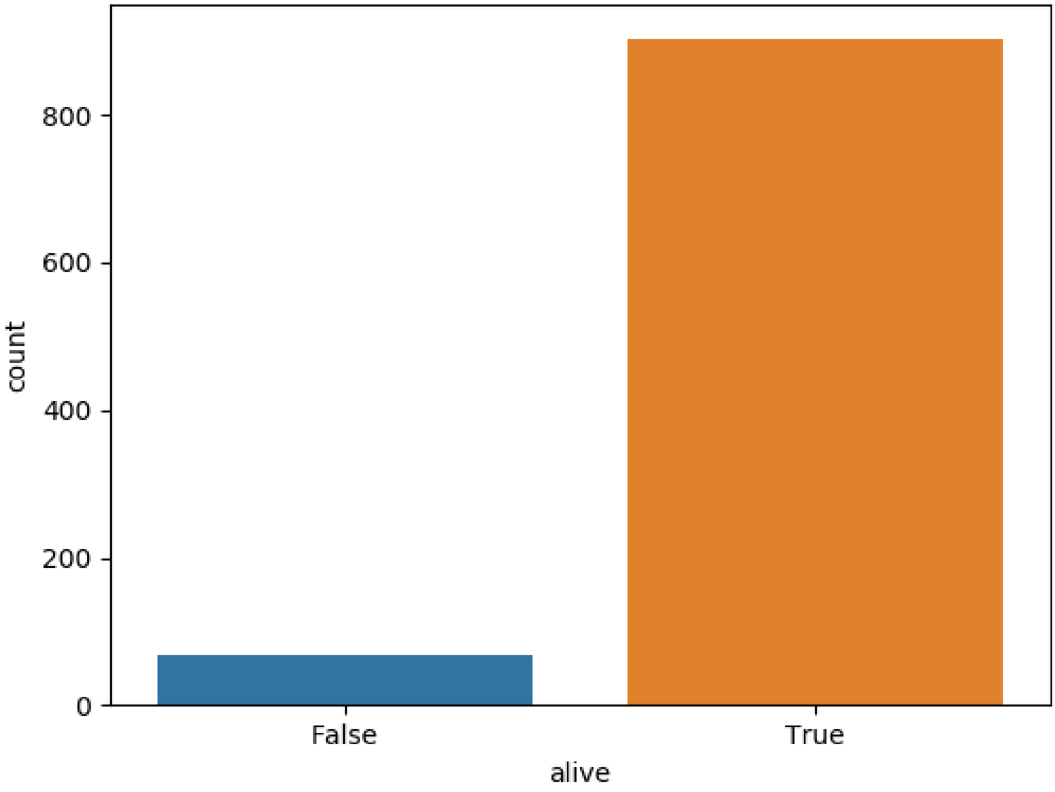
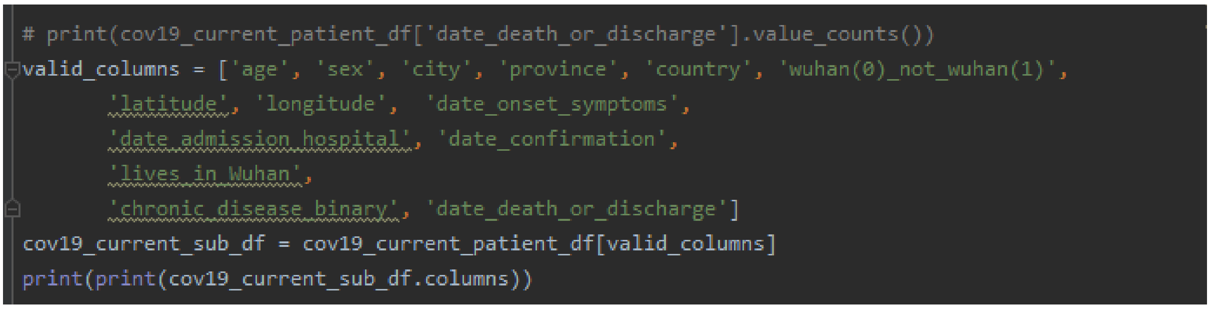


Figure 17 -- data distribution

- **Selecting attributes(columns):**

There are some factors certainly should not be included in the analysis such as ID. Also, there are a few possible factors that could potentially have a huge impact of the death of patients such as age and gender. Selecting attributes based on common sense is a reasonable step here:



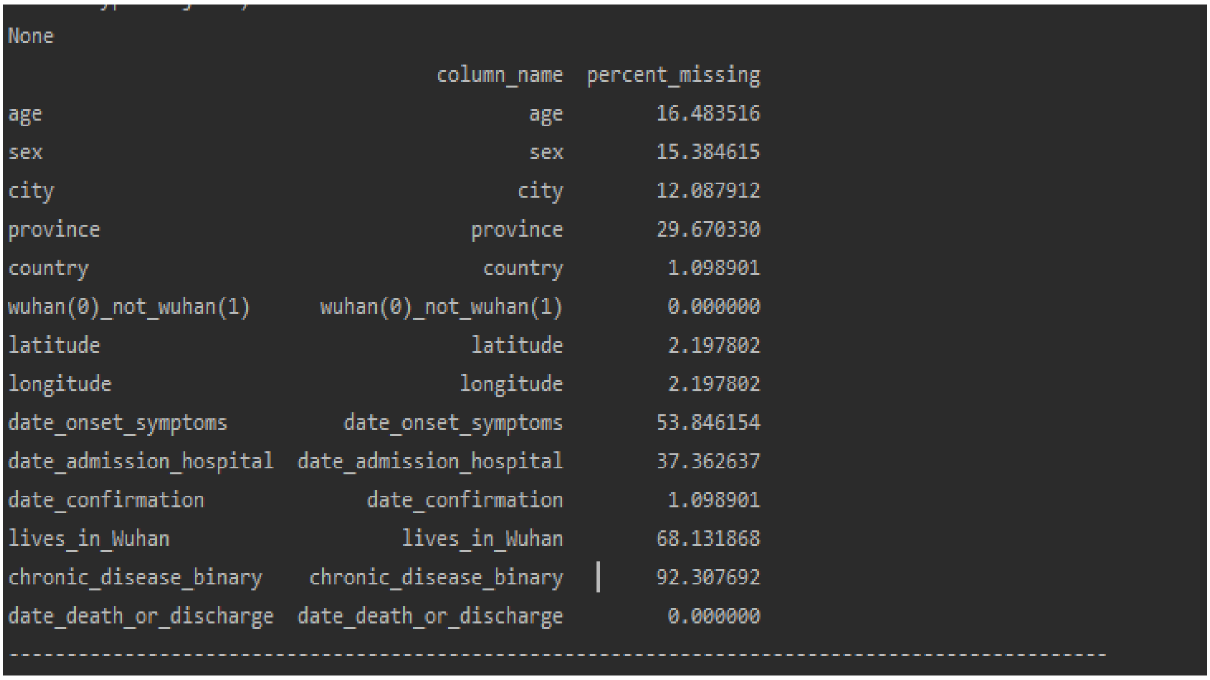
```
# print(cov19_current_patient_df['date_death_or_discharge'].value_counts())
valid_columns = ['age', 'sex', 'city', 'province', 'country', 'wuhan(0)_not_wuhan(1)',
        'latitude', 'longitude',  'date_onset_symptoms',
        'date_admission_hospital', 'date_confirmation',
        'lives_in_Wuhan',
        'chronic_disease_binary', 'date_death_or_discharge']
cov19_current_sub_df = cov19_current_patient_df[valid_columns]
print(print(cov19_current_sub_df.columns))
```

Figure 18 -- selecting data

3.2 Clean the data

- **Missing data**

Due to the special nature of the new coronavirus data, many attributes in this data have a low degree of completion. In other words, there are many missing values in these attributes. However, this does not mean that these attributes are not important. It is difficult for us to extract valid information from these attributes in this research.



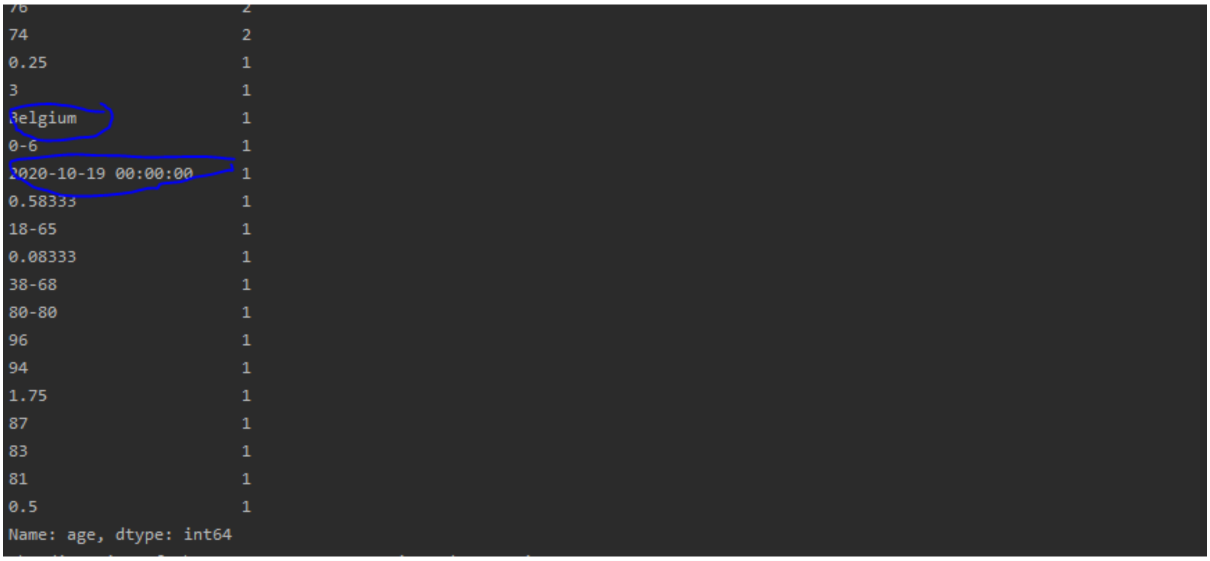
```
None
                                     column_name  percent_missing
age                                          age        16.483516
sex                                          sex        15.384615
city                                        city        12.087912
province                                province        29.670330
country                                  country         1.098901
wuhan(0)_not_wuhan(1)      wuhan(0)_not_wuhan(1)         0.000000
latitude                                latitude         2.197802
longitude                              longitude         2.197802
date_onset_symptoms          date_onset_symptoms        53.846154
date_admission_hospital  date_admission_hospital        37.362637
date_confirmation              date_confirmation         1.098901
lives_in_Wuhan                    lives_in_Wuhan        68.131868
chronic_disease_binary    chronic_disease_binary        92.307692
date_death_or_discharge  date_death_or_discharge         0.000000
---------------------------------------------------------------------------------------------
```

Figure 19 – Data audit – missing data

Attribute I will drop out here is “chronic_disease_binary” and “lives_in_Wuhan”which having a high missing rate. Also, attributes that contains overlapped information are drops like ‘latitude’ and ‘longitude’.

- **Data errors**

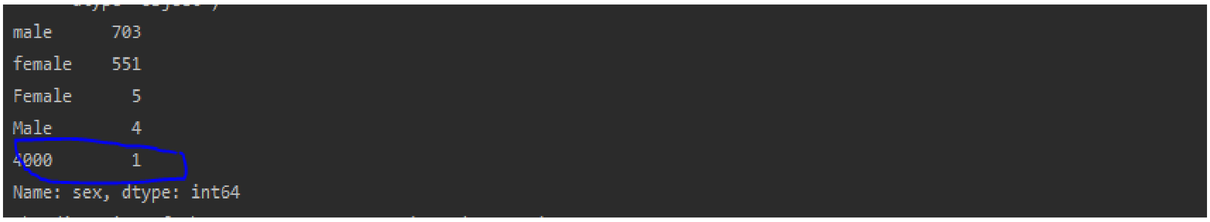
As mentioned earlier, since the current outbreak of new coronaviruses is still ongoing, the relevant data is still being continuously updated, and the data used in this study is no exception. In addition, most of these data are collected manually, so you can see that there are many data errors in many attributes. For example:



```
76                     2
74                     2
0.25                   1
3                      1
Belgium                1
0-6                    1
2020-10-19 00:00:00    1
0.58333                1
18-65                  1
0.08333                1
38-68                  1
80-80                  1
96                     1
94                     1
1.75                   1
87                     1
83                     1
81                     1
0.5                    1
Name: age, dtype: int64
```

Figure 20 – data inspection

The age of most observations in the patient data is empty, and there is another sample with a value of "Belgium" in this column. Obviously this should be the patient's nationality or travel history rather than age. This may be because a large part of this data is collected by health care workers, who are usually busy, and it is common for errors or omissions of certain data. The same also happens in other attributes like sex:



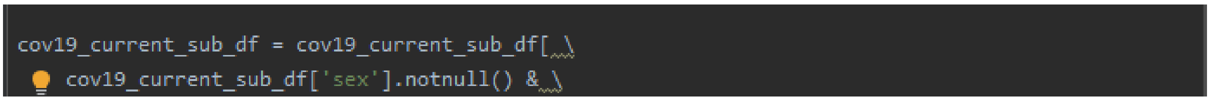
```
male      703
female    551
Female      5
Male        4
4000        1
Name: sex, dtype: int64
```

Figure 21 -- data inspection

The sex of most patient data is empty, and there is a sample with a value of 4000 in this column which does not make any sense at all.

To resolve these problems, I need to subset out data and filter out those invalid observations.

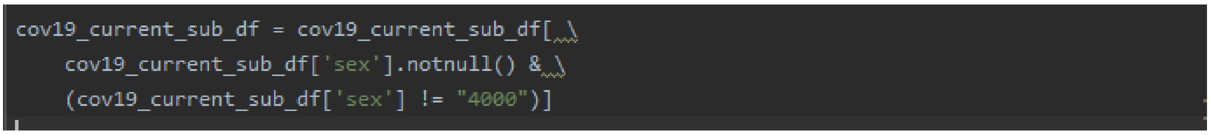
I start by filter it on the sex column because it is easier. To filter out blank values, I could make the values of this column to be a string and check for their length:



```
cov19_current_sub_df = cov19_current_sub_df[ \
    cov19_current_sub_df['sex'].notnull() & \
```

Figure 22 -- data selection

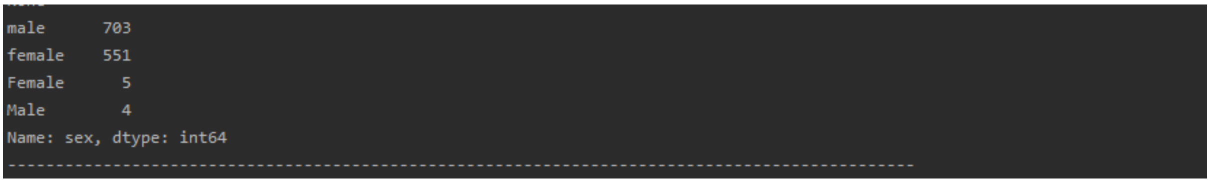
And then I could filter out the value “4000”:



```
cov19_current_sub_df = cov19_current_sub_df[ \
    cov19_current_sub_df['sex'].notnull() & \
    (cov19_current_sub_df['sex'] != "4000")]
```

Figure 23 -- cleaning

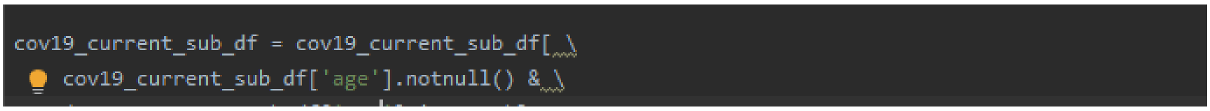
Then I get the updated distribution of the sex column:



```
male      703
female    551
Female      5
Male        4
Name: sex, dtype: int64
---------------------------------------------------------------------------------------------
```

Figure 24 -- cleaning

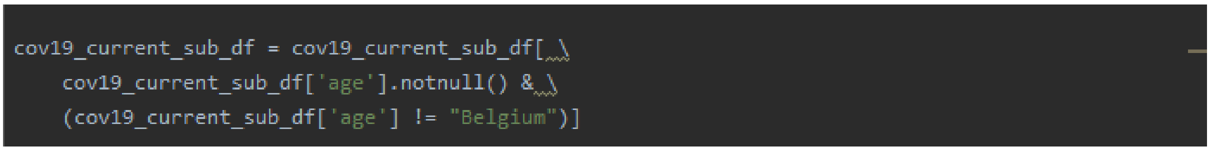
Now I start adding logical conditions on age. I first filter out values "N/A" and blank values:



```
cov19_current_sub_df = cov19_current_sub_df[ \
    cov19_current_sub_df['age'].notnull() & \
```

Figure 25 -- cleaning

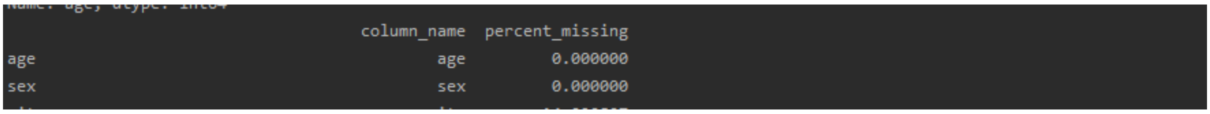
Also, the same as I did for the sex column, I could do the same to filter out values in age like "Belgium":



```
cov19_current_sub_df = cov19_current_sub_df[ \
    cov19_current_sub_df['age'].notnull() & \
    (cov19_current_sub_df['age'] != "Belgium")]
```

Figure 26 -- cleaning

Now if we look at the missing rate again:



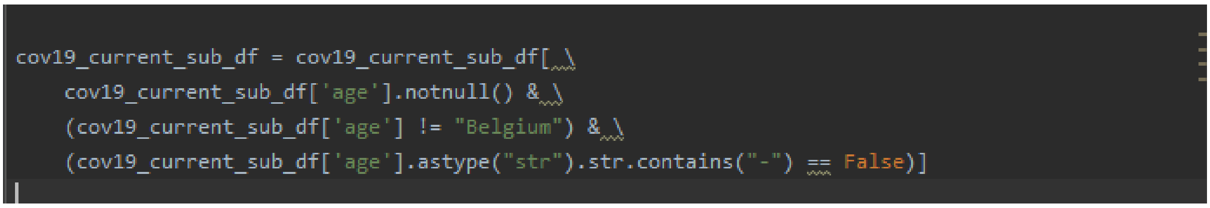
```
                                     column_name  percent_missing
age                                          age         0.000000
sex                                          sex         0.000000
```

Figure 27 -- cleaning

It shows that I have successfully decreased the missing rate of both the age column and the sex column to 0%.

- **Coding inconsistencies**

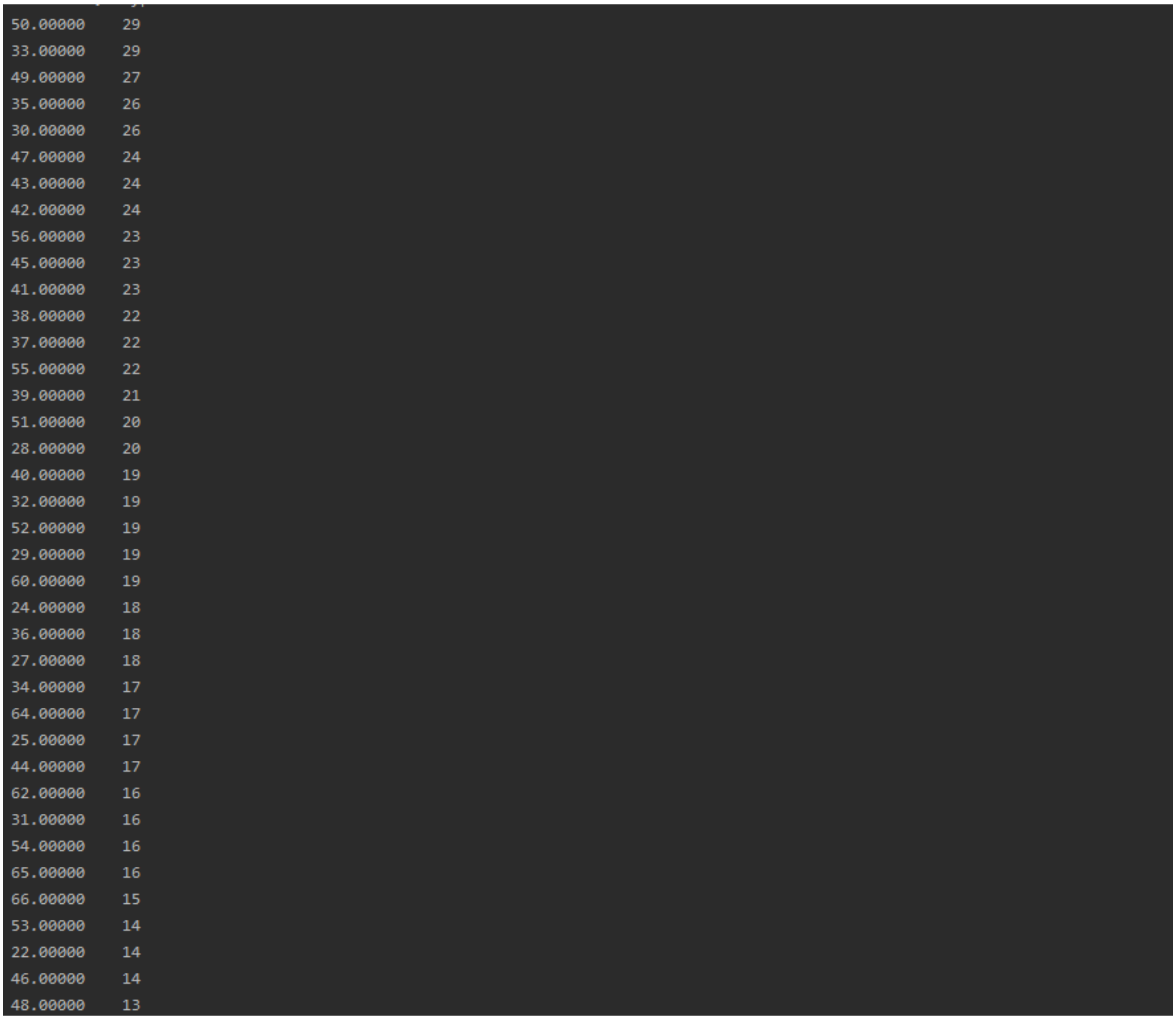
For the data mining goal, although I have successfully raised the complete rate of those attributes to a good level, I still have problems of coding inconsistency. For example, in the patient data, there are some observations having "18-65" in the age column which is not helpful for our study. I would like the age column contains just age numbers. Thus, I need to further filter the age column by getting rid of things like "*-*":



```
cov19_current_sub_df = cov19_current_sub_df[ \
    cov19_current_sub_df['age'].notnull() & \
    (cov19_current_sub_df['age'] != "Belgium") & \
    (cov19_current_sub_df['age'].astype("str").str.contains("-") == False)]
```

Figure 28 -- cleaning

Now the distribution graph of the age column looks like:

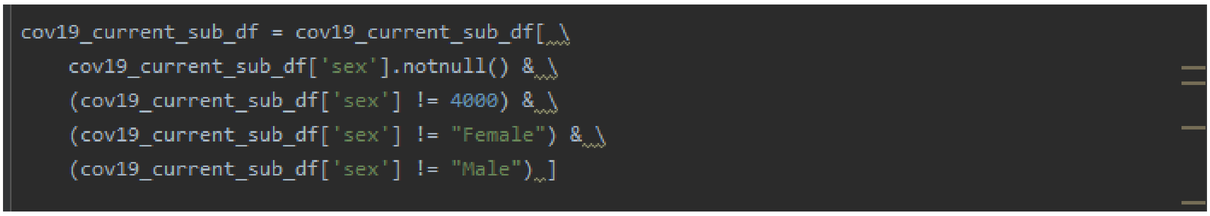


```
50.00000    29
33.00000    29
49.00000    27
35.00000    26
30.00000    26
47.00000    24
43.00000    24
42.00000    24
56.00000    23
45.00000    23
41.00000    23
38.00000    22
37.00000    22
55.00000    22
39.00000    21
51.00000    20
28.00000    20
40.00000    19
32.00000    19
52.00000    19
29.00000    19
60.00000    19
24.00000    18
36.00000    18
27.00000    18
34.00000    17
64.00000    17
25.00000    17
44.00000    17
62.00000    16
31.00000    16
54.00000    16
65.00000    16
66.00000    15
53.00000    14
22.00000    14
46.00000    14
48.00000    13
```

Figure 29 -- data distribution

Again, I need to do similar stuff on the sex column. Noted that from the distribution graph of the sex column, the only thing I need to worry about is some values in camelCase but most of the value in lower case:

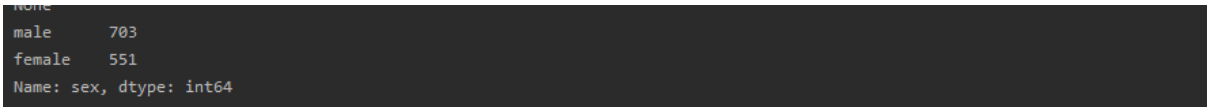
Since there is only a small portion of data having that problem, I will simply filter them out:



```
cov19_current_sub_df = cov19_current_sub_df[ \
    cov19_current_sub_df['sex'].notnull() & \
    (cov19_current_sub_df['sex'] != 4000) & \
    (cov19_current_sub_df['sex'] != "Female") & \
    (cov19_current_sub_df['sex'] != "Male") ]
```

Figure 30 -- cleaning

After doing that, the distribution of the sex column looks like:



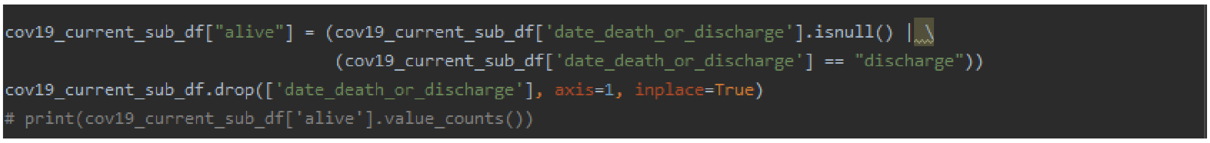
```
None
male      703
female    551
Name: sex, dtype: int64
```

Figure 31-- data distribution

3.3 Construct the data

- **Deriving attributes**

One of the goals of my data mining in this study was to discover the relationship between the mortality of people with new coronavirus infection and other factors. As stated in section 3.1, I need to identify the dead patient observations from my patient data. The way of doing it is to use the column "date_death_or_discharge". As discussed before, if the patient was dead at the time this data was collected, this column will show the patient's death date. Or if the patient was discharged when this data was collected, this column will show "discharge". Base on that, I will derive a new column named "alive" as a flag type with 0 if the patient is dead and 1 otherwise.



```
cov19_current_sub_df["alive"] = (cov19_current_sub_df['date_death_or_discharge'].isnull() | \
                                  (cov19_current_sub_df['date_death_or_discharge'] == "discharge"))
cov19_current_sub_df.drop(['date_death_or_discharge'], axis=1, inplace=True)
# print(cov19_current_sub_df['alive'].value_counts())
```

Figure 32 -- cleaning

After that, the distribution of the derived attribute “alive” looks like:

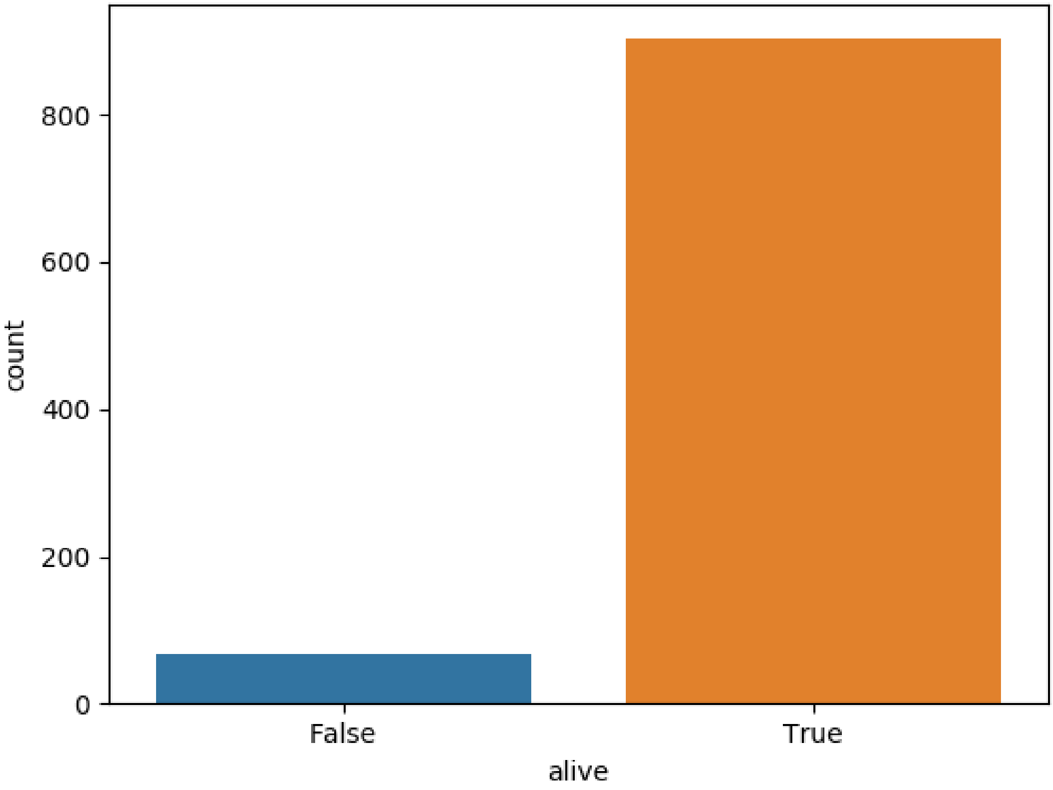
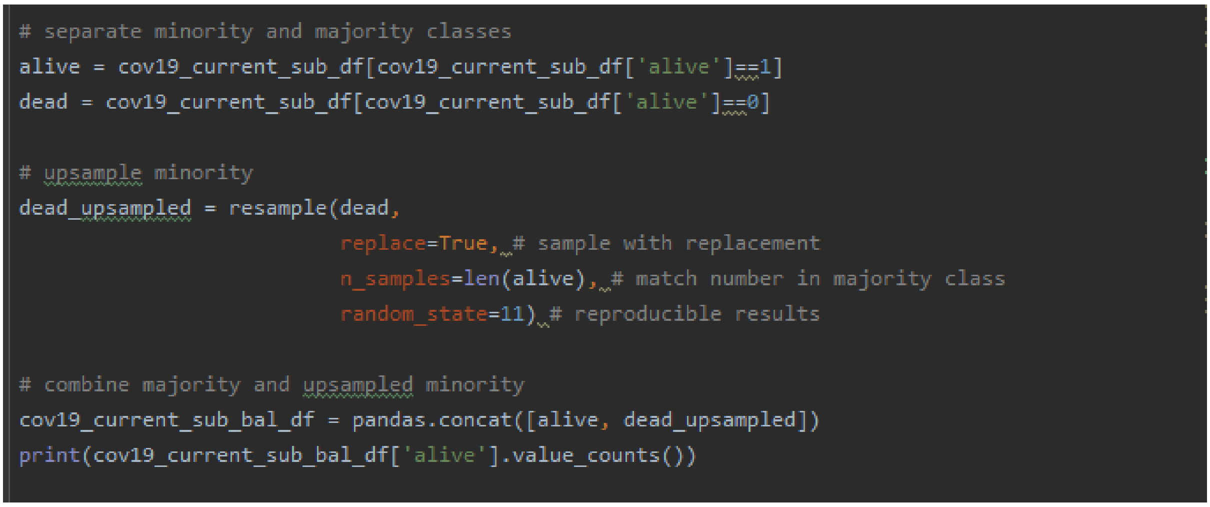


Figure 33 -- data distribution

- **Generating records**

The results from the derived attributes show that 7% of the observations in my filtered pre-treated patient data were dead patients, not far from the 3.4% mortality rate of the new coronavirus reported on the Worldometers website (Coronavirus (COVID-19) Mortality Rate, 2020). However, for data analysis purposes, this is called unbalanced data. This may have a negative influence on the overall predictive power of my model later because my model may just be lazy and classify all observations to the major group of the target attribute and still get a high accuracy. Therefore, I would balance my patient data for better analysis. Because the number of observations of dead patients and the number of non-dead patients varies greatly, I will not use the method of reducing the number of observations of deceased patients. I will balance the number of observations by randomly increasing the number of observations of deceased patients.



```
# separate minority and majority classes
alive = cov19_current_sub_df[cov19_current_sub_df['alive']==1]
dead = cov19_current_sub_df[cov19_current_sub_df['alive']==0]

# upsample minority
dead_upsampled = resample(dead,
                          replace=True, # sample with replacement
                          n_samples=len(alive), # match number in majority class
                          random_state=11) # reproducible results

# combine majority and upsampled minority
cov19_current_sub_bal_df = pandas.concat([alive, dead_upsampled])
print(cov19_current_sub_bal_df['alive'].value_counts())
```

Figure 34 – Balancing data

Now I have a balanced data for my analysis.

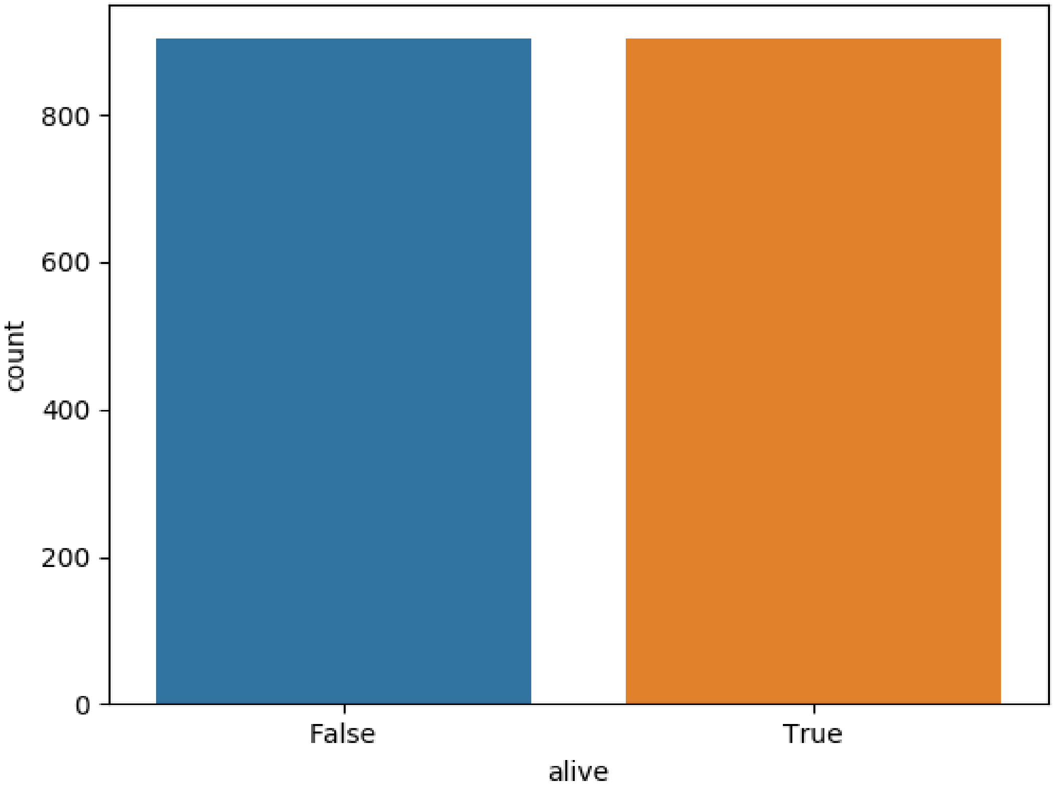


Figure 35 -- data distribution

3.4 Integrate various data sources

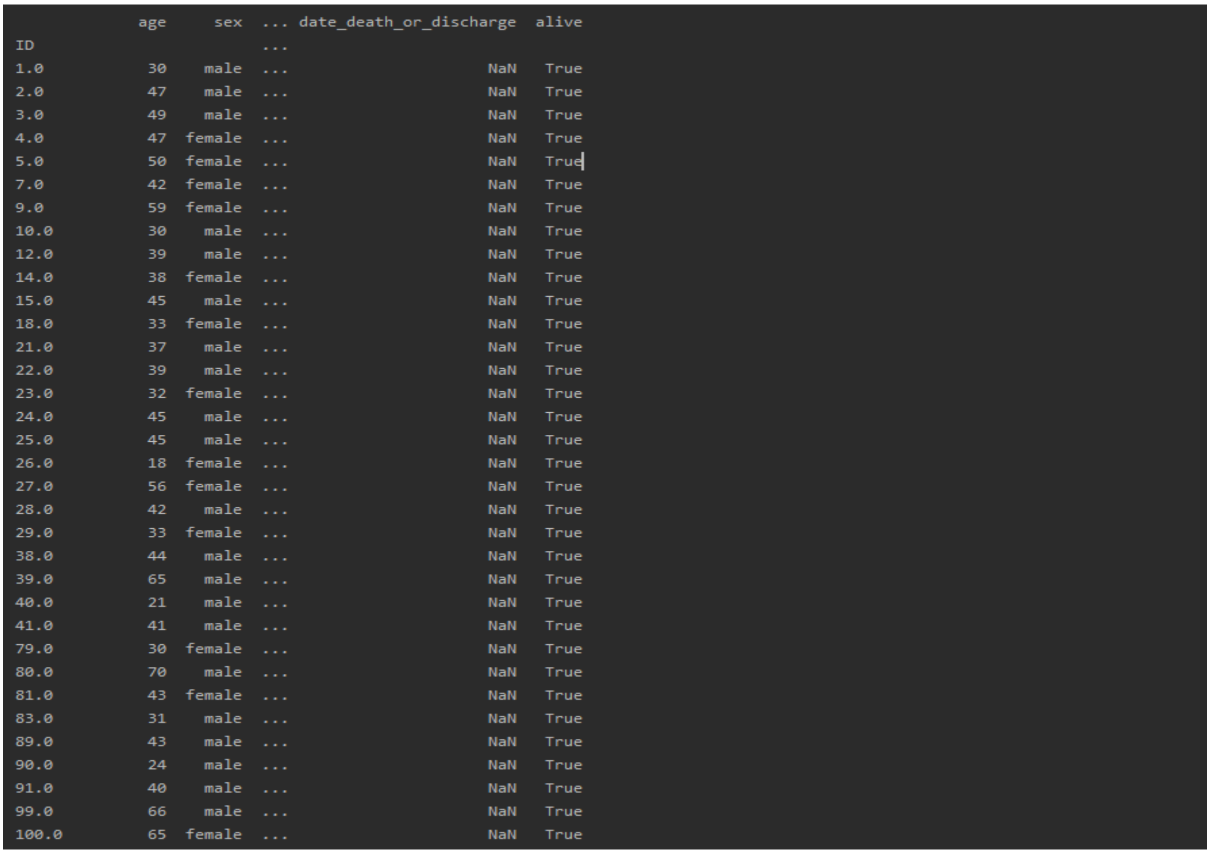
As mentioned above, the data set used in this research was compiled by a data scientist named SRK (https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset) on Kaggle based on the Github repository from Johns Hopkins University Center for Systems Science and

Engineering (https://github.com/CSSEGISandData/COVID-19). Since the new coronavirus is still outbreaking, this topic is currently very special, so I have not found a data set that is more suitable for this study or a data set that can be supplemented. Once there are such data sets, I can integrate them to the dataset I use in this study.

3.5 Format the data as required

This is the final step of data preparation. An overview on the data that will be used in this study is essential and data type should be checked.

- **The coronavirus patient data**



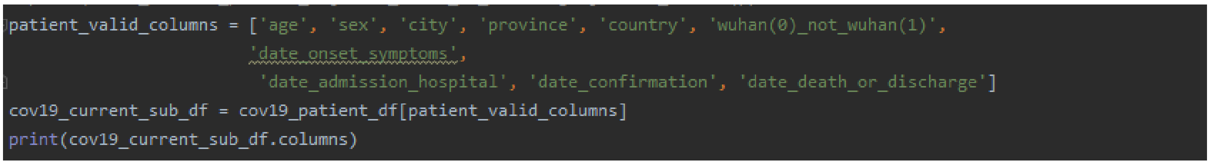
```
             age     sex  ... date_death_or_discharge  alive
ID                        ...                               
1.0           30    male  ...                     NaN   True
2.0           47    male  ...                     NaN   True
3.0           49    male  ...                     NaN   True
4.0           47  female  ...                     NaN   True
5.0           50  female  ...                     NaN   True
7.0           42  female  ...                     NaN   True
9.0           59  female  ...                     NaN   True
10.0          30    male  ...                     NaN   True
12.0          39    male  ...                     NaN   True
14.0          38  female  ...                     NaN   True
15.0          45    male  ...                     NaN   True
18.0          33  female  ...                     NaN   True
21.0          37    male  ...                     NaN   True
22.0          39    male  ...                     NaN   True
23.0          32  female  ...                     NaN   True
24.0          45    male  ...                     NaN   True
25.0          45    male  ...                     NaN   True
26.0          18  female  ...                     NaN   True
27.0          56  female  ...                     NaN   True
28.0          42    male  ...                     NaN   True
29.0          33  female  ...                     NaN   True
38.0          44    male  ...                     NaN   True
39.0          65    male  ...                     NaN   True
40.0          21    male  ...                     NaN   True
41.0          41    male  ...                     NaN   True
79.0          30  female  ...                     NaN   True
80.0          70    male  ...                     NaN   True
81.0          43  female  ...                     NaN   True
83.0          31    male  ...                     NaN   True
89.0          43    male  ...                     NaN   True
90.0          24    male  ...                     NaN   True
91.0          40    male  ...                     NaN   True
99.0          66    male  ...                     NaN   True
100.0         65  female  ...                     NaN   True
```

Figure 36 – Data inspection - patient data

4 Data Transformation

4.1 Reduce the data

Feature selection is an essential part for a data analysis process. However, for the data mining goal of this study, I am interested in finding the correlation between attributes. However, some of the columns may not an important. In this study, I will be selecting features through logical process and filter out the attributes that should not be affecting the death of patients such as news source.



```
patient_valid_columns = ['age', 'sex', 'city', 'province', 'country', 'wuhan(0)_not_wuhan(1)',
                         'date_onset_symptoms',
                          'date_admission_hospital', 'date_confirmation', 'date_death_or_discharge']
cov19_current_sub_df = cov19_patient_df[patient_valid_columns]
print(cov19_current_sub_df.columns)
```

Figure 37 -- reducing

4.2 Project the data

In statistics, data transformation is the application of a deterministic mathematical function to each point in a data set—that is, each data point zi is replaced with the transformed value yi = f(zi), where f is a function. Transforms are usually applied so that the data appear to more closely meet the assumptions of a statistical inference procedure that is to be applied, or to improve the interpretability or appearance of graphs (“Data transformation(statistics)”, n.d.).

In this study, most of the variables are categorical which are not suitable for this step.

5 Data-mining Method(s) Selection

5.1 Match and discuss the objectives of data mining to data mining methods

Within the field of machine learning, there are two main types of tasks: supervised, and unsupervised. The main difference between the two types is that supervised learning is done using ground truth, or in other words, we have prior knowledge of what the output values for our samples should be. Therefore, the goal of supervised learning is to learn a function that, given a sample of data and desired outputs, best approximates the relationship between input and output observable in the data. Unsupervised learning, on the other hand, does not have labeled outputs, so its goal is to infer the natural structure present within a set of data points (Soni, 2018).

- **Supervised learning**

Supervised learning is typically done in the context of classification, when we want to map input to output labels, or regression, when we want to map input to a continuous output. Common algorithms in supervised learning include logistic regression, naive bayes, support vector machines, artificial neural networks, and random forests. In both regression and classification, the goal is to find specific relationships or structure in the input data that allow us to effectively produce correct output data. Note that “correct” output is determined entirely from the training data, so while we do have a ground truth that our model will assume is true, it is not to say that data labels are always correct in real-world situations. Noisy, or incorrect, data labels will clearly reduce the effectiveness of your model (Soni, 2018).

- **Unsupervised learning**

The most common tasks within unsupervised learning are clustering, representation learning, and density estimation. In all of these cases, we wish to learn the inherent structure of our data without using explicitly-provided labels. Some common algorithms include k-means clustering, principal component analysis, and autoencoders. Since no labels are provided, there is no specific way to compare model performance in most unsupervised learning methods. Two common use-cases for unsupervised learning are exploratory analysis and dimensionality reduction. Unsupervised learning is very useful in exploratory analysis because it can automatically identify structure in data. For example, if an analyst were trying to segment consumers, unsupervised clustering methods would be a great starting point for their analysis. In situations where it is either impossible or impractical for a human to propose trends in the data, unsupervised learning can provide initial insights that can then be used to test individual hypotheses (Soni, 2018).

According to the data mining objectives in this study, they are:

- Find the relationship between mortality and other factors in people with new coronavirus infections.
- Produce a model that have predicting power for the mortality patients of new observations.

For the objectives, supervised learning should be used to achieve it, because in the previous stage there was a target field generated.

5.2 Select the appropriate data-mining method(s) based on discussion

Based on the different data mining methods discussed above, the data mining method chosen in this study is classification and regression under supervised learning.

For the data mining objectives, the goal is to find the relationship between mortality and other factors in people with new coronavirus infections and to produce a model with high accuracy. In previous steps, a list of potentially relevant attributes is selected, and a classification method will be applied here to model the relationship between the target attribute and other attributes.

6 Data- mining Algorithm(s) Selection

6.1 Conduct exploratory analysis and discuss

In this step, I will explore data mining algorithms. Based on the data mining objectives of this study, the data mining method will be used in this study includes the regression model and classification in supervised learning. There are a lot of algorithms available under these categories, I will now discuss some of these algorithms and discuss the suitability of them for this study:

- **Classification**
    - Decision Tree

      A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from the root to leaf represent classification rules (“Decision tree”, n.d.).

      Decision tree is one of the potential algorithms for analysis regarding the data mining goal of this study which is to find the relationship between mortality and other factors in people with new coronavirus infections. Output of a decision tree is interpretable, and the running efficiency is high. However, it is easy to overfit the data.

    - Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set (“Random forest”, n.d.).

Random forest is another potential algorithm that might work for analysis regarding the first data goal of this study. However, the running efficiency is relatively low, and the result is not stable.

- Neural Network

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of real biological neurons or an artificial neural network, for solving artificial intelligence problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1 (“Neural network”, n.d.).

Neural network is normally used when there is a categorical output variable with multiple levels. Also, the result is not interpretable and the running efficient is low and it normally overfit the data. Therefore, it might not be very suitable for this study.

- Gradient Boosting Tree

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Gradient boosting is typically used with decision trees (especially CART trees) of a fixed size as base learners. For this special case, Friedman proposes a modification to gradient boosting method which improves the quality of fit of each base learner.

- Naïve Bayes

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features (“Naive Bayes classifier”, n.d.).

Naïve Bayes classifier is normally used in tasks involving Natural Language Processing because this kind of task is normally easy to be converted to a multivariate data format and a Naïve Bayes classifier is fast and easy to construct. Therefore, it might not be very suitable for this study.

- **Regression**
  - Linear Regression

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called a simple linear regression. For more than one explanatory variable, the process is called multiple linear regression (“Linear regression”, n.d.).

Linear regression could be one of the algorithms for analysis regarding the data mining goal of this study which is to find the relationship between mortality and other factors in people with new coronavirus infections. However, it is normally used when the target field is continuous so that Logistic regression discussed below might be a better choice for this study.

  - Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labelled "0" and "1" (“Logistic regression”, n.d.).

Logistic regression is the algorithm that best suitable for the data mining goal of this study in theory because the output variable of the patient data is binary. Also, it is relatively easy to be constructed and the result is interpretable.

  - Time Series Analysis

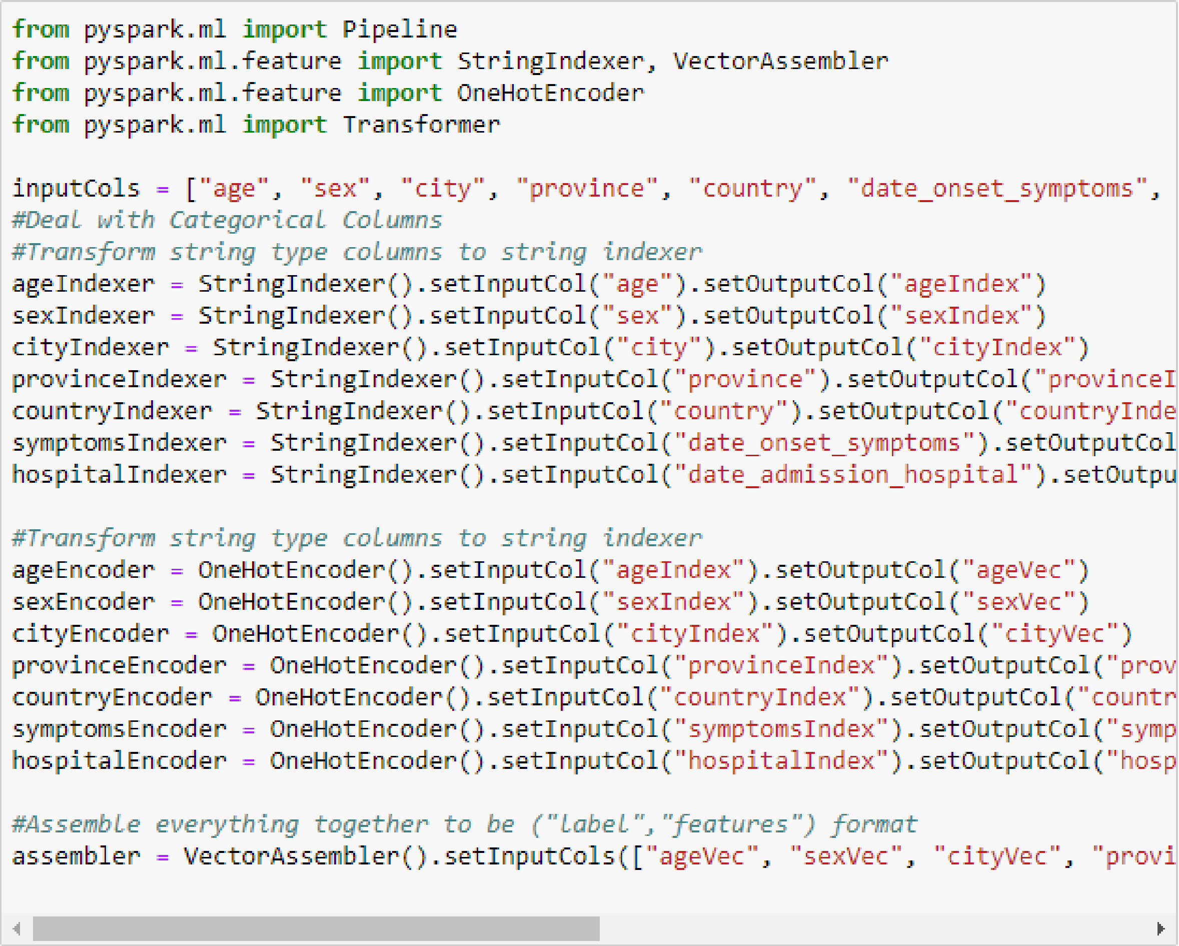
Time series analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values. While regression analysis is often employed in such a way as to test theories that the current values of one or more independent time series affect the current value of another time series, this type of analysis of time series is not called "time series analysis", which focuses on comparing values of a single time series or multiple dependent time series at different points in time. Interrupted time series analysis is the analysis of interventions on a single time series (“Time series”, n.d.).

6.2 Select data-mining algorithms based on discussion

Based on the data mining objectives and the discussion above, I will choose decision tree algorithms, logistic regression, and random forest as the data mining algorithms for this study. For the data mining goals of this study, both a decision tree algorithm and a Logistic regression will be mainly used in the analysis because they can handle binary output variable type, efficient in running time and the result is easy to interpret. The random forest algorithm also worth a try to see whether there could be a model with a better predicting power.

6.3 Build/Select appropriate model(s) and choose relevant parameter(s)

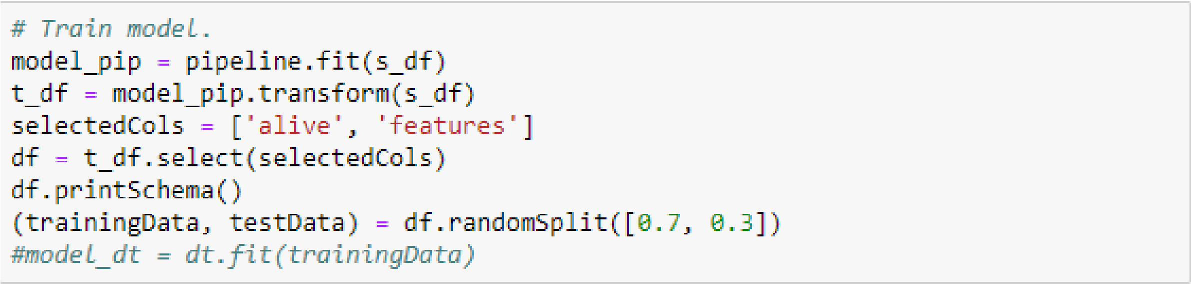
In PySpark, a dataset needs to be transformed as a "libsvm" format to be able to fitted by the tree based classifiers. The following figure shows the way of doing that:



```
from pyspark.ml import Pipeline
from pyspark.ml.feature import StringIndexer, VectorAssembler
from pyspark.ml.feature import OneHotEncoder
from pyspark.ml import Transformer

inputCols = ["age", "sex", "city", "province", "country", "date_onset_symptoms",
#Deal with Categorical Columns
#Transform string type columns to string indexer
ageIndexer = StringIndexer().setInputCol("age").setOutputCol("ageIndex")
sexIndexer = StringIndexer().setInputCol("sex").setOutputCol("sexIndex")
cityIndexer = StringIndexer().setInputCol("city").setOutputCol("cityIndex")
provinceIndexer = StringIndexer().setInputCol("province").setOutputCol("provinceI
countryIndexer = StringIndexer().setInputCol("country").setOutputCol("countryInde
symptomsIndexer = StringIndexer().setInputCol("date_onset_symptoms").setOutputCol
hospitalIndexer = StringIndexer().setInputCol("date_admission_hospital").setOutpu

#Transform string type columns to string indexer
ageEncoder = OneHotEncoder().setInputCol("ageIndex").setOutputCol("ageVec")
sexEncoder = OneHotEncoder().setInputCol("sexIndex").setOutputCol("sexVec")
cityEncoder = OneHotEncoder().setInputCol("cityIndex").setOutputCol("cityVec")
provinceEncoder = OneHotEncoder().setInputCol("provinceIndex").setOutputCol("prov
countryEncoder = OneHotEncoder().setInputCol("countryIndex").setOutputCol("countr
symptomsEncoder = OneHotEncoder().setInputCol("symptomsIndex").setOutputCol("symp
hospitalEncoder = OneHotEncoder().setInputCol("hospitalIndex").setOutputCol("hosp

#Assemble everything together to be ("label","features") format
assembler = VectorAssembler().setInputCols(["ageVec", "sexVec", "cityVec", "provi
```

Figure 38 -- model fitting

After building the "indexers" and "Encoders", they would be passed in a pipeline model to transform the data:

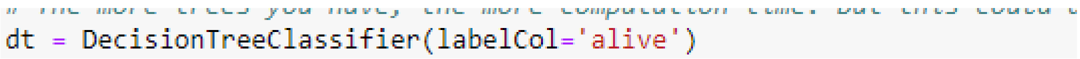


```
# Train model.
model_pip = pipeline.fit(s_df)
t_df = model_pip.transform(s_df)
selectedCols = ['alive', 'features']
df = t_df.select(selectedCols)
df.printSchema()
(trainingData, testData) = df.randomSplit([0.7, 0.3])
#model_dt = dt.fit(trainingData)
```

Figure 39 -- model fitting

- **Decision Tree**

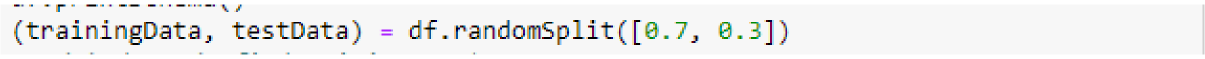
The first step will be to build a Decision Tree classifier:



```
dt = DecisionTreeClassifier(labelCol='alive')
```

Figure 40 -- model fitting

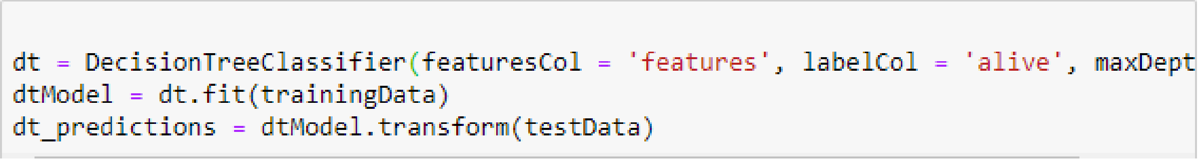
To avoid overfitting the data, a cross-validation with 70% of data as training set and 30% of data as test set is used. To fit the model on training set and produce predictions for test set.



```
(trainingData, testData) = df.randomSplit([0.7, 0.3])
```

Figure 41 -- model fitting

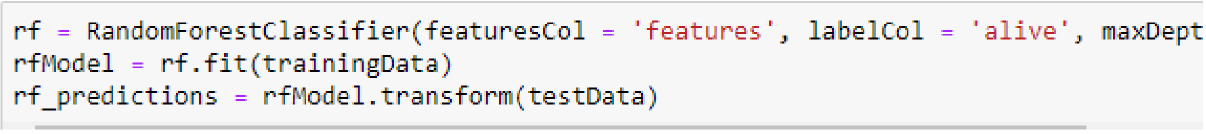
The model is fitted by the following:



```
dt = DecisionTreeClassifier(featuresCol = 'features', labelCol = 'alive', maxDept
dtModel = dt.fit(trainingData)
dt_predictions = dtModel.transform(testData)
```

Figure 42 -- model fitting

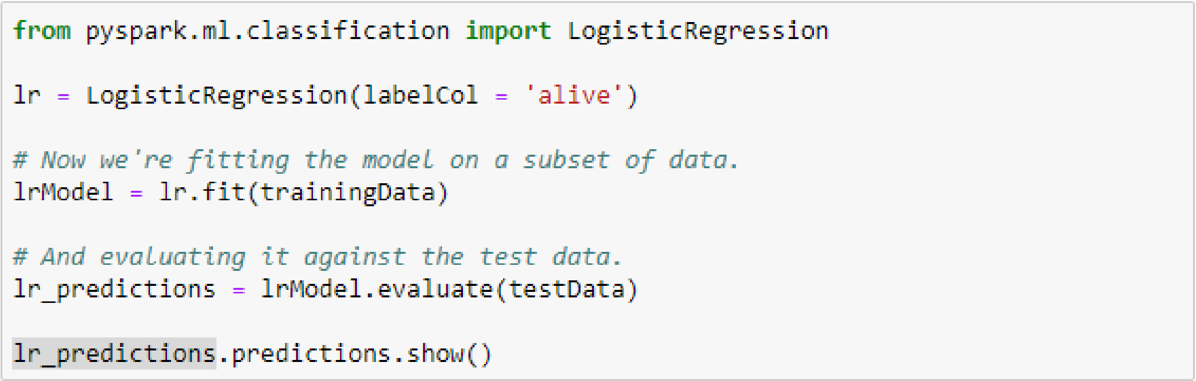
- **Random Forest**



```
rf = RandomForestClassifier(featuresCol = 'features', labelCol = 'alive', maxDept
rfModel = rf.fit(trainingData)
rf_predictions = rfModel.transform(testData)
```

Figure 43 -- model fitting

- **Logistic Regression**



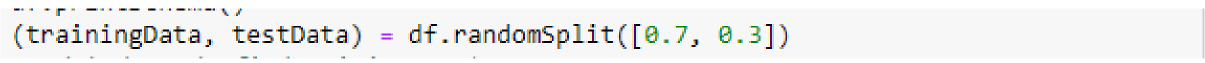
```
from pyspark.ml.classification import LogisticRegression

lr = LogisticRegression(labelCol = 'alive')

# Now we're fitting the model on a subset of data.
lrModel = lr.fit(trainingData)

# And evaluating it against the test data.
lr_predictions = lrModel.evaluate(testData)

lr_predictions.predictions.show()
```

Figure 44 -- model fitting

7 Data Mining

7.1 Create and justify test designs

As stated above, to avoid overfitting to the data, a cross-validation test is designed to split the data into a 70/30 radio training and test data for the analysis of the data mining goal of this study.



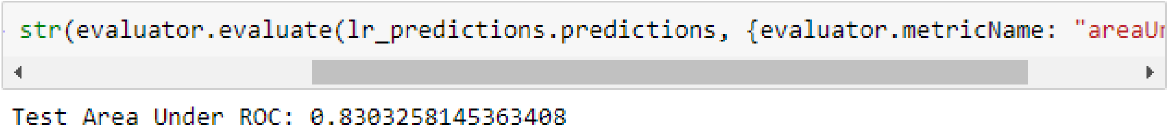
```
(trainingData, testData) = df.randomSplit([0.7, 0.3])
```

Figure 45 -- model evaluation

7.2 Conduct data mining - classify, regress, cluster

For the data mining objective which is to find the relationship between mortality and other factors in people with new coronavirus infections:

- Logistic regression



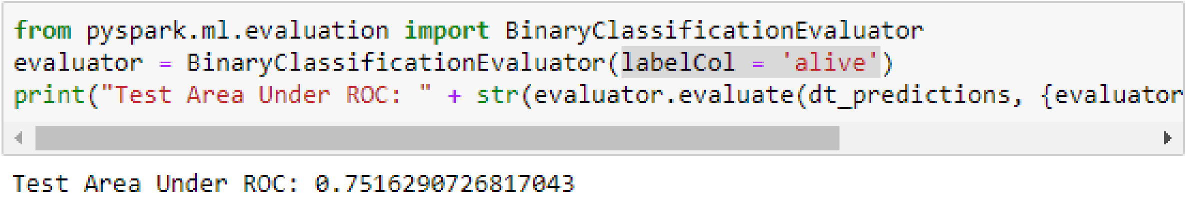
```
str(evaluator.evaluate(lr_predictions.predictions, {evaluator.metricName: "areaU
```

```
Test Area Under ROC: 0.8303258145363408
```

Figure 46 -- model evaluation

The result from the Logistic regression model is about 76% which is not very impressing in this case.

- Decision tree



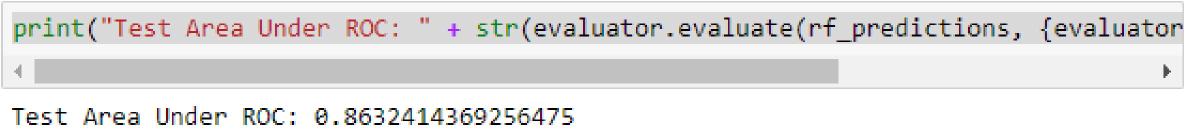
```
from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator = BinaryClassificationEvaluator(labelCol = 'alive')
print("Test Area Under ROC: " + str(evaluator.evaluate(dt_predictions, {evaluator
```

```
Test Area Under ROC: 0.7516290726817043
```

Figure 47 -- model evaluation

The result from the Decision Tree model is not very impressive. The model achieved over 75% accuracy on the test data.

- Random forest



```
print("Test Area Under ROC: " + str(evaluator.evaluate(rf_predictions, {evaluator
```

```
Test Area Under ROC: 0.8632414369256475
```

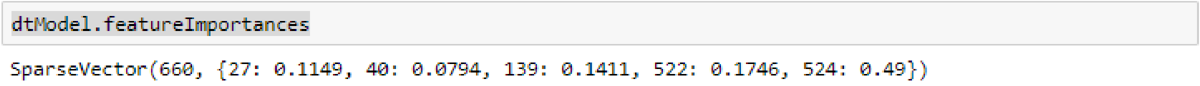
Figure 48 -- model evaluation

The result from the Random forest model is even better. The model achieved over 86% accuracy on the test data.

Out of the three algorithms, the Random forest model seems to outperform the other two.

7.3 Search for patterns

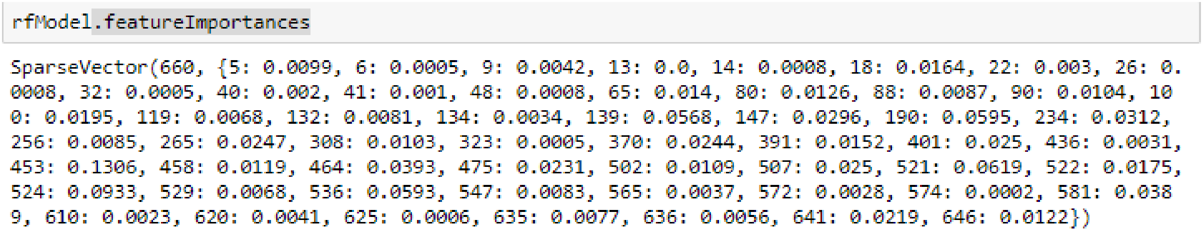
For the data mining objective which is to find the relationship between mortality and other factors in people with new coronavirus infections, the calculated predictor importance from the two analyses results of the Decision Tree model and the Random forest model are showed below:



```
dtModel.featureImportances
```

```
SparseVector(660, {27: 0.1149, 40: 0.0794, 139: 0.1411, 522: 0.1746, 524: 0.49})
```

Figure 49 -- model evaluation



```
rfModel.featureImportances
```

```
SparseVector(660, {5: 0.0099, 6: 0.0005, 9: 0.0042, 13: 0.0, 14: 0.0008, 18: 0.0164, 22: 0.003, 26: 0.
0008, 32: 0.0005, 40: 0.002, 41: 0.001, 48: 0.0008, 65: 0.014, 80: 0.0126, 88: 0.0087, 90: 0.0104, 10
0: 0.0195, 119: 0.0068, 132: 0.0081, 134: 0.0034, 139: 0.0568, 147: 0.0296, 190: 0.0595, 234: 0.0312,
256: 0.0085, 265: 0.0247, 308: 0.0103, 323: 0.0005, 370: 0.0244, 391: 0.0152, 401: 0.025, 436: 0.0031,
453: 0.1306, 458: 0.0119, 464: 0.0393, 475: 0.0231, 502: 0.0109, 507: 0.025, 521: 0.0619, 522: 0.0175,
524: 0.0933, 529: 0.0068, 536: 0.0593, 547: 0.0083, 565: 0.0037, 572: 0.0028, 574: 0.0002, 581: 0.038
9, 610: 0.0023, 620: 0.0041, 625: 0.0006, 635: 0.0077, 636: 0.0056, 641: 0.0219, 646: 0.0122})
```

Figure 50 -- model evaluation

One of the drawbacks for PySpark is that tree based model like Decision tree model cannot be easily viewed like Sklearn.

8 Interpretation

8.1 Study and discuss the mined patterns

Choosing the right model type is one of the key factors for successful data mining. In the previous steps, the Logistic regression model and the random forest model showed that they are better models to be used than the Decision tree model for the data mining goals of this study, which is to find the relationship between the mortality of people with new coronavirus infection and other factors and to produce reliable predictions. After that, the prediction accuracies of these two models are impressive. However, this is far from the end of these data mining goals, and there is still much room for improvement. For example, if mentioned earlier, the data on new coronavirus are very special and are still being continuously updated. Once there is better quality data in the future that can replace the data I currently use, the quality of this research can be improved.

8.2 Visualize the data, results, models, and patterns

Data visualization is another important step in data mining. Generally, after visualizing the data, you can get some important information, such as whether the data needs to be transformed or how the overall trend of the data changes.

- Attribute distribution

A good way to understand the data is to visualize the attributes. In the previous steps, if you visualize the attribute distribution in advance, it is much easier to clean up the data. In addition, data visualization provides a very intuitive way to know the impact of data cleansing on the distribution of this attribute.

The distribution graph for the 'sex' attribute before cleaning:

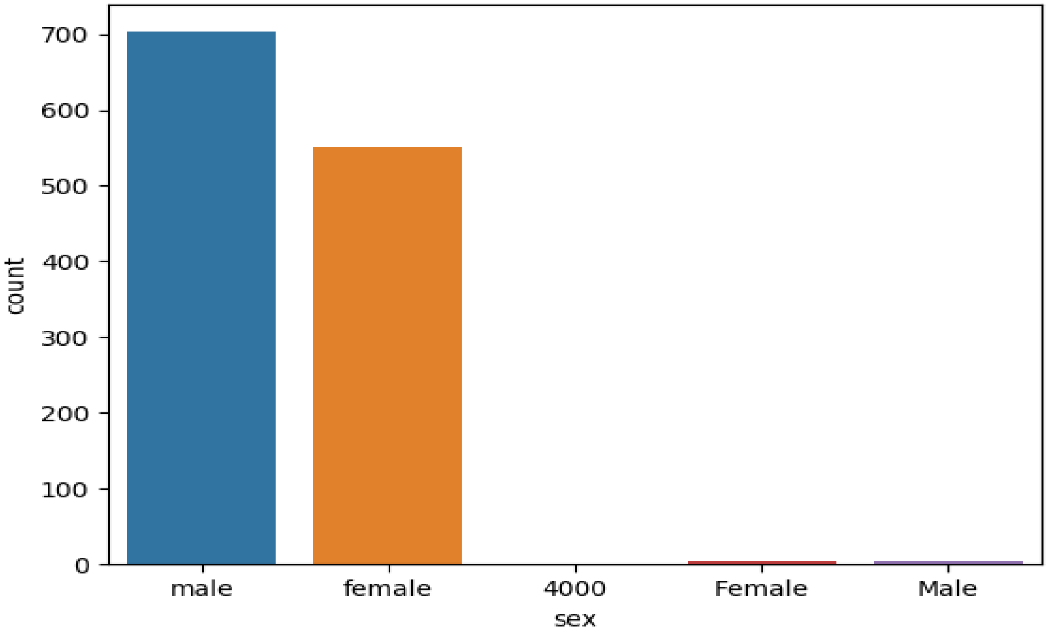


Figure 51 -- distribution graph

The distribution graph for the 'sex' attribute after cleaning:

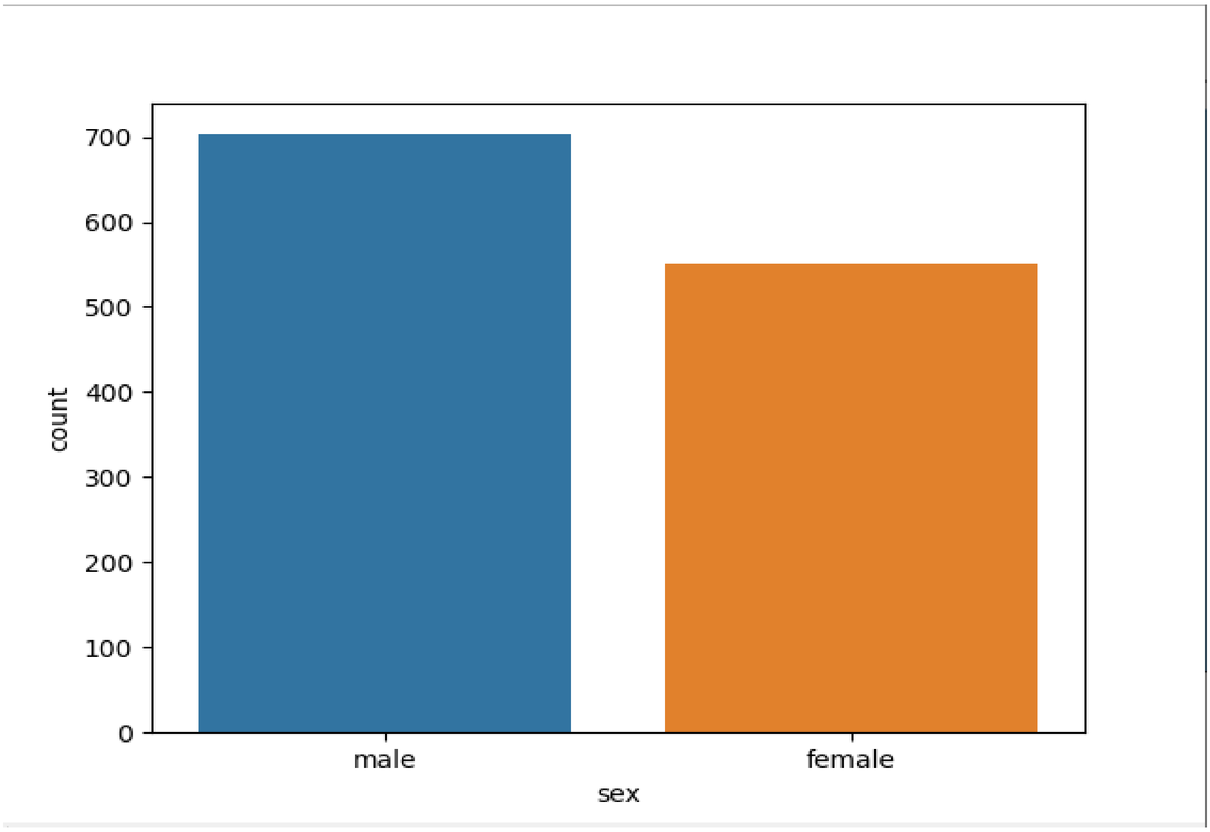


Figure 52 -- distribution graph

The distribution graph for the ‘age’ attribute before cleaning:

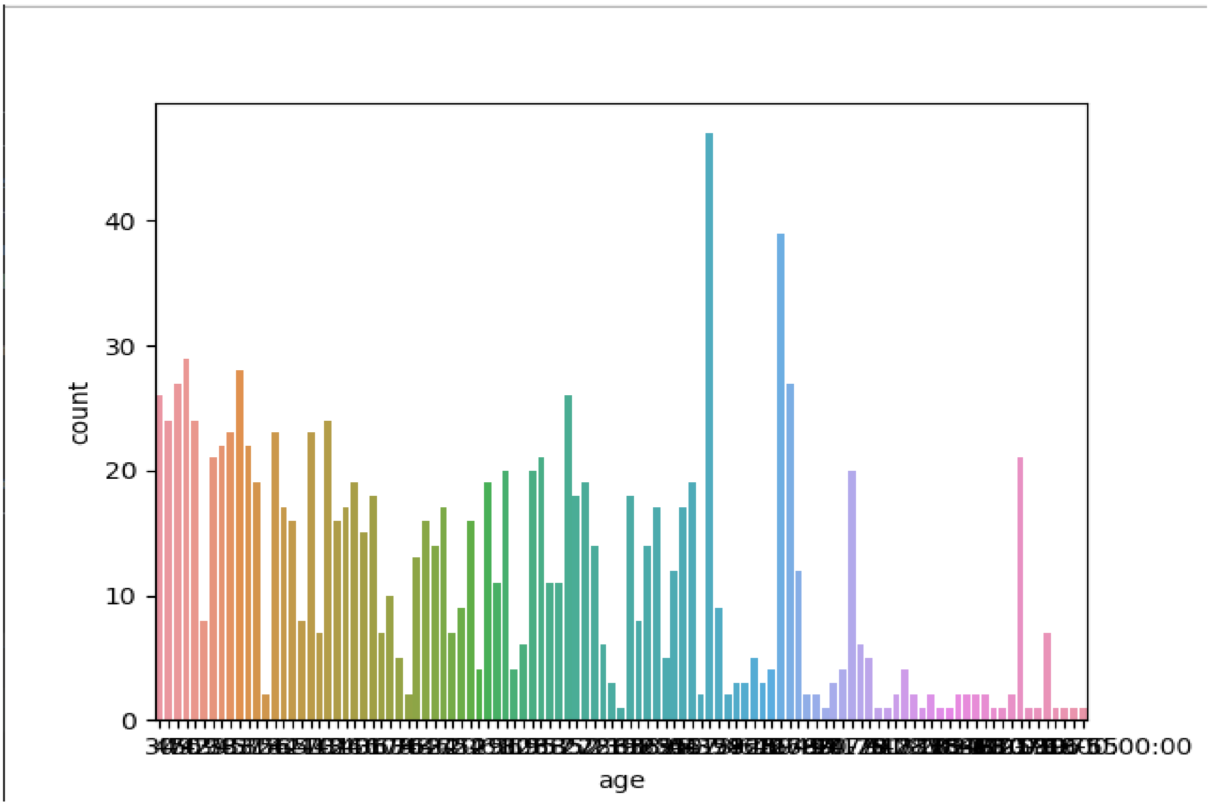


Figure 53 -- distribution graph

The distribution graph for the ‘age’ attribute after cleaning:

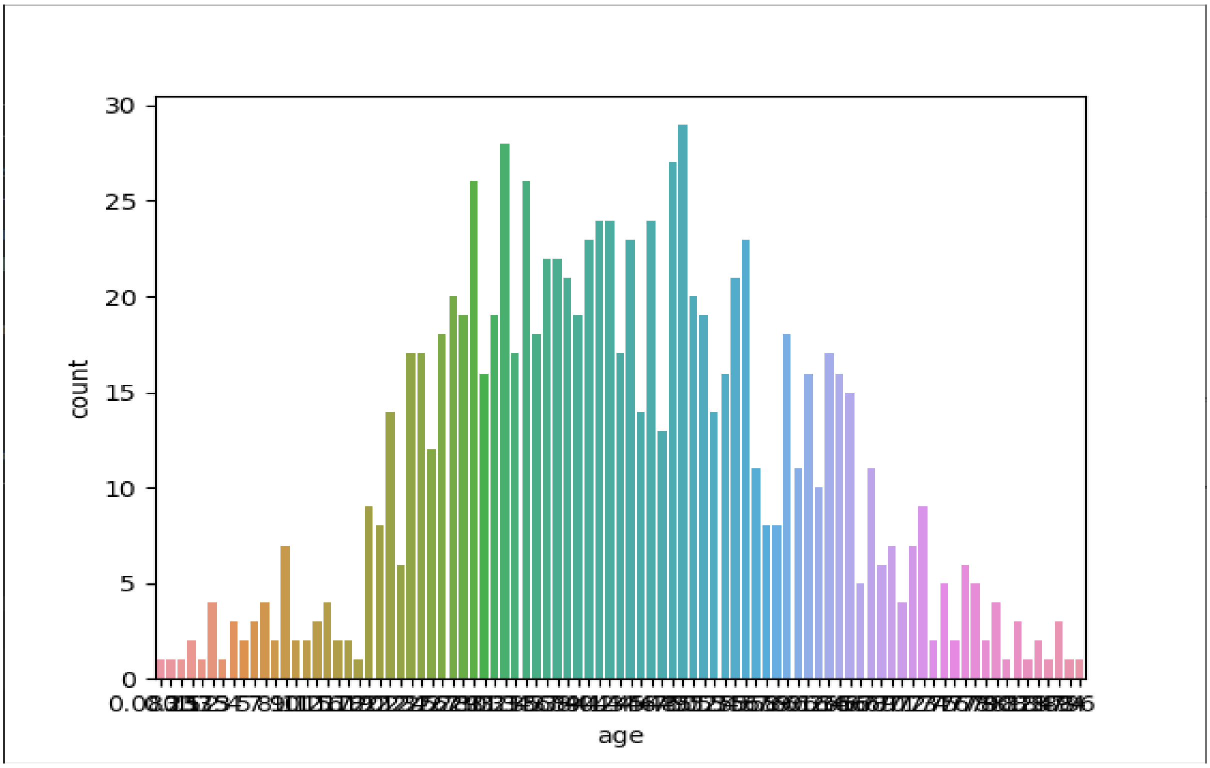


Figure 54 -- distribution graph

The distribution graph for the target attribute ‘alive’ before upsampling:

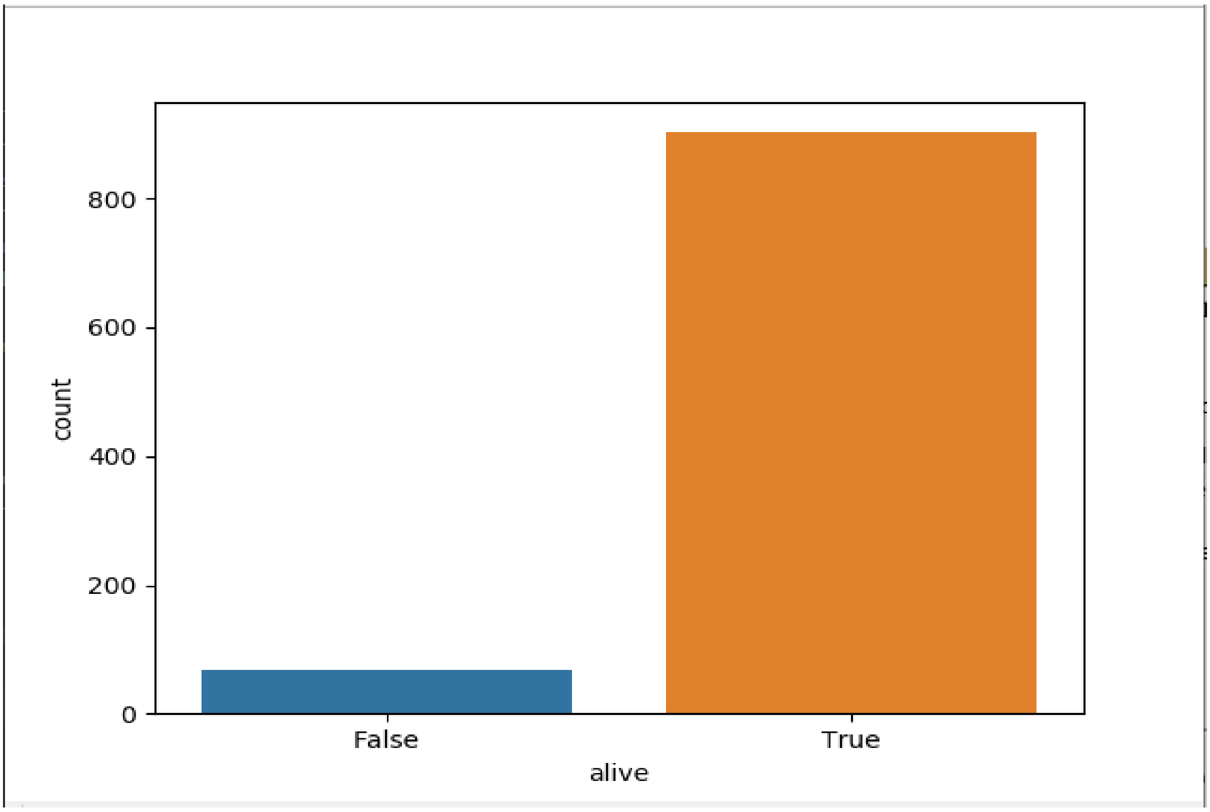


Figure 55 -- distribution graph

The distribution graph for the target attribute ‘alive’ after upsampling:

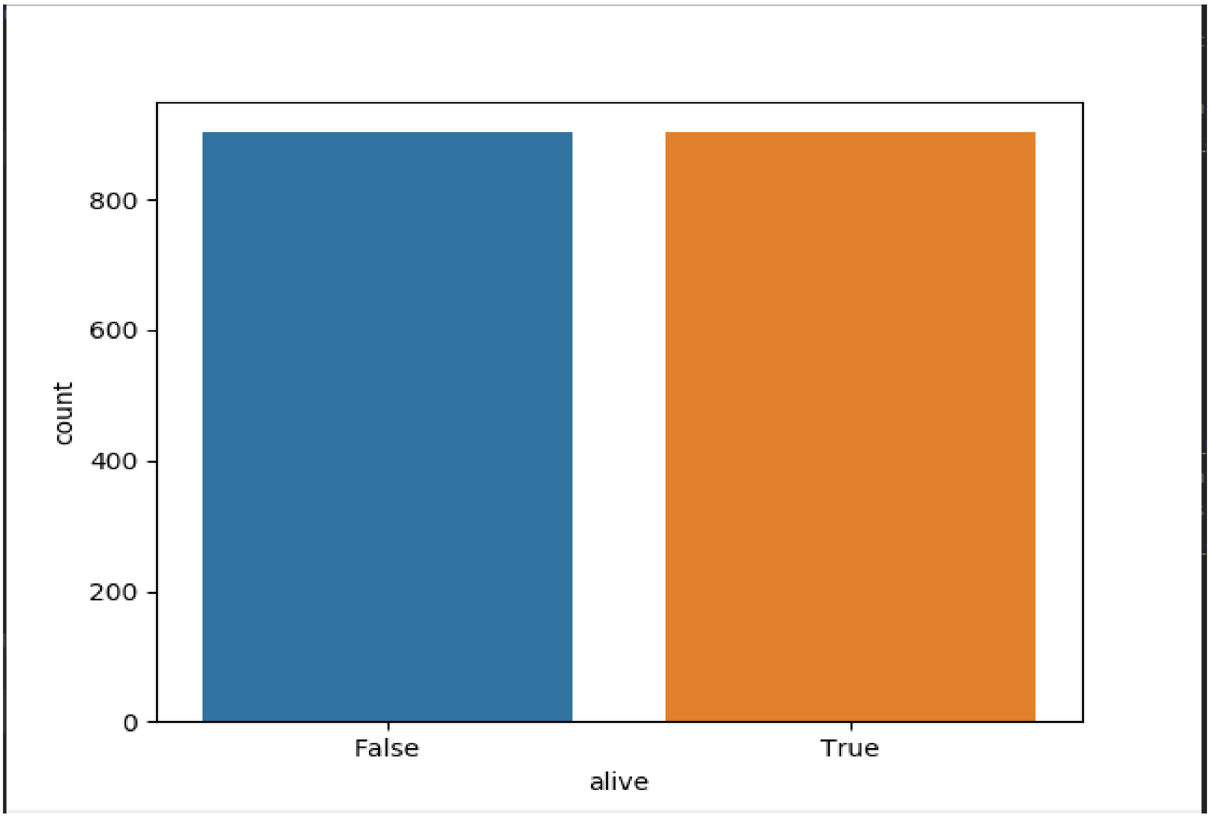


Figure 56 -- distribution graph

- Predictor importance

As mentioned earlier, the importance of predictors given by the two models is different. The bar plot of predictor importance from the decision tree model：

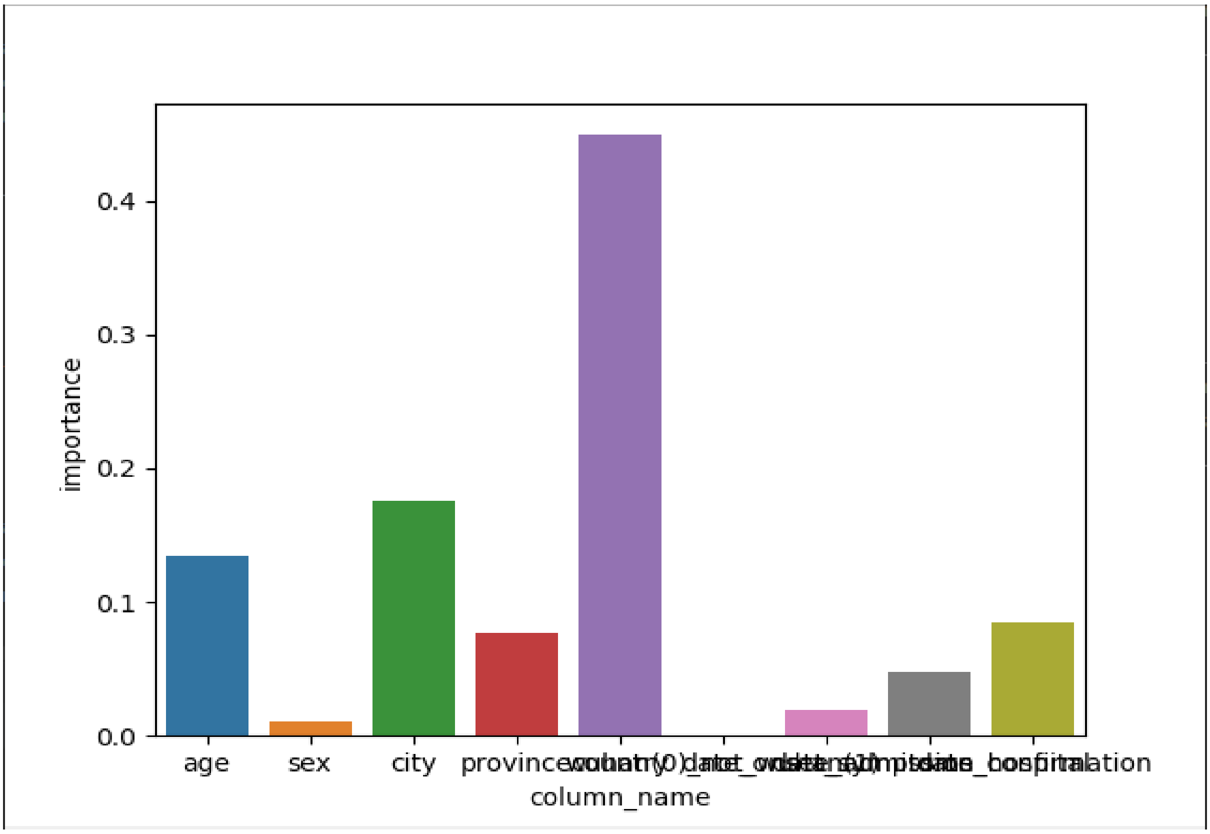


Figure 57 – Predictor importance – Decision tree model

The bar plot of predictor importance from the random forest model：

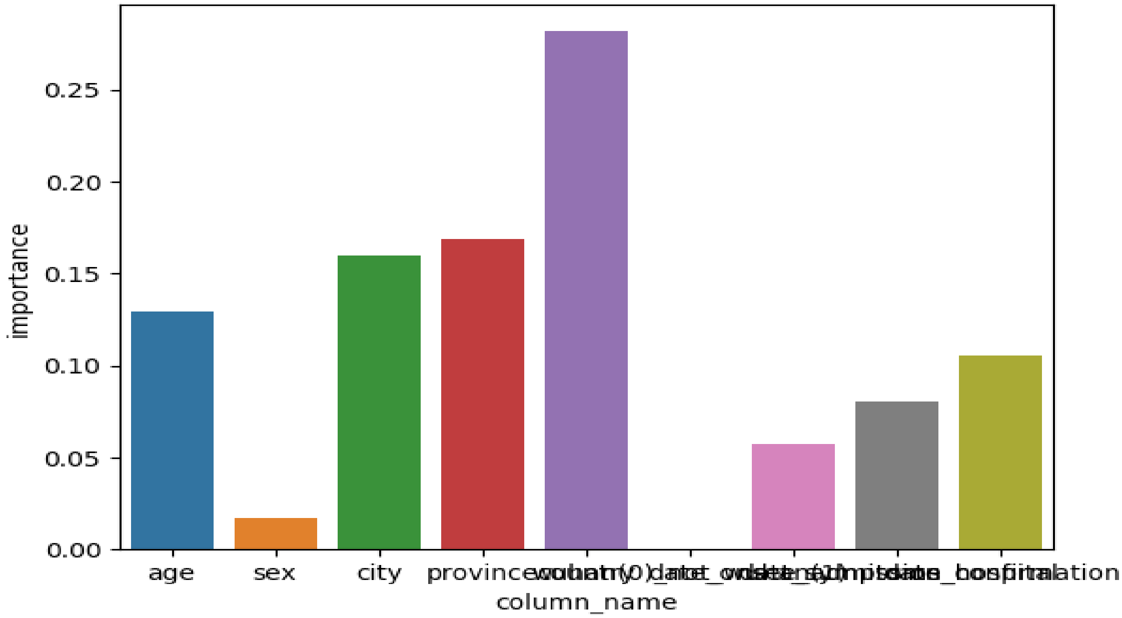


Figure 58 – Predictor importance – Random forest model

- Model results

But as mentioned before, this is not the perfect model for this data mining goal. In fact, usually perfect models are impossible to discover.

8.3 Interpret the results, models, and patterns

From the result of the decision tree model, it is very difficult to directly interpret the results because the generated decision tree is too large and takes a long time to understand deeply. On the other hand, since this is one of the advantages of the decision tree algorithm and random forest algorithm, it is more straightforward to explain and discuss the importance of predictors from the model. The results show that the important predictors are age, date of confirmation, country and city, which is consistent with common sense. There is a lot of news and research that shows that the mortality rate of people with new coronavirus infection has a great relationship with age. In this outbreak, older people are more vulnerable than young people for various reasons, such as resistance. And the date and location of the patient's confirmation are also in line with common sense because the amount of medical resources in each country is different and patients who are identified earlier are more likely to get medical resources.

8.4 Assess and evaluate results, models and patterns

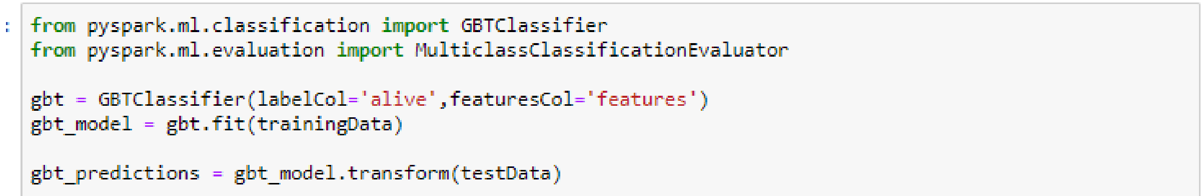
After the entire data mining process, the results can be said to be quite surprising. The study aimed to find the relationship between mortality and other factors in people with new coronavirus infections and the result is pretty impressing even though the direct interpretation is hard. Also, the study aimed to produce reliable prediction for new observations and the accuracies showed that the models from this study is successful. However, as stated before, there is still a lot of improvement that can be made. Overall, I am satisfied with the result of this study.

8.5 Iterate prior steps (1 - 7) as required

1. Business understanding
   The current outbreak of new coronavirus is almost one of the hottest topics in the world. Regarding the data mining goals of this study, patient's mortality may also be related to other factors such as its own chronic medical history, etc. This may be the direction that this study can be extended. With more and more research on new coronaviruses, more people understand this. However, there are still many unsolved mysteries. For example, how they appeared or whether they were created manually. This research can be further expanded to address other data mining goals with appropriate settings.
2. Data understanding
   Repeating this step may not make much sense for this study.
3. Data preparation
   As new coronavirus outbreaks are still ongoing, the data sources used in this study are constantly being updated. So far, the data sources used in this study are still being updated daily.
4. Data transformation
   Repeating this step may not make much sense for this study.
5. Data mining method selection
   Repeating this step may not make much sense for this study.
6. Data mining algorithm selection
   In the previous steps, the model using the decision tree algorithm gave very good results. I think that the model using the gradient boosting algorithm is also very worth trying because

gradient boosting is the algorithm that uses the different technique and often gives out a good accuracy.

6.3 Build/Select appropriate model(s) and choose relevant parameter(s)



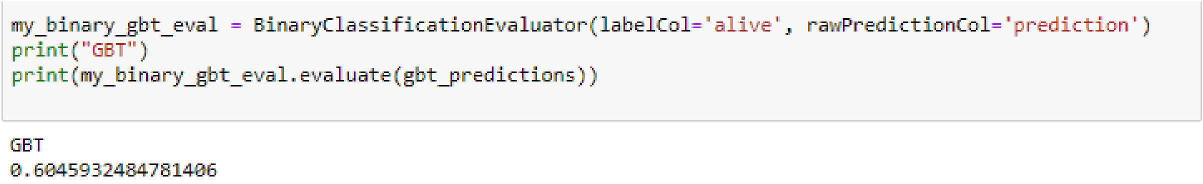
```
from pyspark.ml.classification import GBTClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

gbt = GBTClassifier(labelCol='alive',featuresCol='features')
gbt_model = gbt.fit(trainingData)

gbt_predictions = gbt_model.transform(testData)
```

Figure 59 -- model fitting

7. Data mining

7.2 Conduct data mining - classify, regress, cluster

- Gradient Boosting tree



```
my_binary_gbt_eval = BinaryClassificationEvaluator(labelCol='alive', rawPredictionCol='prediction')
print("GBT")
print(my_binary_gbt_eval.evaluate(gbt_predictions))
```

```
GBT
0.6045932484781406
```

Figure 60 -- model evaluation

The result from the Gradient Boosting tree model is worse than the result from any of the model used in the previous sections. The model achieved only 60% accuracy on the test data. The reason of that might be related to the hyperparameter settings or data types of the dataset used.

8. Interpretation

8.1 Study and discuss the mined patterns

Although gradient boosting algorithms are generally regarded as algorithms that can provide more accurate results, the above results do not support this view. One reason may be because the data used in this study is more categorical variables, so it seems to be more suitable for decision tree algorithm or random forest algorithm.

8.3 Interpret the results, models, and patterns

One of the biggest disadvantages of gradient boosting algorithms is the inability to interpret the results. This is one of the reasons why I think it is not suitable for this study.

8.4 Assess and evaluate results, models and patterns

In terms of accuracy, the model using the gradient boosting is not a very good model, with only 60% accuracy. It is inferior to the previous model using decision tree algorithm or random forest algorithm.

Disclaimer:

STUDENT ID: 8279325
Name: Johnson Zhou
UPI: zzho612

I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright. (See: https://www.auckland.ac.nz/en/students/forms-policies-and-guidelines/student-policies-and-guidelines/academic-integrity-copyright.html). I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data.

References:

1. United Nation 17 Sustainable Goals.
2. IBM SPSS Modeler CRISP-DM Guide.
3. World Health Organization Coronavirus disease (COVID-19) outbreak.
4. 2019–20 coronavirus outbreak. (n.d.). In Wikipedia. Retrieved March 12, 2020, from https://en.wikipedia.org/wiki/2019%E2%80%9320_coronavirus_outbreak
5. SRK. (n.d.). Novel Corona Virus 2019 Dataset. (2019). Retrieved March 20, 2020, from https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset
6. 2019 Novel Coronavirus COVID-19 (2019-nCoV) Data Repository by Johns Hopkins CSSE. (2019). Retrieved May 1, 2009, from https://github.com/CSSEGISandData/COVID-19
7. Coronavirus (COVID-19) Mortality Rate. (2020, March 5). Worldometers. https://www.worldometers.info/coronavirus/coronavirus-death-rate/
8. Data transformation (statistics). (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Data_transformation_(statistics)
9. Soni, D. 2018. Supervised vs. Unsupervised Learning. Toward data science. Retrieved from https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d
10. Decision tree. (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Decision_tree
11. Random forest. (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Random_forest
12. Neural network. (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Neural_network
13. Naive Bayes classifier. (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Naive_Bayes_classifier
14. Linear regression. (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Linear_regression
15. Logistic regression. (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Logistic_regression
16. Time series. (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Time_series#Models
17. Lift (data mining). (n.d.). In Wikipedia. Retrieved March 29, 2020, from https://en.wikipedia.org/wiki/Lift_(data_mining)