Predicting COVID death determinants

Liu Chi Yun, Yunting Chiu

2021/4/19

Introduction

Here we have Covid-19 dataset included 12 variables and 51 observations. We will use regression trees methods to predict COVID death determinants.

Covid Data

```
library(tidyverse)
library(rpart)
library(rpart.plot)
covid_df_train <- read_csv("./data/covidForRegression.csv")
covid_df_train
```

```
## # A tibble: 51 x 12
##    confirmed deaths American_Indian~ Asian_alone Black_or_Africa~
##        <dbl>  <dbl>            <dbl>       <dbl>            <dbl>
##  1    436087   6486            23265       66129          1319551
##  2     52926    254           115544       43678            22551
##  3    708041  12001           332273      241721           343729
##  4    281382   4549            17216       46078           467468
##  5   3147207  36615           321112     5865435          2282144
##  6    383008   5462            57578      188461           240538
##  7    237815   6819             9052      166393           396745
##  8     73233   1252             4353       36592           219418
##  9     34905    867             1886       28722           320704
## 10   1627603  25011            59558      599799          3441062
## # ... with 41 more rows, and 7 more variables:
## #   Native_Hawaiian_and_Other_Pacific_Islander_alone <dbl>,
## #   Some_other_race_alone <dbl>, Two_or_more_races <dbl>, White_alone <dbl>,
## #   HouseholdIncome <dbl>, Latitude <dbl>, Longitude <dbl>
```

Decision Trees

- To Classify: Classification Tree
- To Predict: Regression Tree

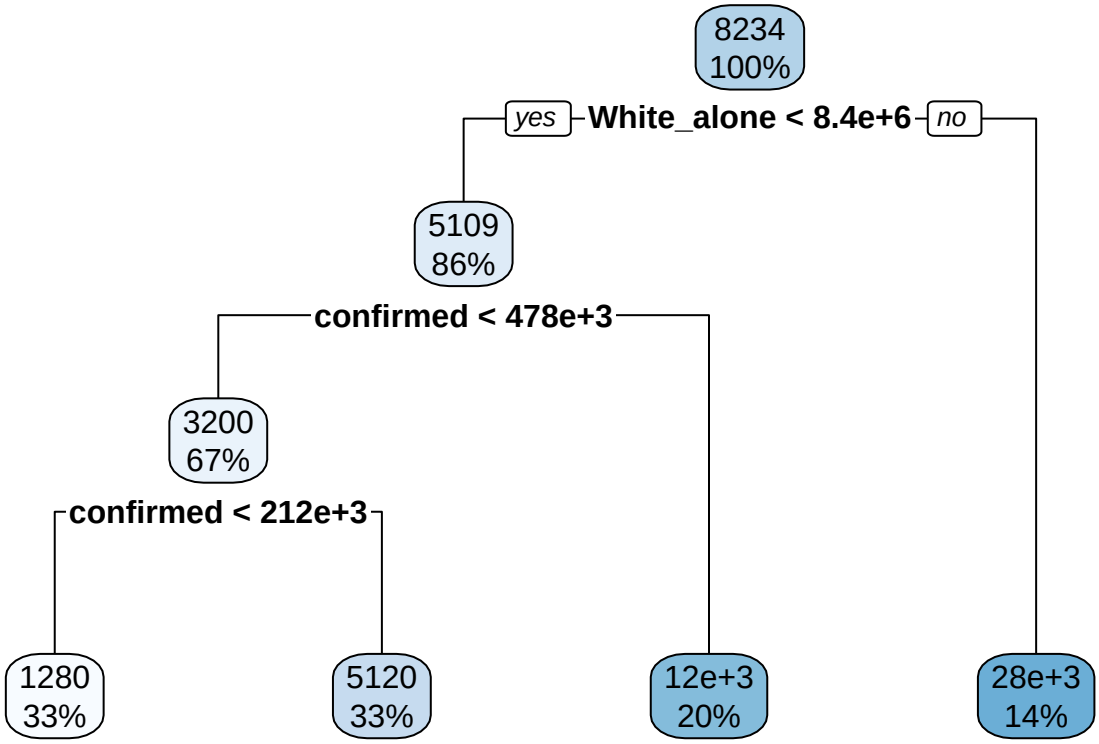
Decision Tree is one of the popular analytic methods. It is a non-parametric regression method. Non-parametric no need to assume the data linearity and normality. Decision Tree can use to build a non-linear model. There are two types of Decision Trees. If the problem we want to determine is having a categorical response variable, and we need to split a response variable into classes, we will use a Classification Tree, which is an algorithm that is able to identify the class of categorical variables. For example, Yes or no, death or alive, male or female. If we want to predict a response variable, which is a continuous response variable, we

will use a Regression Tree, which is an algorithm that is the same as regression analysis giving a continuous result. The predicted result will be continuous. For example, people's' height and weight. Here we want to predict COVID death, which is a continuous response variable. In the following, we will build a Regression tree to Predict COVID death.

Regression Trees | Analysiss

Prediciting COVID death case Using Regression Trees

```
reg.tree <- rpart(deaths ~ ., data = covid_df_train)
rpart.plot(reg.tree)
```

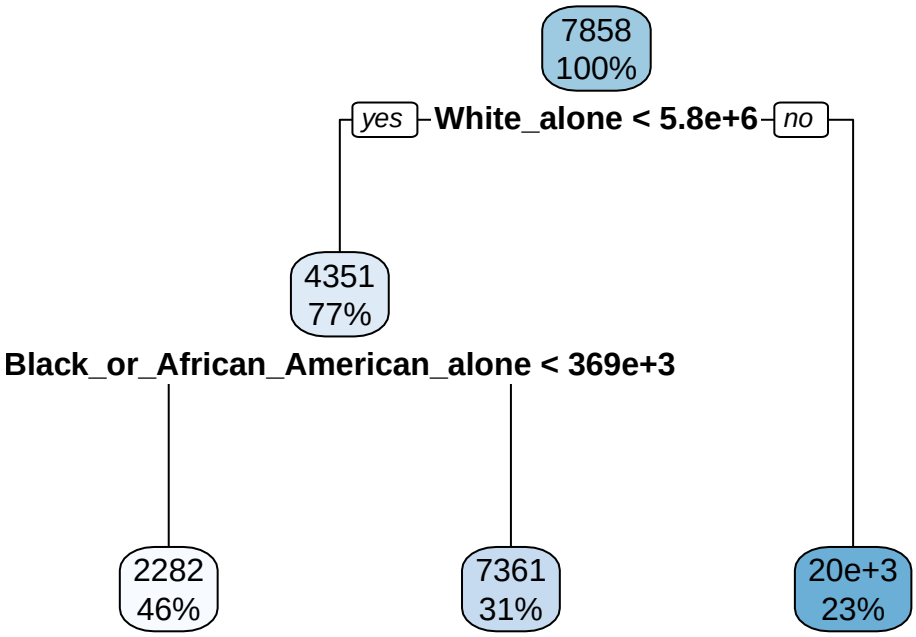


Build a training set

Here we use covid data to build a regression tree to predic death case.

We randomly split data into 70% of the observations for training the models, and leaving 30% for validation.

```
# build a training set
# ensure reproducibility
set.seed(100)
# sample 70% of the row indices for training the models
train <- sample(1:nrow(covid_df_train), nrow(covid_df_train)*0.7)
tree.covid <- rpart(deaths ~ ., subset = train, data = covid_df_train, method = "anova")
rpart.plot(tree.covid)
```




```
summary(tree.covid)
```

```
## Call:
## rpart(formula = deaths ~ ., data = covid_df_train, subset = train,
##     method = "anova")
##   n= 35
##
##           CP nsplit rel error    xerror      xstd
## 1 0.67253561      0 1.0000000 1.0401256 0.3566814
## 2 0.07783447      1 0.3274644 0.5191207 0.1672439
## 3 0.01000000      2 0.2496299 0.4687290 0.1677479
##
## Variable importance
##                     White_alone                       confirmed
##                              23                              20
## Black_or_African_American_alone          Some_other_race_alone
##                              16                              14
##               Two_or_more_races                     Asian_alone
##                              14                              11
##                       Longitude                        Latitude
##                               1                               1
##
## Node number 1: 35 observations,    complexity param=0.6725356
##   mean=7858.029, MSE=6.170986e+07
##   left son=2 (27 obs) right son=3 (8 obs)
##   Primary splits:
##       White_alone                    < 5807693   to the left,  improve=0.6725356, (0 missing)
##       confirmed                      < 586455.5  to the left,  improve=0.6329116, (0 missing)
##       Some_other_race_alone          < 330157    to the left,  improve=0.5113241, (0 missing)
##       Black_or_African_American_alone < 1172574   to the left,  improve=0.4906350, (0 missing)
##       Asian_alone                    < 318567.5  to the left,  improve=0.4739144, (0 missing)
##   Surrogate splits:
##       confirmed                      < 586455.5  to the left,  agree=0.971, adj=0.875, (0 split)
##       Black_or_African_American_alone < 1366993   to the left,  agree=0.914, adj=0.625, (0 split)
```

```
##       Some_other_race_alone           < 313499.5  to the left,  agree=0.914, adj=0.625, (0 split)
##       Two_or_more_races               < 257444    to the left,  agree=0.914, adj=0.625, (0 split)
##       Asian_alone                     < 285532    to the left,  agree=0.886, adj=0.500, (0 split)
##
## Node number 2: 27 observations,    complexity param=0.07783447
##   mean=4351.333, MSE=9934962
##   left son=4 (16 obs) right son=5 (11 obs)
##   Primary splits:
##       Black_or_African_American_alone < 368957    to the left,  improve=0.6267070, (0 missing)
##       confirmed                       < 198306.5  to the left,  improve=0.5886668, (0 missing)
##       White_alone                     < 1697487   to the left,  improve=0.5241572, (0 missing)
##       Some_other_race_alone           < 26820.5   to the left,  improve=0.4601538, (0 missing)
##       Asian_alone                     < 44878     to the left,  improve=0.3479326, (0 missing)
##   Surrogate splits:
##       confirmed             < 373415.5  to the left,  agree=0.815, adj=0.545, (0 split)
##       Longitude             < -92.91345 to the left,  agree=0.815, adj=0.545, (0 split)
##       Some_other_race_alone < 53460     to the left,  agree=0.741, adj=0.364, (0 split)
##       White_alone           < 1697487   to the left,  agree=0.741, adj=0.364, (0 split)
##       Latitude              < 35.03085  to the right, agree=0.741, adj=0.364, (0 split)
##
## Node number 3: 8 observations
##   mean=19693.12, MSE=5.487854e+07
##
## Node number 4: 16 observations
##   mean=2282.375, MSE=2390688
##
## Node number 5: 11 observations
##   mean=7360.727, MSE=5625689
```

Values on the node represent:

- death cases
- Percentage of observations account for each node

Print the rules of regression tree

```
rpart.rules(x = tree.covid, cover = TRUE)
```

```
##  deaths                                                                                cover
##    2282 when White_alone <  5807693 & Black_or_African_American_alone <  368957     46%
##    7361 when White_alone <  5807693 & Black_or_African_American_alone >= 368957     31%
##   19693 when White_alone >= 5807693                                                  23%
```

`White_alone`is the first layer variable, which is the most important variable. If the state includes the white pepole is greater than `5,800,000`, the average death case is 20,000 people, accounting for 23% of total death cases. In contrast, if the white people is less than `5,800,000`, the average number of deaths is 4351 people, accounting for 77% of total death cases. The second important independent variable is `Black_or_African_American_alone`. According to the tree plot, if the state includes less than `369,000` people of the black or African American race, the average number of deaths is 2282, accounting for 46 percent of total death cases. Conversely, if the state includes greater than `369,000` people of them, the average number of deaths is 7361, accounting for 31 percent of total death cases.

Check importance

```
tree.covid$variable.importance
```

```
##                      White_alone                        confirmed
##                        1513703720                       1362697662
## Black_or_African_American_alone           Some_other_race_alone
##                        1075968307                        968988969
##                Two_or_more_races                      Asian_alone
##                         907857919                        726286335
##                         Longitude                         Latitude
##                          91696575                         61131050
```

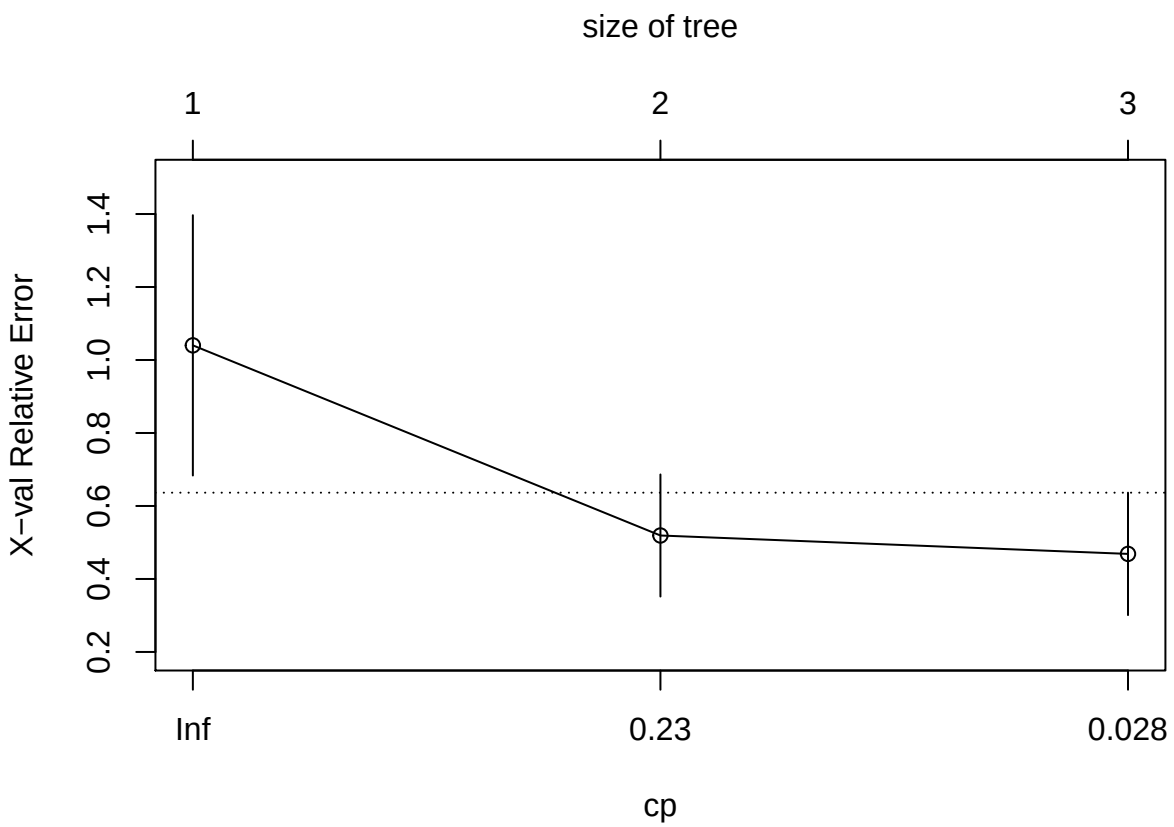
Variable.importance command can examine variable importance for each predictor variables. Here we can see that the most important variable is White_alone, and the last one is Latitude.

cross-validation

```
printcp(tree.covid)
```

```
##
## Regression tree:
## rpart(formula = deaths ~ ., data = covid_df_train, subset = train,
##     method = "anova")
##
## Variables actually used in tree construction:
## [1] Black_or_African_American_alone White_alone
##
## Root node error: 2159844989/35 = 61709857
##
## n= 35
##
##         CP nsplit rel error  xerror    xstd
## 1 0.672536      0   1.00000 1.04013 0.35668
## 2 0.077834      1   0.32746 0.51912 0.16724
## 3 0.010000      2   0.24963 0.46873 0.16775
```

```
plotcp(x = tree.covid)
```



Rpart function, by default, will cross-validate the results of the tree and trim the tree.

The complexity parameter (cp) is used to control the size of the tree and to select the optimal tree size.

Y-axis illustrates the relative cross validation error for various cp values. Smaller cp values lead to larger trees (we can see the upper x-axis for tree size)

If the cost of adding another variable to the tree from the current node is above the value of cp, then tree building does not continue.

Then, we can use this predict function to make prediction.

Prediction function

```
# predict testing set
pred <- predict(tree.covid, covid_df_train[-train,])
pred
```

```
##         1         2         3         4         5         6         7         8
##  2282.375 19693.125  2282.375  7360.727  2282.375  7360.727  2282.375  2282.375
##         9        10        11        12        13        14        15        16
##  2282.375 19693.125 19693.125  2282.375  2282.375  2282.375  7360.727  2282.375
```