Rituparna Das

Predicting Covid Mortality with Machine Learning

- Download the Covid-19 dataset from the URL:
  https://www.kaggle.com/kimjihoo/coronavirusdataset
  The dataset is online and publicly available with a free Kaggle account.
  Clean the dataset for prediction
- This dataset was released by Korea Centers for Disease Control and Prevention, and it contains information the Covid-19 cases in South Korea. The aim is to predict mortality among confirmed CoVID-19 patients in South Korea using logistic regression and support vector machines.

(1) Use %80 of your data as the training data and %20 of your data as the test data., and you can train only one model (either logistic regression orsupport vector machines). As a hint, you could follow a similar approach to the following paper (in terms of choosing the predictors etc.):

[Predicting CoVID-19 community mortality risk using machine learning and development of an online prognostic tool, 2020] by Das et al.

However, this paper is given as an example, you do not have to follow it. Other approaches are also welcome.

(2) Explain your reasoning and models. Visualize your classifiers. There is no unique way to visualize a classifier, you can use your imagination and provide plots that are helpful to understand the decision boundary. Make a table that summarizes your classification errors for the model(s) you trained.

COVID-19 is still one of the most serious threats to humanity. Many academics from all over the world are working hard to find a way to counter the small, invisible threat. Parallel to medical science, computer technology is making a significant contribution to COVID-19 research by speeding up vital data processing. Various machine learning-based classifiers and predictive models have also been presented, which can aid in the early detection of disease. In this research, we present a COVID-19 patient categorization model based on machine learning and using logistic regression.

The dataset used for the model has been collected from Kaggle

and we are achieving 92% accuracy.

Code:

```
> setwd("~/R")

> df<-read.csv("PatientInfo.csv")

> head(df)
```

```
patient_id    sex age country province        city       infection_case infected_by
1      1e+09   male 50s   Korea    Seoul  Gangseo-gu      overseas inflow
2      1e+09   male 30s   Korea    Seoul Jungnang-gu      overseas inflow
3      1e+09   male 50s   Korea    Seoul   Jongno-gu contact with patient
2002000001
4      1e+09   male 20s   Korea    Seoul     Mapo-gu      overseas inflow
```

```
5      1e+09 female 20s   Korea    Seoul Seongbuk-gu contact with patient
1000000002

6      1e+09 female 50s   Korea    Seoul   Jongno-gu contact with patient
1000000003

contact_number symptom_onset_date confirmed_date released_date deceased_date
state

1             75         2020-01-22     2020-01-23    2020-02-05
released

2             31                        2020-01-30    2020-03-02
released

3             17                        2020-01-30    2020-02-19
released

4              9         2020-01-26     2020-01-30    2020-02-15
released

5              2                        2020-01-31    2020-02-24
released

6             43                        2020-01-31    2020-02-19
released
```

> colSums(is.na(df))

patient_id sex age country province

0 0 0 0 0

city infection_case infected_by contact_number symptom_onset_date

0 0 0 0 0

confirmed_date released_date deceased_date state

0 0 0 0

> summary(df)

```
patient_id            sex                 age              country
Min.   :1.000e+09   Length:5165        Length:5165        Length:5165
1st Qu.:1.000e+09   Class :character   Class :character   Class :character
Median :2.000e+09   Mode  :character   Mode  :character   Mode  :character
Mean   :2.864e+09
3rd Qu.:6.001e+09
Max.   :7.000e+09
province             city           infection_case     infected_by
Length:5165        Length:5165        Length:5165        Length:5165
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
contact_number     symptom_onset_date confirmed_date     released_date
Length:5165        Length:5165        Length:5165        Length:5165
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character



deceased_date         state
Length:5165        Length:5165
Class :character   Class :character
Mode  :character   Mode  :character
```

> str(df)

```
'data.frame':     5165 obs. of  14 variables:
  $ patient_id        : num  1e+09 1e+09 1e+09 1e+09 1e+09 ...
$ sex               : chr  "male" "male" "male" "male" ...
$ age               : chr  "50s" "30s" "50s" "20s" ...
$ country           : chr  "Korea" "Korea" "Korea" "Korea" ...
$ province          : chr  "Seoul" "Seoul" "Seoul" "Seoul" ...
$ city              : chr  "Gangseo-gu" "Jungnang-gu" "Jongno-gu" "Mapo-gu"
...
$ infection_case    : chr  "overseas inflow" "overseas inflow" "contact
with patient" "overseas inflow" ...
$ infected_by       : chr  "" "" "2002000001" "" ...
$ contact_number    : chr  "75" "31" "17" "9" ...
$ symptom_onset_date: chr  "2020-01-22" "" "" "2020-01-26" ...
$ confirmed_date    : chr  "2020-01-23" "2020-01-30" "2020-01-30" "2020-01-
30" ...
$ released_date     : chr  "2020-02-05" "2020-03-02" "2020-02-19" "2020-02-
15" ...
$ deceased_date     : chr  "" "" "" "" ...
$ state             : chr  "released" "released" "released" "released" ...
```

> library(plyr)

> df$state <- revalue(df$state, c("released"=1))

```
> df$state <- revalue(df$state, c("deceased"=0))

> df$state <- revalue(df$state, c("isolated"=0))

> head(df)

patient_id    sex age country province        city       infection_case
infected_by

1      1e+09   male 50s   Korea    Seoul  Gangseo-gu      overseas inflow

2      1e+09   male 30s   Korea    Seoul Jungnang-gu      overseas inflow

3      1e+09   male 50s   Korea    Seoul   Jongno-gu contact with patient
2002000001

4      1e+09   male 20s   Korea    Seoul     Mapo-gu      overseas inflow

5      1e+09 female 20s   Korea    Seoul Seongbuk-gu contact with patient
1000000002

6      1e+09 female 50s   Korea    Seoul   Jongno-gu contact with patient
1000000003

contact_number symptom_onset_date confirmed_date released_date
deceased_date state

1             75         2020-01-22     2020-01-23    2020-02-05
1

2             31                        2020-01-30    2020-03-02
1

3             17                        2020-01-30    2020-02-19
1

4              9         2020-01-26     2020-01-30    2020-02-15
1

5              2                        2020-01-31    2020-02-24
1

6             43                        2020-01-31    2020-02-19
1

> str(df)

'data.frame':     5165 obs. of  14 variables:

  $ patient_id        : num  1e+09 1e+09 1e+09 1e+09 1e+09 ...

$ sex               : chr  "male" "male" "male" "male" ...

$ age               : chr  "50s" "30s" "50s" "20s" ...

$ country           : chr  "Korea" "Korea" "Korea" "Korea" ...

$ province          : chr  "Seoul" "Seoul" "Seoul" "Seoul" ...

$ city              : chr  "Gangseo-gu" "Jungnang-gu" "Jongno-gu" "Mapo-gu"
...

$ infection_case    : chr  "overseas inflow" "overseas inflow" "contact
with patient" "overseas inflow" ...
```

```
$ infected_by       : chr  "" "" "2002000001" "" ...
$ contact_number    : chr  "75" "31" "17" "9" ...
$ symptom_onset_date: chr  "2020-01-22" "" "" "2020-01-26" ...
$ confirmed_date    : chr  "2020-01-23" "2020-01-30" "2020-01-30" "2020-01-
30" ...
$ released_date     : chr  "2020-02-05" "2020-03-02" "2020-02-19" "2020-02-
15" ...
$ deceased_date     : chr  "" "" "" "" ...
$ state             : chr  "1" "1" "1" "1" ...
```

Next step for the data processing is the dividing the dataset in training set and testing set. In our model 80% data has been used for training and remaining 20% has been used for testing the model.

```
> library(caTools)
> library(caret)
> df$state <- as.numeric(df$state)
> set.seed(123)
> samp <- createDataPartition(df$state, p = 0.8, list = FALSE)
> training <- df[samp,]
> testing <- df[-samp,]
> lm1 <- glm(state~(sex+age+country+province+city+infection_case), data = training, family = "binomial")
> summary(lm1)
>summary(lm1)$coefficient
```

```
predict <- predict(lm1, type = 'response')
```

Confusion matrix has been computed for analyzing the system performance.

```
#confusion matrix
table(training$state, predict > 0.5)
```

```
##
##      FALSE TRUE
##    0  1596  195
##    1   147 2194
```

```
> library(ROCR)
```

```
> ROCRpred <- prediction(predict, training$state)
```

```
> ROCRperf <- performance(ROCRpred, 'tpr','fpr')
> plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```