Building a Model to Predict Outcome of Covid Cases

Sean Lau Kuang Qi

4th October 2024

Introduction

The purpose of this report is to present the findings of a study focused on predicting the outcome of COVID-19 victims based on several key predictors, including their age group, if they were admitted to an ICU, source of infection, gender, if they were hospitalized, and outbreak association. The primary objective of this study is to identify patterns and relationships among these variables that can be used to accurately predict whether or not an individual is likely to succumb to COVID-19. Through a detailed analysis of the data, this report provides insights that can be used to improve our understanding of the factors that contribute to COVID-19 outcomes.

Overall Question: What are the patterns and relationships among key predictors such as age group, admission to ICU, source of infection, gender, hospitalization history, and outbreak association that can be used to accurately predict whether or not an individual is likely to succumb to COVID-19?

Background About Dataset

The city of Toronto has been experiencing an ongoing COVID-19 outbreak amidst a global pandemic, and Toronto Public Health is actively responding to this situation. All confirmed and probable cases reported to and managed by Toronto Public Health since January 2020 have been included in this dataset, which provides demographic, geographic, and severity information. The dataset encompasses cases that are sporadic, as well as those that are related to outbreaks. The data has been extracted from the provincial Case & Contact Management System (CCM) and published by Toronto Public Health.

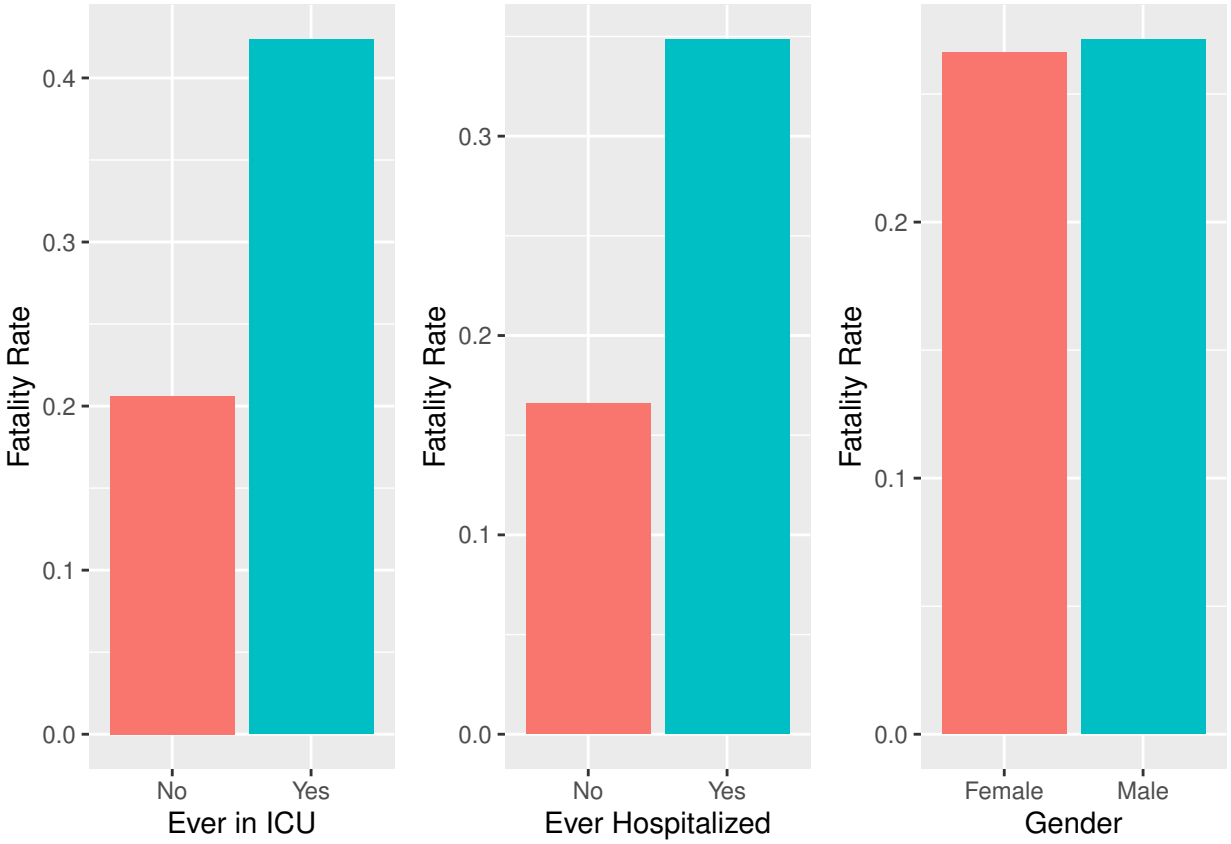
Description of Variables and the Data

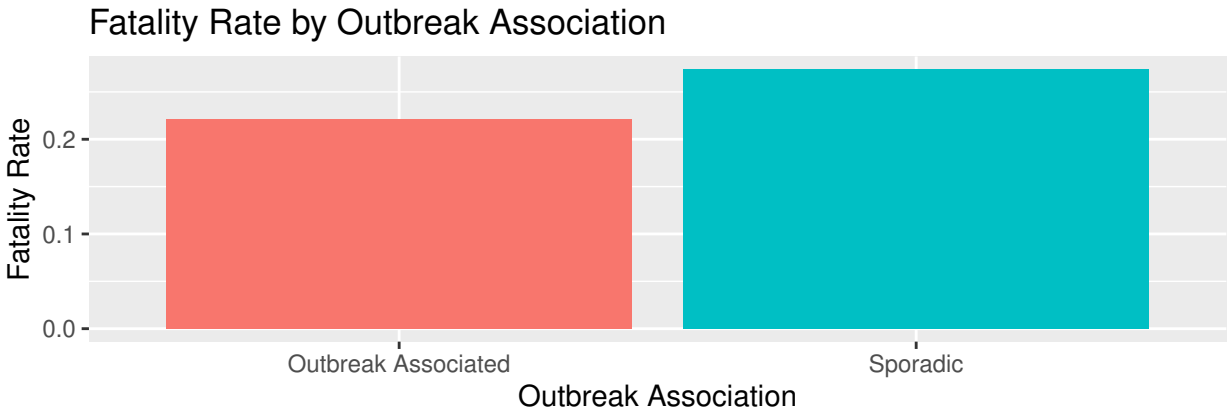
Variable	Description
Age Group	Age at time of illness. Age groups (in years): < or equal to 19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90+
Ever in ICU	Cases that were admitted to the intensive care unit (ICU) related to their COVID-19 infection (includes cases that are currently in ICU and those that have been discharged or are deceased).

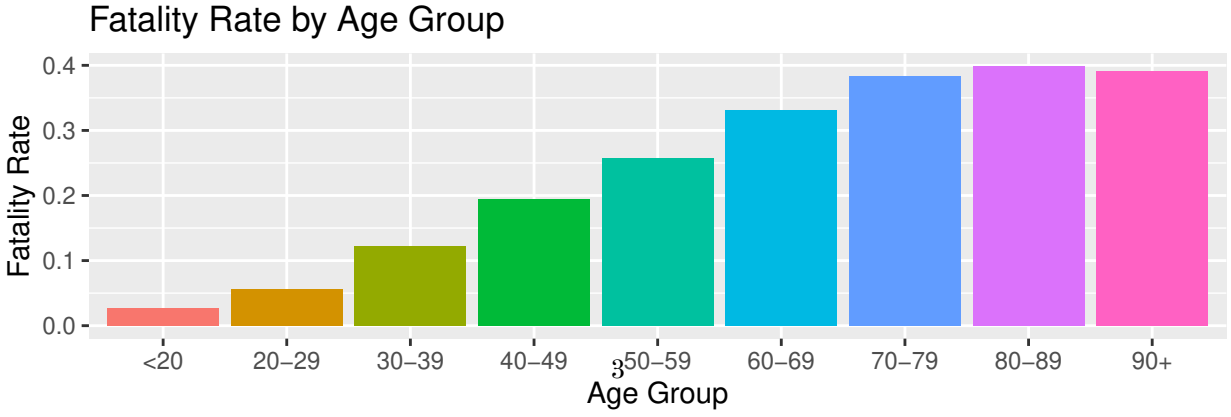
Variable	Description
Source of Infection	The most likely way that cases acquired their COVID-19 infection is determined by examining several data fields including: A public health investigator's assessment of the most likely source of infection, being associated with a confirmed COVID-19 outbreak, and reported risk factors such as contact with a known case or travel. Descriptions: - Household contact: Case who acquired infection from a household contact with a confirmed or probable COVID-19 case (e.g. family member, roommate). - Close contact with a case: Case who acquired infection from a close contact with a confirmed or probable COVID-19 case (e.g. co-worker). - Outbreaks, Congregate Settings: confirmed outbreaks in Toronto in shelters, correctional facilities, group homes, or other congregate settings such as hostels or rooming houses. - Outbreaks, Healthcare Institutions: confirmed outbreaks in Toronto in long-term care homes, retirement homes, hospitals, chronic care hospitals, or other institutional settings. - Outbreaks, Other Settings: confirmed outbreaks in Toronto in workplaces, schools, day cares, or outbreaks outside of Toronto. We do not validate outbreaks that occur in other health units, as such these cases may not be linked to confirmed outbreaks. - Travel: Case that travelled outside of Ontario in the 14 days prior to their symptom onset or test date, whichever is the earliest. - Community: Cases who did not travel outside of Ontario, did not identify being a close contact with a COVID-19 case, and were not part of a known confirmed COVID-19 outbreak.
Client Gender	Self-reported gender. Gender is a system that operates in a social context and generally classifies people based on their assigned biological sex.
Ever Hospitalized	Cases that were hospitalized related to their COVID-19 infection (includes cases that are currently hospitalized and those that have been discharged or are deceased).
Outcome	- Fatal: Any case that has died and has been marked as Outcome equals 'Fatal' and Type of Death does not equal 'Disease of Public Health Significance was unrelated to cause of death' in the provincial reporting system (CCM). - Resolved: Cases who have: A case outcome description in CCM of 'Recovered' OR Case outcome description is equal to 'Fatal' AND Type of Death is equal to 'Disease of Public Health Significance was unrelated to cause of death' OR Today's date is more than 14 days from episode date AND the case is not currently hospitalized/intubated/in ICU AND Case outcome description is not equal to 'Fatal' where Type of Death is not equal to 'Disease of Public Health Significance was unrelated to cause of death'. - Active: All other cases.
Outbreak Associated	Outbreak associated cases are associated with outbreaks of COVID-19 in Toronto healthcare institutions and healthcare settings (e.g. long-term care homes, retirement homes, hospitals, etc.) and other Toronto congregate settings (such as homeless shelters).
Classification	The application of the provincial case definition to categorize the cases as confirmed or probable, according to standard criteria.

Exploratory Data Analysis

Showing Fatality Rates vs Levels of Predictor Varibales







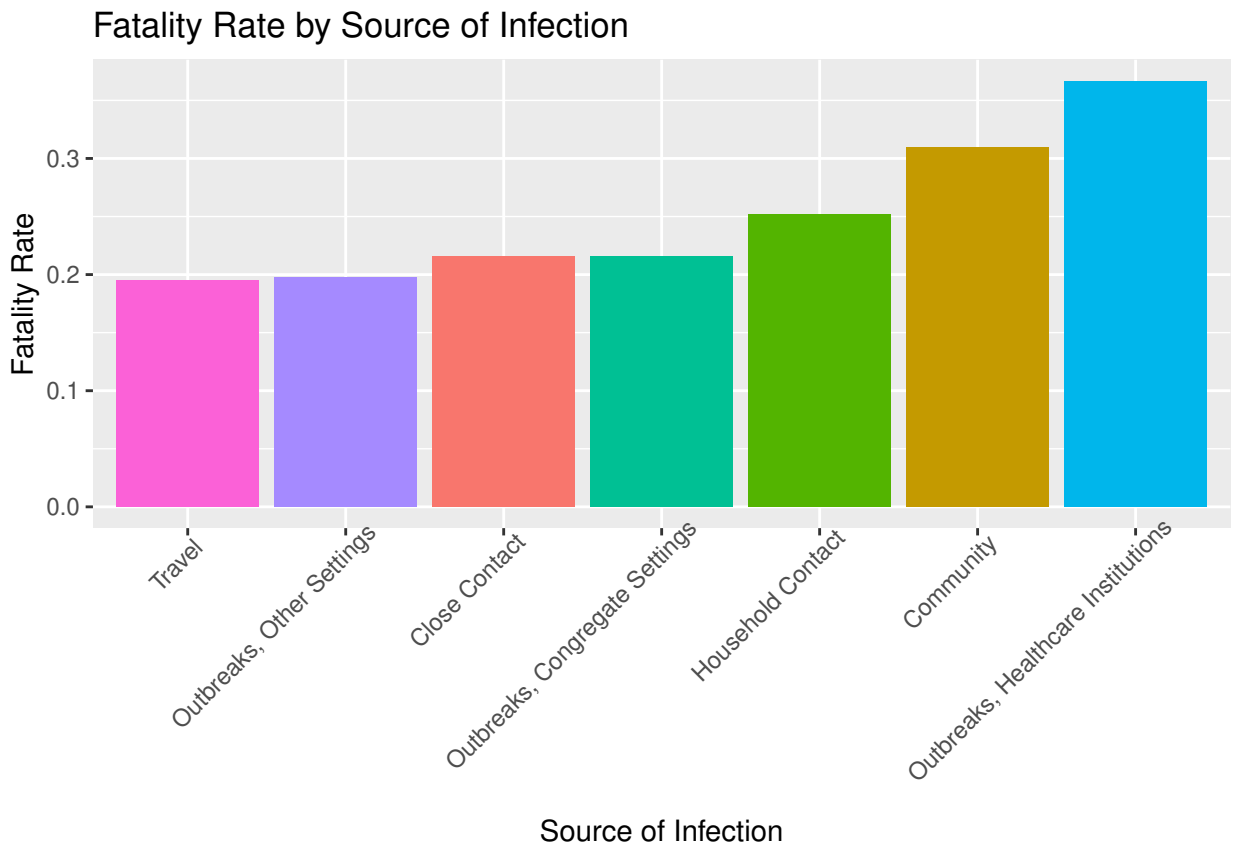


Table 2: Contigency Table of Fatality " vs Age Group

	Fatal(0) / Resolved(1)		
Age Group	FATAL	RESOLVED	Total
19 and younger	2 (2.7%)	72 (97.3%)	74 (100.0%)
20 to 29 Years	5 (5.6%)	85 (94.4%)	90 (100.0%)
30 to 39 Years	13 (12.1%)	94 (87.9%)	107 (100.0%)
40 to 49 Years	21 (19.4%)	87 (80.6%)	108 (100.0%)
50 to 59 Years	29 (25.7%)	84 (74.3%)	113 (100.0%)
60 to 69 Years	40 (33.1%)	81 (66.9%)	121 (100.0%)
70 to 79 Years	44 (38.3%)	71 (61.7%)	115 (100.0%)
80 to 89 Years	47 (39.8%)	71 (60.2%)	118 (100.0%)
90 and older	34 (39.1%)	53 (60.9%)	87 (100.0%)
Total	235 (25.2%)	698 (74.8%)	933 (100.0%)

Hypothesis Testing

```r
# Create contingency table for Age.Group and Client.Gender
age_outcome_table <- table(clean_data$Age.Group, clean_data$Outcome)

# Perform Chi-squared test of independence
age_outcome_chi_square <- chisq.test(age_outcome_table)

# Print the result
```

```r
print(age_outcome_chi_square) ## Age and Outcome are not independent since
```

```
##
##  Pearson's Chi-squared test
##
## data:  age_outcome_table
## X-squared = 86.566, df = 8, p-value = 2.311e-15
```

We set alpha to 0.05 and Ho: there is not a strong relationship between the two variables being considered (they are independent), and Ha: there is a significant relationship between the two variables (dependent). The p-value is approximately 0, indicating that we reject Ho, and conclude that an individuals age group and covid recovery outcome are not independent.

Model Selection

```r
set.seed(123)
## Split data into training and testing set (Cross-Validation)
train_index <- createDataPartition(clean_data$Outcome, p = 0.8, list = FALSE)
train <- clean_data[train_index, ]
test <- clean_data[-train_index, ]

# Factor "Outcome" variable as 1 (Resolved) or 0 (Fatal)
clean_data$Outcome<- ifelse(clean_data$Outcome == "RESOLVED", 1, 0)
train$Outcome <- ifelse(train$Outcome == "RESOLVED", 1, 0)
test$Outcome <- ifelse(test$Outcome == "RESOLVED", 1, 0)

# Fit the logistic regression model
model <- glm(Outcome ~ ., data = train, family = "binomial")
# summary(model)
model$aic
```

```
## [1] 718.4048
```

```r
# Step-wise fit
initial_model <- glm(Outcome ~ 1, data =train, family = "binomial")

step_model_both <- step(model, direction = "both", trace = 0)
step_model_backward <- step(model, direction = "backward", trace = 0)
step_model_forward <- step(model, direction = "forward",
                           scope = list(lower = initial_model, upper = model),
                           trace = 0)

step_model_both$aic
```

```
## [1] 716.8074
```

```r
step_model_backward$aic
```

```
## [1] 716.8074
```

```r
step_model_forward$aic
```

```
## [1] 718.4048
```

```r
## Backward and Bi-directional have the lowest AIC's and offer the same final
## model which excludes the variables, "Cases", "Ever.Hospitalized".
```

Manual Selection

```
# Assess the effect of dropping each predictor variable
dropped_variable_effects <- drop1(model, test = "Chi")

## Setting alpha at 0.01, "Client.Gender" appears to not be a significant
## predictor variable so we drop it to see how the model might improve.

# Manually Selected Model
model2 <- glm(Outcome ~ Age.Group + Ever.in.ICU + Source.of.Infection
              + Outbreak.Associated + Classification
    , data = train, family = "binomial")
model2$aic
```

```
## [1] 724.4575
```

```
## AIC actually increases when we exclude the gender variable
```

Here we used the drop1() function to try and further simplify the automatically-selected model by dropping the least significant variables using a significance level of 0.01. The variable that we dropped using this method was Client.Gender.

Model Diagnostics

```
test$Outcome <- as.factor(test$Outcome)
train$Outcome <- as.factor(train$Outcome)

# Evaluate the model's performance on test data (Step-wise Model)
test$predicted_prob <- predict(step_model_both, test, type = "response")
test$predicted_outcome <- as.factor(ifelse(test$predicted_prob > 0.5, 1, 0))

# Check Predictions for Full Model with all variables
test$predicted_prob2 <- predict(model2, newdata = test, type = "response")
test$predicted_outcome2 <- as.factor(ifelse(test$predicted_prob2 > 0.5, 1, 0))

# Performance metrics (Step Wise)
confusion_matrix <- confusionMatrix(test$Outcome, test$predicted_outcome)
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  20  27
##          1  17 122
##
##                Accuracy : 0.7634
##                  95% CI : (0.6957, 0.8225)
##     No Information Rate : 0.8011
##     P-Value [Acc > NIR] : 0.9136
##
##                   Kappa : 0.3262
##
##  Mcnemar's Test P-Value : 0.1748
```

```
##
##             Sensitivity : 0.5405
##             Specificity : 0.8188
##          Pos Pred Value : 0.4255
##          Neg Pred Value : 0.8777
##              Prevalence : 0.1989
##          Detection Rate : 0.1075
##    Detection Prevalence : 0.2527
##       Balanced Accuracy : 0.6797
##
##        'Positive' Class : 0
##
```

```
accuracy <- confusion_matrix$overall["Accuracy"]
precision <- confusion_matrix$byClass["Pos Pred Value"] ## Accurate Fatal Pred
recall <- confusion_matrix$byClass["Sensitivity"]
f1_score <- confusion_matrix$byClass["F1"] ## Measures Precision and Accuracy
print(paste("F1-score:", f1_score))
```

```
## [1] "F1-score: 0.476190476190476"
```

```
# Performance metrics (Full Model)
confusion_matrix2 <- confusionMatrix(test$Outcome, test$predicted_outcome2)
accuracy2 <- confusion_matrix2$overall["Accuracy"]
precision2 <- confusion_matrix2$byClass["Pos Pred Value"]##Accurately detect bad
recall2 <- confusion_matrix2$byClass["Sensitivity"]
f1_score2 <- confusion_matrix2$byClass["F1"]
print(paste("F1-score:", f1_score2))
```

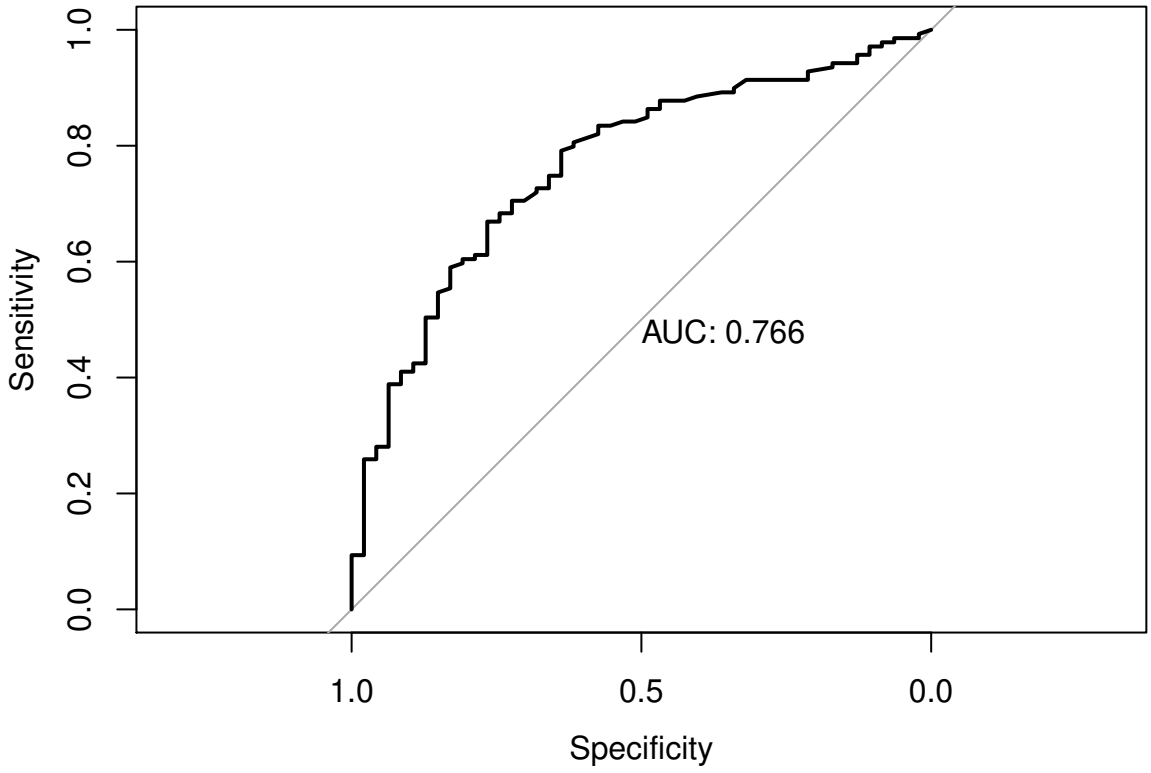
```
## [1] "F1-score: 0.338028169014085"
```

```
# ROC curve (Step wise fit)
roc_obj <- roc(test$Outcome, test$predicted_prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc <- pROC::auc(roc_obj)
plot(roc_obj, print.auc = TRUE, print.auc.corners = TRUE)
```



F1 score which accounts for accuracy and precision reduces in our manually selected model, in addition to this model also having a higher AIC. Therefore we opted to go with the step wise model where the area under the ROC curve is 0.766 which suggests that this model is an acceptable fit with good performace.

Coefficient Intepretation

From our model summary we can interpret our regression coefficients as such:

Age.Group40 to 49 Years: The estimate for this age group is -2.74259. This means that the log-odds of a positive outcome for someone in the 40 to 49 age group is 2.74259 units lower compared to the reference age group, holding all other predictors constant. The coefficient is significant at the 1% level (p-value = 0.009651).

Ever.in.ICUYes: The estimate for this variable is -0.72290. This means that the log-odds of a positive outcome for someone who has ever been in ICU is 0.72290 units lower compared to someone who has never been in ICU, holding all other predictors constant. The coefficient is significant at the 1% level (p-value = 0.000862).

Source.of.InfectionOutbreaks, Healthcare Institutions: The estimate for this variable is -1.10173. This means that the log-odds of a positive outcome for someone who contracted COVID-19 from an outbreak in a healthcare institution is 1.10173 units lower compared to the reference source of infection category, holding all other predictors constant. The coefficient is significant at the 1% level (p-value = 0.007272).

Outbreak.AssociatedSporadic: The estimate for this variable is -0.85741. This means that the log-odds of a positive outcome for someone who is associated with a sporadic outbreak is 0.85741 units lower compared to someone who is not, holding all other predictors constant. The coefficient is significant at the 1% level (p-value = 0.005642).

ClassificationPROBABLE: The estimate for this variable is 1.41836. This means that the log-odds of a positive outcome for someone with a probable classification is 1.41836 units higher compared to someone

with a different classification, holding all other predictors constant. The coefficient is significant at the 1% level (p-value = 1.97e-06).

Bootstrapping

```
##                                                    Within_Bootstrap_Interval
## (Intercept)                                                             TRUE
## Age.Group20 to 29 Years                                                 TRUE
## Age.Group30 to 39 Years                                                 TRUE
## Age.Group40 to 49 Years                                                 TRUE
## Age.Group50 to 59 Years                                                 TRUE
## Age.Group60 to 69 Years                                                 TRUE
## Age.Group70 to 79 Years                                                 TRUE
## Age.Group80 to 89 Years                                                 TRUE
## Age.Group90 and older                                                   TRUE
## Ever.in.ICUYes                                                          TRUE
## Source.of.InfectionCommunity                                            TRUE
## Source.of.InfectionHousehold Contact                                    TRUE
## Source.of.InfectionOutbreaks, Congregate Settings                       TRUE
## Source.of.InfectionOutbreaks, Healthcare Institutions                   TRUE
## Source.of.InfectionOutbreaks, Other Settings                            TRUE
## Source.of.InfectionTravel                                               TRUE
## Client.GenderMALE                                                       TRUE
## Client.GenderNON-BINARY                                                 TRUE
## Client.GenderOTHER                                                      TRUE
## Client.GenderTRANS MAN                                                  TRUE
## Client.GenderTRANS WOMAN                                                TRUE
## Client.GenderTRANSGENDER                                                TRUE
## Outbreak.AssociatedSporadic                                             TRUE
## ClassificationPROBABLE                                                  TRUE
```

Discussion/Conclusion

The goal of this report was to identify the factors that are likely to influence the outcome of Covid Cases and build a model to predict whether the outcome of cases will be Fatal or Resolved. From our analysis, we identified that the variables Age, ICU visit, Infection Source, Gender and whether or not cases stemmed from outbreaks are the best variables to have in our model to predict what the outcome of an individual will be if infected with covid.

By analyzing the factors that affect recovery outcome of individuals, this can help in various ways, such as defining covid guidelines for different demographics in the population and helping mitigate casualties in higher risk groups. Some of the limitations of our analysis would be the dataset being relatively small for such a task since it only contains 1218 observations and there were a lot of categorical variables which are more difficult to work with and interpret, this was in addition to many ordinal variables with more than two levels making it even more challenging to apply regression techniques effectively. Additionally, the dataset may contain bias or inaccurate representation which we tried to filter out in our Data Massaging. Despite all of these limitations, we believe that our model has achieved the goal of identifying factors to be considered when predicting most probable outcome of Covid Cases.

For future research, it would be interesting to know by how much the factors we identified can increase or decrease odds of fatality in covid cases. Although we were able to identify factors that impact Outcome, there is still the question of which factors have the most significant influence on increasing or decreasing the fatlaity rate of individuals.

Appendix: All code for this report

```
# ```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(caret)
library(corrplot)
library(knitr)
library(tidyverse)
library(janitor)
library(mosaic)
library(dplyr)
library(stats)
library(ggplot2)
library(boot)
library(ResourceSelection)
library(pROC)
library(gridExtra)
library(kableExtra)
library(MASS)
library(patchwork)
# Group data to get count
aggregated_data <- read.csv("Clean_data.csv") %>%
  group_by(Age.Group, Ever.in.ICU, Source.of.Infection, Client.Gender, Ever.Hospitalized, Outcome,Outbr
  summarise(Cases = n()) %>%
  ungroup()

# Filter out bad entries
clean_data <- aggregated_data %>%
  filter(!(Age.Group == "" | is.na(Age.Group)),
         !( Ever.in.ICU == "No information" | ## Drop entries w/o info
             Ever.in.ICU == "" | is.na(Ever.in.ICU)),
         !( Client.Gender == "NOT LISTED, PLEASE SPECIFY" |
             Client.Gender == "UNKNOWN" | is.na(Client.Gender)),
         !( Outcome == "ACTIVE" | ## Drop Active Cases
             Outcome == "" | is.na(Outcome)),
         !(Source.of.Infection == "No Information" ## Drop entries w/o info
           |Source.of.Infection == "Pending"| is.na(Source.of.Infection)))

# Checking for any NA entries
sum(is.na(clean_data))

# Treat Categorical Variables as factors
clean_data$Age.Group <- as.factor(clean_data$Age.Group)
clean_data$Ever.in.ICU <- as.factor(clean_data$Ever.in.ICU)
clean_data$Ever.Hospitalized <- as.factor(clean_data$Ever.Hospitalized)
clean_data$Source.of.Infection <- as.factor(clean_data$Source.of.Infection)
clean_data$Client.Gender <- as.factor(clean_data$Client.Gender)
clean_data$Classification <- as.factor(clean_data$Classification)
clean_data$Outbreak.Associated <- as.factor(clean_data$Outbreak.Associated)

# Some of the levels for a couple variables
levels(clean_data$Outbreak.Associated)
levels(clean_data$Client.Gender)
```

```r
group_by_age_group = clean_data %>%
  mutate(age_group = if_else(Age.Group == "19 and younger", "<20",
                       if_else(Age.Group == "20 to 29 Years", "20-29",
                        if_else(Age.Group == "30 to 39 Years", "30-39",
                         if_else(Age.Group == "40 to 49 Years", "40-49",
                          if_else(Age.Group == "50 to 59 Years", "50-59",
                           if_else(Age.Group == "60 to 69 Years", "60-69",
                            if_else(Age.Group == "70 to 79 Years", "70-79",
                             if_else(Age.Group == "80 to 89 Years", "80-89",
                              if_else(Age.Group == "90 and older", "90+", Age.Group)))))))))) %>%
  group_by(age_group) %>%
  summarise(fatal = sum(Outcome == "FATAL"),
            resolved = sum(Outcome == "RESOLVED"),
            non_active_cases = sum(Outcome != "ACTIVE"),
            fatality_rate = fatal / (resolved + fatal))
group_by_source_of_infection = clean_data %>%
  filter(Source.of.Infection != "Pending") %>%
  group_by(Source.of.Infection) %>%
  summarise(fatal = sum(Outcome == "FATAL"),
            resolved = sum(Outcome == "RESOLVED"),
            non_active_cases = sum(Outcome != "ACTIVE"),
            fatality_rate = fatal / (resolved + fatal)) %>%
  arrange(fatality_rate)
group_by_outbreak_association = clean_data %>%
  group_by(Outbreak.Associated) %>%
  summarise(fatal = sum(Outcome == "FATAL"),
            resolved = sum(Outcome == "RESOLVED"),
            non_active_cases = sum(Outcome != "ACTIVE"),
            fatality_rate = fatal / (resolved + fatal))
group_by_ever_in_icu = clean_data %>%
  group_by(Ever.in.ICU) %>%
  summarise(fatal = sum(Outcome == "FATAL"),
            resolved = sum(Outcome == "RESOLVED"),
            non_active_cases = sum(Outcome != "ACTIVE"),
            fatality_rate = fatal / (resolved + fatal))

group_by_ever_hospitalized = clean_data %>%
  group_by(Ever.Hospitalized) %>%
  summarise(fatal = sum(Outcome == "FATAL"),
            resolved = sum(Outcome == "RESOLVED"),
            non_active_cases = sum(Outcome != "ACTIVE"),
            fatality_rate = fatal / (resolved + fatal))
group_by_gender = clean_data %>%
  filter(Client.Gender %in% c("MALE", "FEMALE")) %>%
  mutate(gender = if_else(Client.Gender == "FEMALE", "Female",
                      if_else(Client.Gender == "MALE", "Male", Client.Gender))) %>%
  group_by(gender) %>%
  summarise(fatal = sum(Outcome == "FATAL"),
            resolved = sum(Outcome == "RESOLVED"),
            non_active_cases = sum(Outcome != "ACTIVE"),
            fatality_rate = fatal / (resolved + fatal))

p1 <- ggplot(group_by_ever_in_icu, aes(x = Ever.in.ICU, y = fatality_rate)) +
```

```
  geom_bar(aes(fill = Ever.in.ICU), stat = "identity", show.legend = F) +
  labs(x = "Ever in ICU", y = "Fatality Rate")
p2 <- ggplot(group_by_ever_hospitalized, aes(x = Ever.Hospitalized, y = fatality_rate)) +
  geom_bar(aes(fill = Ever.Hospitalized), stat = "identity", show.legend = F) +
  labs(x = "Ever Hospitalized", y = "Fatality Rate")
p3 <- ggplot(group_by_gender, aes(x = gender, y = fatality_rate)) +
  geom_bar(aes(fill = gender), stat = "identity", show.legend = F) +
  labs(x = "Gender", y = "Fatality Rate")
p4 <- ggplot(group_by_source_of_infection, aes(x = reorder(Source.of.Infection, fatality_rate), y = fata
  geom_bar(aes(fill = Source.of.Infection), stat = "identity", show.legend = F) +
  labs(x = "Source of Infection", y = "Fatality Rate", title = "Fatality Rate by Source of Infection") +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.9, size = 9))
p5 <- ggplot(group_by_age_group, aes(x = age_group, y = fatality_rate)) +
  geom_bar(aes(fill = age_group), stat = "identity", show.legend = F) +
  labs(x = "Age Group", y = "Fatality Rate", title = "Fatality Rate by Age Group")
p6 <- ggplot(group_by_outbreak_association, aes(x = Outbreak.Associated, y = fatality_rate)) +
  geom_bar(aes(fill = Outbreak.Associated), stat = "identity", show.legend = F) +
  labs(x = "Outbreak Association", y = "Fatality Rate", title = "Fatality Rate by Outbreak Association"

grid.arrange(p1, p2, p3, ncol = 3)
grid.arrange(p6, p5, ncol= 1, nrow=2)
p4
clean_data %>%
  # cross-tabulate counts of two columns
  tabyl(Age.Group, Outcome) %>%
  # add a total row, add a total column
  adorn_totals(where = c("row", "col")) %>%
  # convert to proportions with row denominator
  adorn_percentages(denominator = "row") %>%
  # convert proportions to percents
  adorn_pct_formatting() %>%
  # display as: "count (percent)"
  adorn_ns(position = "front") %>%
  # adjust titles
  adorn_title(
    row_name = "Age Group",
    col_name = "Fatal(0) / Resolved(1)") %>%
  # print elegant results for interactive analysis or for sharing in a report
  # e.g., with knitr::kable()
  knitr::kable(format = "simple",caption = 'Contigency Table of Fatality "
               vs Age Group') %>%
  row_spec(0, bold = T, background = "white") %>%
  kable_styling(latex_options = "HOLD_position")


# Create contingency table for Age.Group and Client.Gender
age_outcome_table <- table(clean_data$Age.Group, clean_data$Outcome)

# Perform Chi-squared test of independence
age_outcome_chi_square <- chisq.test(age_outcome_table)

# Print the result
print(age_outcome_chi_square) ## Age and Outcome are not independent since
```

```r
set.seed(123)
## Split data into training and testing set (Cross-Validation)
train_index <- createDataPartition(clean_data$Outcome, p = 0.8, list = FALSE)
train <- clean_data[train_index, ]
test <- clean_data[-train_index, ]

# Factor "Outcome" variable as 1 (Resolved) or 0 (Fatal)
clean_data$Outcome<- ifelse(clean_data$Outcome == "RESOLVED", 1, 0)
train$Outcome <- ifelse(train$Outcome == "RESOLVED", 1, 0)
test$Outcome <- ifelse(test$Outcome == "RESOLVED", 1, 0)

# Fit the logistic regression model
model <- glm(Outcome ~ ., data = train, family = "binomial")
# summary(model)
model$aic

# Step-wise fit
initial_model <- glm(Outcome ~ 1, data =train, family = "binomial")

step_model_both <- step(model, direction = "both", trace = 0)
step_model_backward <- step(model, direction = "backward", trace = 0)
step_model_forward <- step(model, direction = "forward",
                           scope = list(lower = initial_model, upper = model),
                           trace = 0)

step_model_both$aic
step_model_backward$aic
step_model_forward$aic

## Backward and Bi-directional have the lowest AIC's and offer the same final
## model which excludes the variables, "Cases", "Ever.Hospitalized".
# Assess the effect of dropping each predictor variable
dropped_variable_effects <- drop1(model, test = "Chi")

## Setting alpha at 0.01, "Client.Gender" appears to not be a significant
## predictor variable so we drop it to see how the model might improve.

# Manually Selected Model
model2 <- glm(Outcome ~ Age.Group + Ever.in.ICU + Source.of.Infection
              + Outbreak.Associated + Classification
    , data = train, family = "binomial")
model2$aic

## AIC actually increases when we exclude the gender variable

test$Outcome <- as.factor(test$Outcome)
train$Outcome <- as.factor(train$Outcome)

# Evaluate the model's performance on test data (Step-wise Model)
test$predicted_prob <- predict(step_model_both, test, type = "response")
test$predicted_outcome <- as.factor(ifelse(test$predicted_prob > 0.5, 1, 0))

# Check Predictions for Full Model with all variables
```

```r
test$predicted_prob2 <- predict(model2, newdata = test, type = "response")
test$predicted_outcome2 <- as.factor(ifelse(test$predicted_prob2 > 0.5, 1, 0))

# Performance metrics (Step Wise)
confusion_matrix <- confusionMatrix(test$Outcome, test$predicted_outcome)
print(confusion_matrix)

accuracy <- confusion_matrix$overall["Accuracy"]
precision <- confusion_matrix$byClass["Pos Pred Value"] ## Accurate Fatal Pred
recall <- confusion_matrix$byClass["Sensitivity"]
f1_score <- confusion_matrix$byClass["F1"] ## Measures Precision and Accuracy
print(paste("F1-score:", f1_score))


# Performance metrics (Full Model)
confusion_matrix2 <- confusionMatrix(test$Outcome, test$predicted_outcome2)
accuracy2 <- confusion_matrix2$overall["Accuracy"]
precision2 <- confusion_matrix2$byClass["Pos Pred Value"]##Accurately detect bad
recall2 <- confusion_matrix2$byClass["Sensitivity"]
f1_score2 <- confusion_matrix2$byClass["F1"]
print(paste("F1-score:", f1_score2))


# ROC curve (Step wise fit)
roc_obj <- roc(test$Outcome, test$predicted_prob)
auc <- pROC::auc(roc_obj)
plot(roc_obj, print.auc = TRUE, print.auc.corners = TRUE)
summary(step_model_both)
# Step wise model with lowest AIC
final_model <- glm(Outcome ~ Age.Group + Ever.in.ICU + Source.of.Infection +
                     Client.Gender + Outbreak.Associated + Classification,
                   family = "binomial", data = train)
# Number of samples
B <- 5
coefs_boot <- matrix(NA, nrow = B, ncol = length(coef(final_model)))
set.seed(123) # for reproducibility

for (i in 1:B) {
  # Resample the dataset
  data_boot <- train[sample(nrow(clean_data), replace = TRUE),]

  # Fit the logistic regression model
  model_boot <- update(final_model, data = data_boot)

  coefs_boot[i, ] <- coef(model_boot)
}

# Calculate the 95% confidence intervals for each model coefficient
conf_intervals <- apply(coefs_boot, 2, function(x) {
  quantile(x, probs = c(0.025, 0.975))
})

# Print confidence intervals
```

```r
# print(conf_intervals)

# Get the original coefficients from final model
orig_coefs <- coef(final_model)

# Compare the original coefficients and bootstrap confidence intervals
coef_comparison <- data.frame(
  Original = orig_coefs,
  LowerCI = conf_intervals[1,],
  UpperCI = conf_intervals[2,]
)

# Evaluate if original coefficient is within the confidence interval
coef_comparison$InCI <- coef_comparison$Original >= coef_comparison$LowerCI &
                        coef_comparison$Original <= coef_comparison$UpperCI

# Print the comparison
interval_table <- data.frame(
  Coefficient = coef_comparison[0],
  Within_Bootstrap_Interval = coef_comparison$InCI
)
print(interval_table) ## All estimates are within the 95% bootstrap C.I's
```