R Project: COVID19 Cases

4375 Machine Learning with Dr. Mazidi

David Allen

March 24, 2022

This is a regression dataset for predicting the death rate of the coronavirus based on location and date in 2020.

- Number of Rows: 50k

Source: https://www.kaggle.com/datasets/imdevskp/corona-virus-report

The three algorithms used are Linear Regression, kNN, and Decision Trees.

Load the Data

```
df <- read.csv("data/covid_19_complete.csv")
str(df)
```

```
## 'data.frame':    49068 obs. of  10 variables:
##  $ Province.State: chr  "" "" "" "" ...
##  $ Country.Region: chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
##  $ Lat           : num  33.9 41.2 28 42.5 -11.2 ...
##  $ Long          : num  67.71 20.17 1.66 1.52 17.87 ...
##  $ Date          : chr  "2020-01-22" "2020-01-22" "2020-01-22" "2020-01-22" ...
##  $ Confirmed     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Deaths        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Recovered     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Active        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WHO.Region    : chr  "Eastern Mediterranean" "Europe" "Africa" "Europe" ...
```

Data Exploration and Cleaning

```
summary(df$Confirmed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       4     168   16885    1518 4290259
```

```
summary(df$Deaths)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      0.0      0.0      2.0    884.2     30.0 148011.0
```

```
mean(df$Confirmed == 0)
```

```
## [1] 0.2050012
```

There is massive variance in the data. Confirmed cases range from 0 cases to 4 million! This will have costly effects on the algorithms.

It also appears about 20% of the data is simply 0's from the beginning of the pandemic.

```
sum(df$Province.State == "")
```

```
## [1] 34404
```

Province.State is a feature that is mostly blank for this data set. But ultimately, this doesn't matter for our problem. Latitude, longitude, and WHO region (as a factor) will account for the location data.

Cleaning Steps

- Convert Date to numerical data
- Convert region to a factor

We will need to adjust the date values from categorical to numerical to use in regression:

```
parseDate <- function(date) {
  numeric_date <- as.numeric(as.Date(date, "%Y-%m-%d"))
  return (numeric_date)
}
```

```
df$Date.Numeric <- sapply(df$Date, parseDate)
summary(df$Date.Numeric)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18283   18330   18377   18377   18423   18470
```

Now let's scale the data:

```
df$Date.Numeric <- df$Date.Numeric - mean(df$Date.Numeric)
summary(df$Date.Numeric)
```

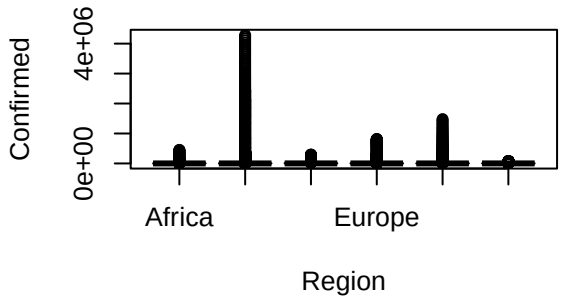
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -93.50  -46.75    0.00    0.00   46.75   93.50
```

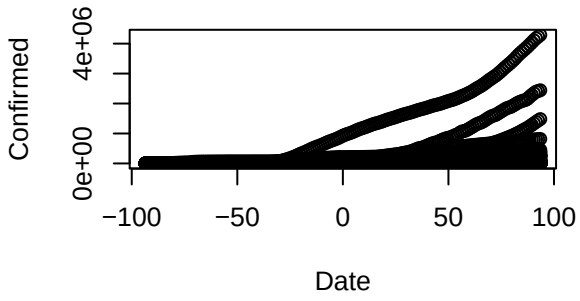
```
df$WHO.Region <- as.factor(df$WHO.Region)
str(df)
```

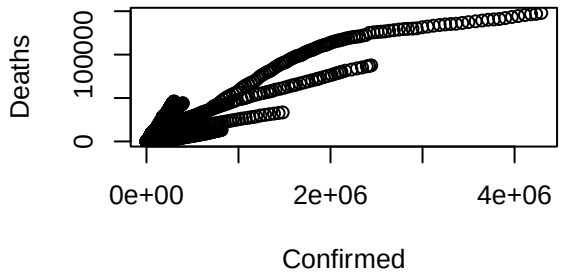
```
## 'data.frame':    49068 obs. of  11 variables:
##  $ Province.State: chr  "" "" "" "" ...
##  $ Country.Region: chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
##  $ Lat           : num  33.9 41.2 28 42.5 -11.2 ...
##  $ Long          : num  67.71 20.17 1.66 1.52 17.87 ...
##  $ Date          : chr  "2020-01-22" "2020-01-22" "2020-01-22" "2020-01-22" ...
##  $ Confirmed     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Deaths        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Recovered     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Active        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WHO.Region    : Factor w/ 6 levels "Africa","Americas",..: 3 4 1 4 1 2 2 4 6 6 ...
##  $ Date.Numeric  : num  -93.5 -93.5 -93.5 -93.5 -93.5 -93.5 -93.5 -93.5 -93.5 -93.5 ...
```

Now we have 4 reliable features to use as predictors for regression: Lat, Long, Date.Numeric, and WHO.Region. There are 4 categories to predict, which could potentially work as features for each other: Confirmed, Deaths, Recovered, and Active. Let's predict Confirmed cases from our 4 predictors.

```
par(mfrow=c(2, 2))
plot(df$Confirmed~df$WHO.Region, ylab="Confirmed", xlab="Region")
plot(df$Confirmed~df$Date.Numeric, ylab="Confirmed", xlab="Date")
plot(df$Deaths~df$Confirmed, ylab="Deaths", xlab="Confirmed")
```







Based on the first graph, the number of cases skyrocketed the most in Asia (as expected), and least in South

America. One can see multiple lines in the following two graphs, most likely due to the different experiences of each region: for Confirmed cases vs. Date, it is clear that as time goes on, the number of cases increases; for Deaths vs. Confirmed, the number of deaths obviously positively correlates with the number of cases.

Testing the Model

```
set.seed(777)
i <- sample(1:nrow(df), nrow(df)*0.8, replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Linear Regression

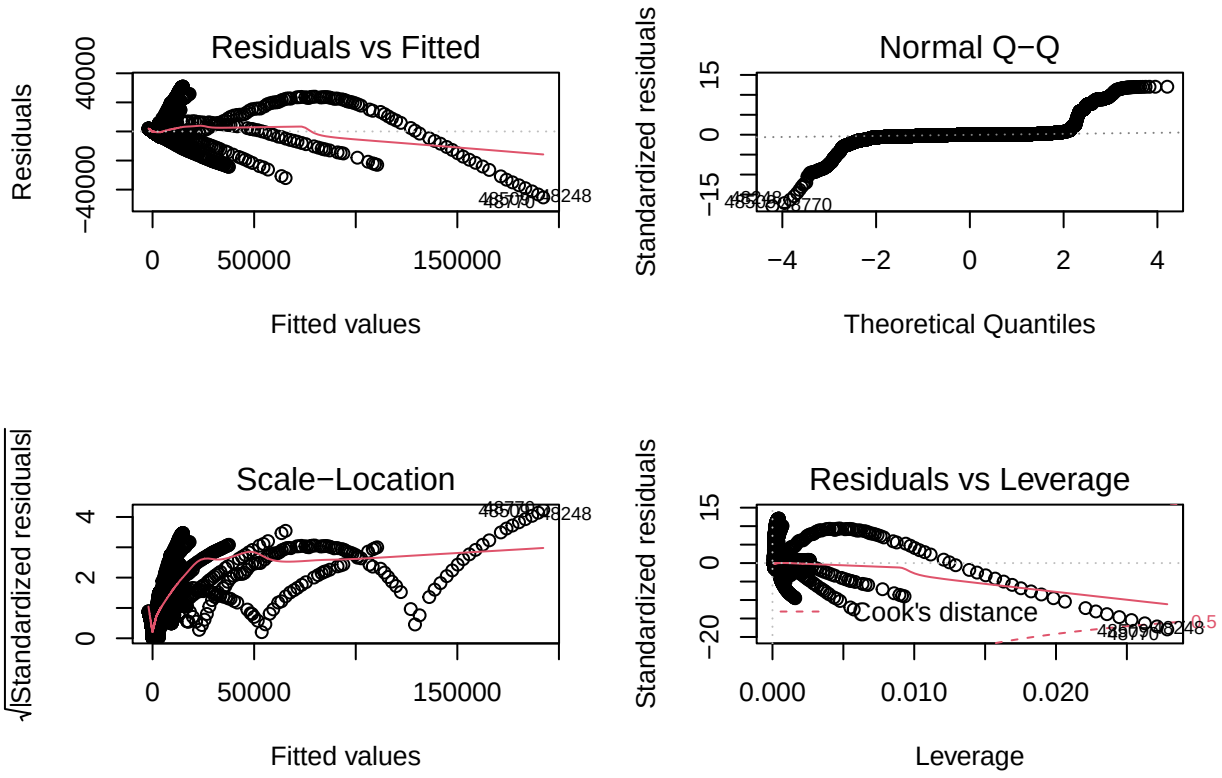
We start with the simplest algorithm. We'll also track the interaction effects between date and region.

```
lm1 <- lm(Deaths~Date.Numeric*WHO.Region+Lat+Long+Date.Numeric+Confirmed, data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = Deaths ~ Date.Numeric * WHO.Region + Lat + Long +
##     Date.Numeric + Confirmed, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45360   -343    -96    109  30826
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 2.600e+01  3.100e+01   0.839
## Date.Numeric                               -1.433e+00  5.571e-01  -2.572
## WHO.RegionAmericas                         -4.064e+02  5.898e+01  -6.891
## WHO.RegionEastern Mediterranean            -2.538e+02  5.796e+01  -4.379
## WHO.RegionEurope                            3.789e+02  4.440e+01   8.534
## WHO.RegionSouth-East Asia                  -1.067e+02  8.138e+01  -1.311
## WHO.RegionWestern Pacific                   5.203e+02  6.374e+01   8.162
## Lat                                         4.686e+00  6.211e-01   7.545
## Long                                       -5.490e+00  4.362e-01 -12.587
## Confirmed                                   4.533e-02  1.032e-04 439.155
## Date.Numeric:WHO.RegionAmericas             1.495e+00  8.009e-01   1.866
## Date.Numeric:WHO.RegionEastern Mediterranean -5.840e+00  9.952e-01  -5.868
## Date.Numeric:WHO.RegionEurope               8.959e+00  7.042e-01  12.721
## Date.Numeric:WHO.RegionSouth-East Asia     -1.147e+01  1.333e+00  -8.601
## Date.Numeric:WHO.RegionWestern Pacific      1.137e+00  7.612e-01   1.494
##                                            Pr(>|t|)
## (Intercept)                                  0.4016
## Date.Numeric                                 0.0101 *
## WHO.RegionAmericas                         5.63e-12 ***
## WHO.RegionEastern Mediterranean            1.20e-05 ***
## WHO.RegionEurope                            < 2e-16 ***
## WHO.RegionSouth-East Asia                    0.1897
```

```
## WHO.RegionWestern Pacific                      3.38e-16 ***
## Lat                                            4.62e-14 ***
## Long                                            < 2e-16 ***
## Confirmed                                       < 2e-16 ***
## Date.Numeric:WHO.RegionAmericas                  0.0620 .
## Date.Numeric:WHO.RegionEastern Mediterranean 4.43e-09 ***
## Date.Numeric:WHO.RegionEurope                   < 2e-16 ***
## Date.Numeric:WHO.RegionSouth-East Asia          < 2e-16 ***
## Date.Numeric:WHO.RegionWestern Pacific           0.1353
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2562 on 39239 degrees of freedom
## Multiple R-squared:  0.8396, Adjusted R-squared:  0.8395
## F-statistic: 1.467e+04 on 14 and 39239 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm1)
```



We get some kooky-looking graphs that clearly illustrate how much the categories the data affect its overall shape.

```
pred1 <- predict(lm1, newdata=test)
cor1 <- cor(pred1, test$Deaths)
cor1
```

```
## [1] 0.9115718
```

```
rmse1 <- sqrt(mean((pred1 - test$Deaths)^2))
rmse1
```

```
## [1] 2456.699
```

kNN Regression

Perhaps the kNN algorithm will have a better understanding of how the data clusters based on predictors.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
predictors <- c(3, 4, 6, 11)
fit <- knnreg(train[predictors], train$Deaths, k=12)
pred2 <- predict(fit, test[predictors])
cor2 <- cor(pred2, test$Deaths)
cor2
```

```
## [1] 0.9265572
```

```
rmse2 <- sqrt(mean((pred2 - test$Deaths)^2))
rmse2
```

```
## [1] 2271.43
```

It appears so – the correlation is slightly increased, and error slightly decreased.

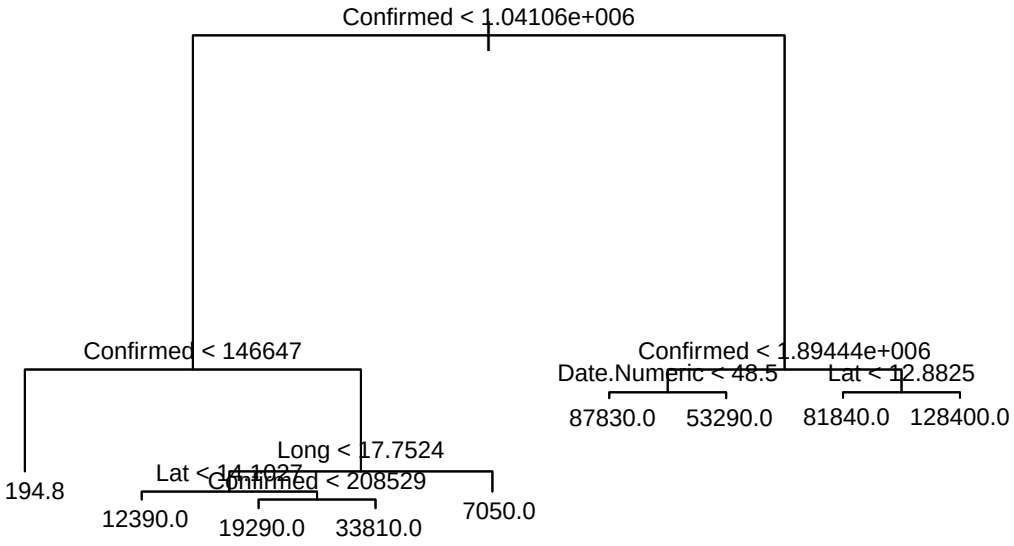
Decision Trees

Decision trees seem like the perfect fit for this problem, as they use predictors as conditions to narrow down the data.

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.1.3
```

```
tree1 <- tree(Deaths~Date.Numeric+Confirmed+WHO.Region+Lat+Long, data=train)
plot(tree1)
text(tree1, cex=0.75, pretty=0)
```

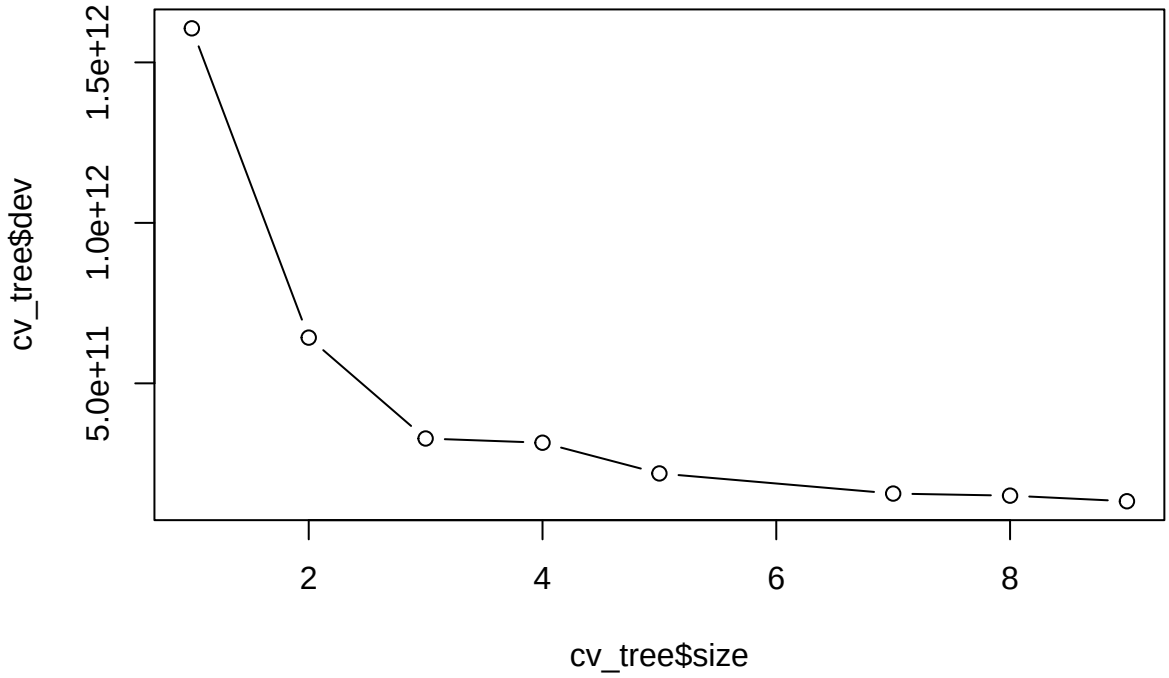



```
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = Deaths ~ Date.Numeric + Confirmed + WHO.Region +
##     Lat + Long, data = train)
## Variables actually used in tree construction:
## [1] "Confirmed"    "Long"         "Lat"          "Date.Numeric"
## Number of terminal nodes:  9
## Residual mean deviance:  2738000 = 1.074e+11 / 39240
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -26470.0   -194.8   -192.8      0.0   -169.8  36560.0
```

Surprisingly, it seems the WHO Region is not helpful to the tree.

```
cv_tree <- cv.tree(tree1)
plot(cv_tree$size, cv_tree$dev, type='b')
```



We opt for no pruning because in a case like this, the tree needs more branches rather than fewer, to utilize all the data available.

The following produces an identical tree:

```
tree_pruned <- prune.tree(tree1, best="9")
```

```
pred3 <- predict(tree_pruned, newdata=test)
cor3 <- cor(pred3, test$Deaths)
cor3
```

```
## [1] 0.9590506
```

```
rmse3 <- sqrt(mean((pred3 - test$Deaths)^2))
rmse3
```

```
## [1] 1691.974
```

It seems a Decision tree well outperforms the previous two models!

Analysis

The error for all three algorithms was abysmal, although the correlations were high. It appears the script was able to understand a general relationship between the date, the location, and the number of cases, but

severely underestimated how dramatically the virus spread, or how there are simply 0 cases before a location has an outbreak.

Here is the ranking of the algorithms (based on correlation and error):

- Third Place: Linear Regression
- Second Place: kNN Regression
- First Place: Decision Trees

It is a pleasant surprise that a Decision Tree made for the best algorithm to make predictions on the data. This regression problem is unique in that certain conditions (the where and when) impact the curve of the number of cases immensely. It is apt that a tree which breaks the data down using these conditions would have the clearest picture of what is going on.

That said, given the error is extremely high for each algorithm, this script will probably not be useful in any application outside of this class. Although another thing to consider is that when dealing with numbers in the hundreds of thousands, being off by one thousand is not too bad. With some further tweaking, perhaps a better algorithm could be devised to handle the variance better and make better predictions. Overall though, I am satisfied with the results of this project.