Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

GSB 530 Group Project

Winter 2023

Business understanding

The data collected by the Israeli Ministry of Health is critical in developing a better understanding of the main characteristics and symptoms associated with COVID-19 cases. This understanding enables the Ministry to develop efficient, data-driven plans and make informed decisions in their fight against COVID-19. By analyzing the data on the main symptoms associated with COVID-19 cases for both males and females separately, with a special focus on elderly people who are at greater risk, the Ministry can gain a more nuanced understanding of how the disease affects different populations groups. By analyzing the data, the Ministry can prioritize testing for populations that are most likely to be infected or transmit the virus and allocate testing resources accordingly. Similarly, data on symptoms can help guide decisions around early quarantine and hospitalization, ensuring that individuals receive the care they need and reducing the risk of transmission Theldis is especially important during times when the healthcare system, including drug supply, hospitalization, and lab capacities, is overwhelmed and at stake.

In addition, providing accurate guidelines to the public for self-diagnosis can be highly beneficial. This would allow individuals and businesses to adjust their early detection mechanisms and work from home policies, as well as take the most efficient preventive actions to limit the spread of COVID-19. Moreover, data associated with exposure to individuals with COVID-19 can help uncover the mysteries behind the transmission and spread mechanism of the virus, including surface transmission and airborne possibilities. This information could be used to develop more effective measures to mitigate the spread of the virus, especially in high-risk areas such as hospitals, nursing homes, and other crowded settings.

Lastly, it is crucial to determine if the people are more susceptible to COVID-19 compared to younger age groups. Such knowledge would be useful for decisions regarding the implementation of strict infection control practices and monitoring in nursing homes and other facilities with high elderly populations. By identifying this vulnerable population, the authorities can prioritize the allocation of resources such as PPE, ventilators, and medical staff to prevent and manage outbreaks in these facilities.

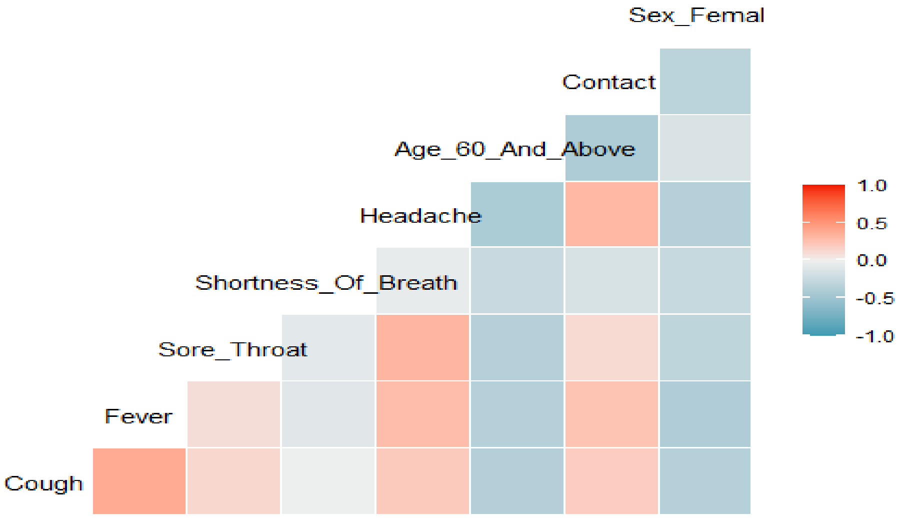
As illustrated, the provided dataset presents numerous business opportunities that can be explored through data mining and analytics techniques. To achieve this, we have identified six critical business questions that we aim to answer:

1) Are elderly people more susceptible to covid19 infection?
2) Are males or females more vulnerable to covid19 infection?
3) Does being in contact with a person with Covid-19 significantly increase the risk of getting the infection?
4) What are the primary and most alarming symptoms or symptoms combination associated with high probability of covid19 cases?
5) Can we use a model to predict whether a random person has Covid or not?
6) Do the data suggest that COVID-19 **symptoms** vary between males and females, and between elderly people and other younger age groups? If the answer is yes, how are the symptom or symptom probability different between those groups/clusters.

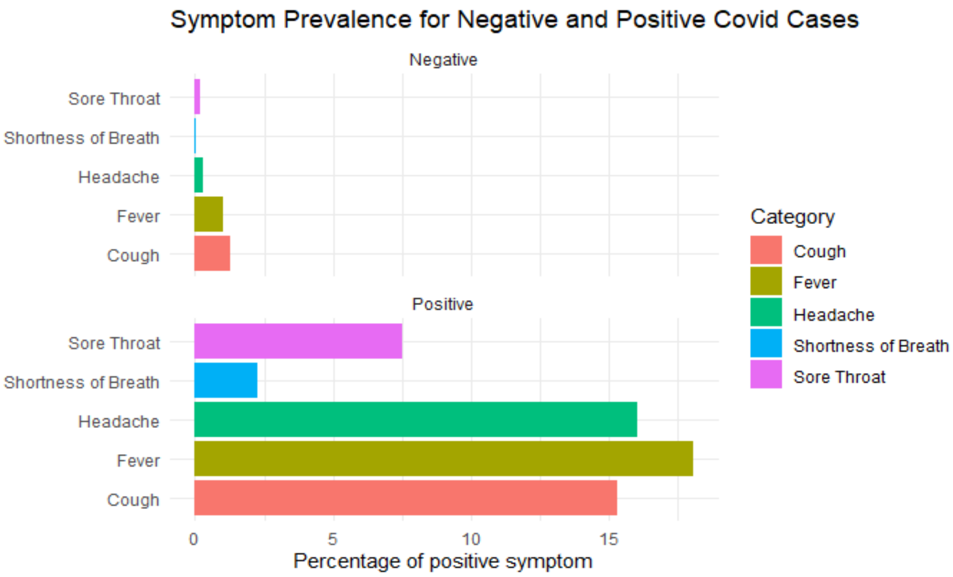
Data understanding

The Covid_Testing dataset contains 1 million rows and 9 columns of data on Covid-19 test results. Each row corresponds to a unique individual tested for Covid-19, with columns providing information on symptoms, age, gender, and contact history. The first 5 columns indicate if the person exhibited symptoms, while Age_60_And_Above and Gender columns show the individual's age and gender. Contact column indicates contact with Covid-19 positive person, and Result column shows test outcome (1 = positive, 0 = negative). The dataset is used to study relationships between symptoms and Covid-19 infection, and to identify risk factors for infection.

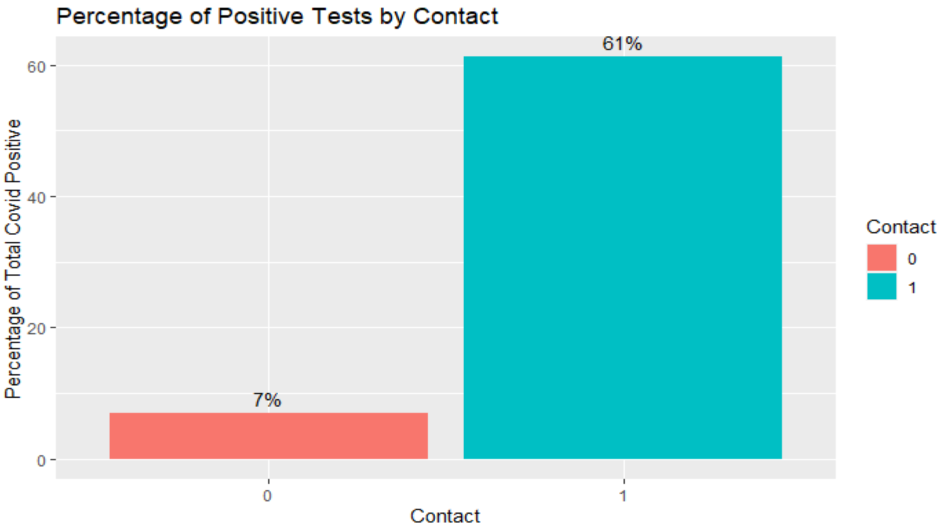
Data exploration on a Covid positive dataset reveals several interesting findings. Firstly, a correlation plot shows a high positive correlation between cough and fever, while contact with an infected person correlates highly with symptoms such as cough, fever, and sore throat.



Secondly, a bar chart comparing symptom prevalence between negative and positive Covid cases indicates that individuals who test positive for Covid report symptoms such as headache, fever, and cough more than 15% of the time, whereas those with negative results report symptoms less than 5% of the time.



Furthermore, analysis of the dataset suggests that coming in contact with an infected person greatly increases the likelihood of testing positive for Covid. Specifically, 61% of people who had contact with an infected person tested positive, compared to only 7% of those who did not have contact with an infected person.



Data preparation

To prepare our data for analysis we first turned the variable *Age_60_And_Abov*e into a dummy variable that equals 1 if the respondent was 60 or above and 0 otherwise. Next, we created the variable *Male*, which equals 1 if the sex of the respondent was male and 0 if female. Finally, we created a variable called *Positive* that equals 1 if the respondent's result was positive and 0 if otherwise. After completing this we removed NAs from our data set. This took our observations from 1030724 to 978493, meaning 52231 (~5%) observations were lost which should not undermine our analysis.

Modeling

In this step, our objective was to identify the most appropriate binary classification models that align with the objectives we identified in the business understanding section. Our ideal selected model or models should be capable of achieving the following objectives.

(a) Provide the **most accurate predictions** for COVID-19 cases, with a special focus on the target class cases. This is particularly important as the misclassification cost of the target class is significantly higher than that of the non-target class. In other words, the cost of missing an actual positive COVID-19 case is much greater than that of wrongly classifying a non-positive COVID-19 case as a potential COVID-19 case.

(b) An interpretable model that would help us identify the contribution of each variable to probability of being a positive COVID-19 case.

(c) Determine whether different age and sex groups exhibit distinct clusters in terms of positive COVID-19 symptoms.

Covid19 Prediction and Symptom Significance Analysis

For the objectives related to prediction accuracy and symptom contribution interpretability, we tried all below models to reach the best covid19 predictability, yet our focus was more on logit and decision tree model for easier interpretability. Additionally, for the model evaluation and confusion matrix calculation, we considered a customized cutoff probability threshold to enhance the target case prediction accuracy. After trying several cutoff values our optimum threshold for better sensitivity, with acceptable specificity and accuracy values was **10% (0.01)**, which matches our target class cases ratio in the provided dataset.

a. Logistic Regression
b. Decision Tree and Ensemble tree (Bagging and Random Forest), yet we excluded boosting tree because it tends to overfit and needs to much processing for large data set
c. Naïve Bayes.

Naïve Bayes

The model that we created had a very great sensitivity. It gave a sensitivity of .9998 (please refer to figure below). Although this is fantastic, our specificity was horrible. The specificity was .0069 (please refer to figure below). We decided to drop this model altogether because even though it would be very valuable to accurately identify all people who are infected with the virus, the specificity is far too low for the model to be taken seriously as the best choice. The results that are shown in the figure below reflect the output of the model which was ran on the full dataset. In addition to this, we ran the model on multiple subsets consisting of only males, only females, people over the age of 60, people under the age of 60, and covid positive people. In any case, the results were negligible.

Model Name	Hyper parameter	Sensitivity	Specificity	Accuracy	Roc-Auc
Naïve Bayes	-	0.9998	0.0069	0.8952	-

Logistic Classification Model

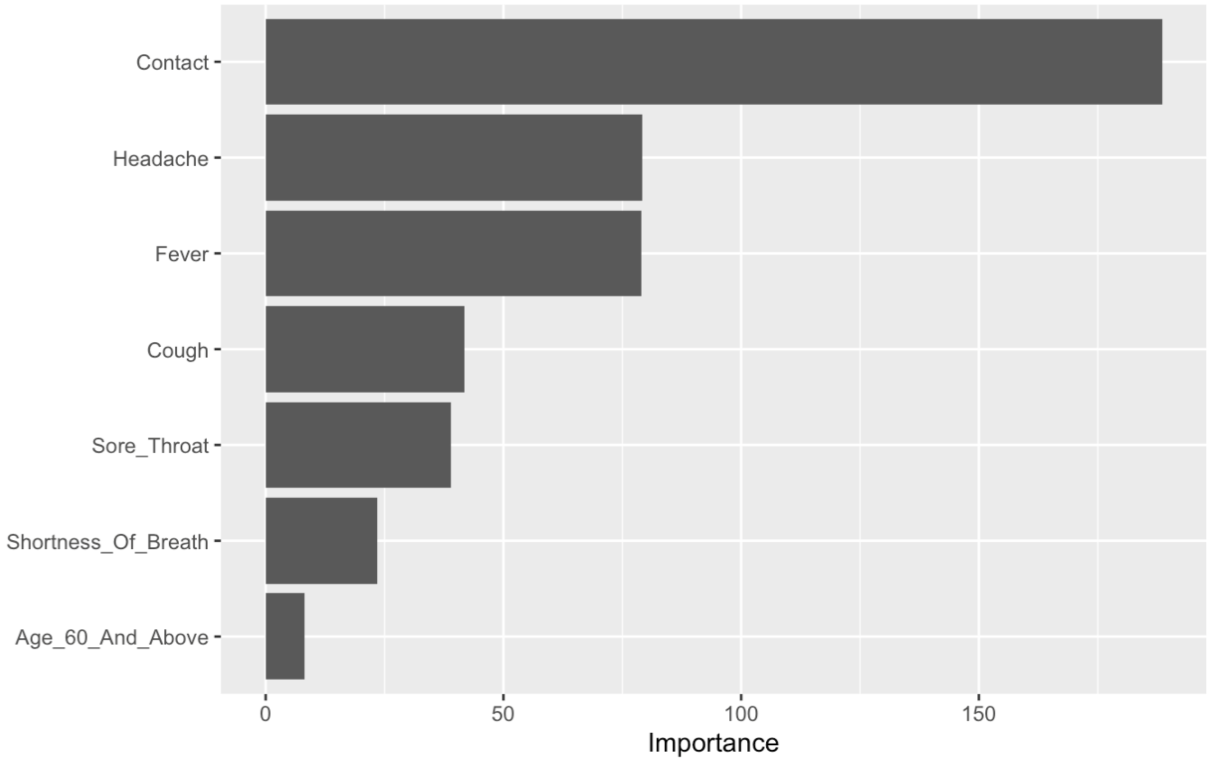
Predictive analytics and categorization frequently make use of this kind of statistical model, also referred to as a logit model. Based on a given dataset of independent variables, logistic regression calculates the likelihood that an event will occur, such as voting or not voting. The logistic model was able to classify truly positive individuals with a positive test result with a 95.70% of probability. While the probability of classifying truly negative individuals as negative was 53.46%.

Model Name	Hyper parameter	Sensitivity	Specificity	Accuracy	ROC-AUC
Logistic Classification	-	0.9569	0.5439	0.9134	0.7504

From below Logistic classification output all of the predictor variables are highly significant with a very low p-value.

Predictors	Estimate	P-value	Exponent of Estimate
Intercept	-2.92526	0.00826	
Cough	0.98568	2E-16	2.679633
Fever	1.79689	2E-16	6.030862
Sore Throat	1.86709	2E-16	6.469443
Shortness of Breath	2.0407	2E-16	7.695995
Headache	2.57768	2E-16	13.16656
Age 60 And Above	0.11915	4.64E-16	1.126539
Contact	2.509	2E-16	12.29263
Male	0.26106	2E-16	1.298306

Looking at the variable importance we found that Contact, Headache, and Fever turned out to be symptoms with higher importance with respect to the whole dataset.

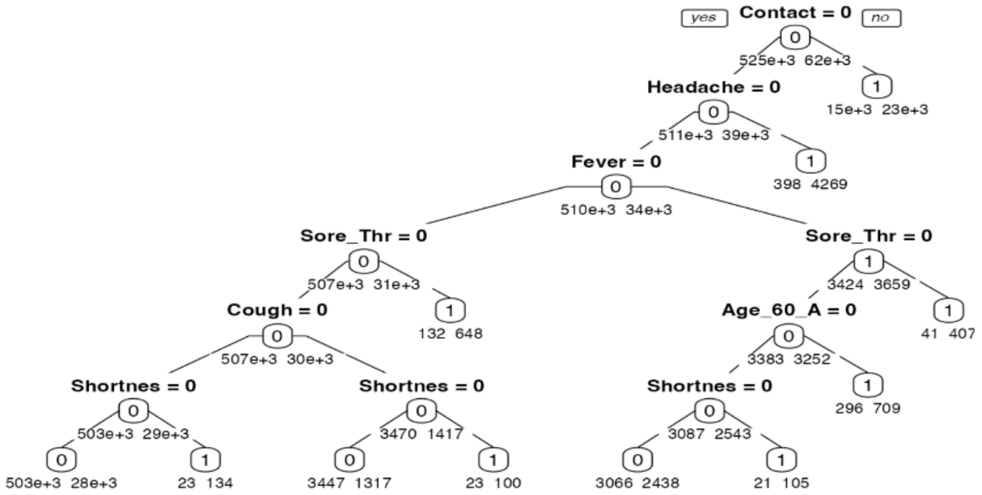


Decision Tree and Ensemble tree Model

We considered the decision tree and ensemble models as being one of the most powerful classification models. Decision tree comes best when it comes to interpretation, understanding variables relationships and importance. On the other hand, ensemble trees help identify complex patterns within the data and avoid overfitting.

For decision tree we identified the best pruned tree by selecting the optimal complexity (CP = 4.5293e-04) providing cross validated classification error within one standard deviation of the least cross validated error (xerror = 0.77334)

Model Name	Hyper parameter	Sensitivity	Specificity	Accuracy	ROC-AUC
Decision Tree	Cp: 4.5293e-04	0.54387	0.95686	0.9134	0.7538



For the ensemble tree models, we utilized bagging to capture any relation or patterns within a small subset of our dataset that might be hindered by the vast majority of the dataset pattern, additionally we utilized the Random Forest technique to further detect any underrepresented pattern in some of our explanatory variables. Below is a table with the summary of the model parameters and results.

Model Name	Hyper parameter	Sensitivity	Specificity	Accuracy	ROC-AUC
Bagging	ntree= 100, mtry = 8	0.43055	0.97640	0.9189	0.7418
Random Forest	ntree= 100, mtry = 3	0.52822	0.96297	0.9172	0.7491

Best Supervised Model Selection

Model Name	Hyper parameter	Sensitivity	Specificity	Accuracy	ROC-AUC
Naïve Bayes	-	0.998	0.0069	0.8952	-
Logistic Classification	-	**0.9570**	0.5346	0.9152	0..7504
Decision Tree	Ntree = 100, mtry = 8	0.43055	0.97640	0.9189	0.7538
Bagging	ntree= 100, mtry = 8	0.43055	0.97640	0.9189	0.7418
Random Forest	ntree= 100, mtry = 3	0.52822	0.96297	0.9172	0.7491

After implementing the above supervised models it is clear that the logistic classification model is the best for our use case. Given that our model is a binary choice model, a measure such as R^2 is not useful for determining model goodness of fit. Thus, the metrics we focused on were accuracy, specificity and sensitivity. The accuracy rates of most of our models were very similar, with Naïve Bayes having a slightly lower accuracy rate than the other models. Since we are making predictions on whether a person has COVID-19, a virus that is particularly contagious due to its spread via airborne particles, maximizing the number of positive outcomes that are classified correctly is crucial. In our case, to do our best in slowing down the spread of the virus, it is important to correctly identify all people who have the virus and what symptoms make one most susceptible to the disease, thus sensitivity is a key metric to consider. We also focused on specificity to ensure that people who are not sick can go on with their lives. It is important to accurately identify the people who are without the disease. As our data is linearly separable and the outcome is binary or dichotomous in nature, the logistic model performed as expected.

Covid19 Sex and Age Group clustering

On the other side, to determine whether different age and sex groups have different symptoms we utilized the K-Means clustering model, in addition to the regular data wrangling and descriptive analysis techniques.

K-Means clustering results

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

We implemented the K-means clustering on positive cases to see if we will have two district clusters for the symptoms associated with covid-19 infection for males versus females, or elders versus younger people.

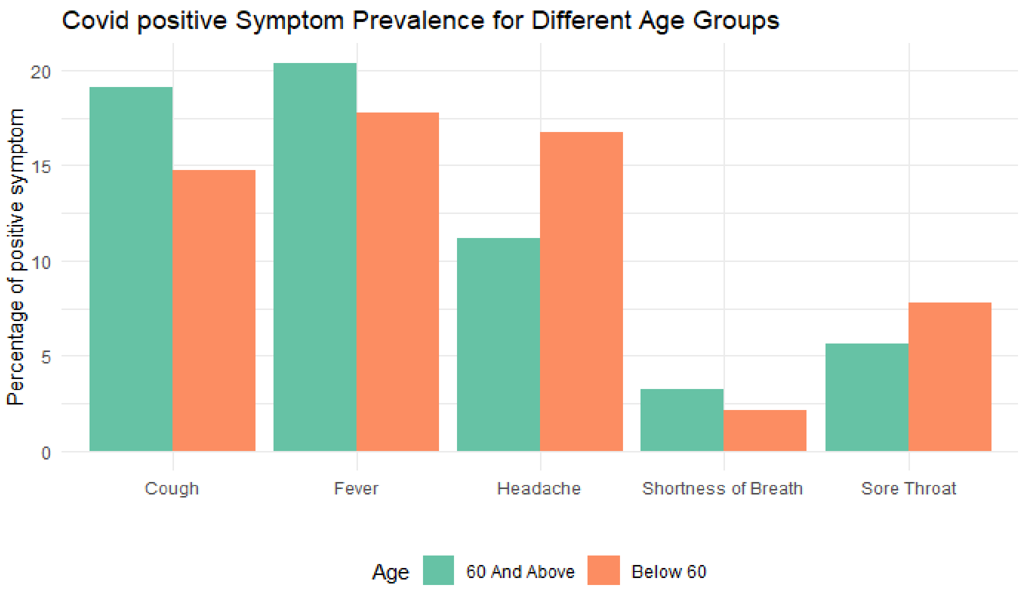
The k-means cluster was able to provide two distinct cluster for males and females, showing that infected Females tend to show high cough, sore throat and headache symptoms compared to men who were showing higher fever symptoms.

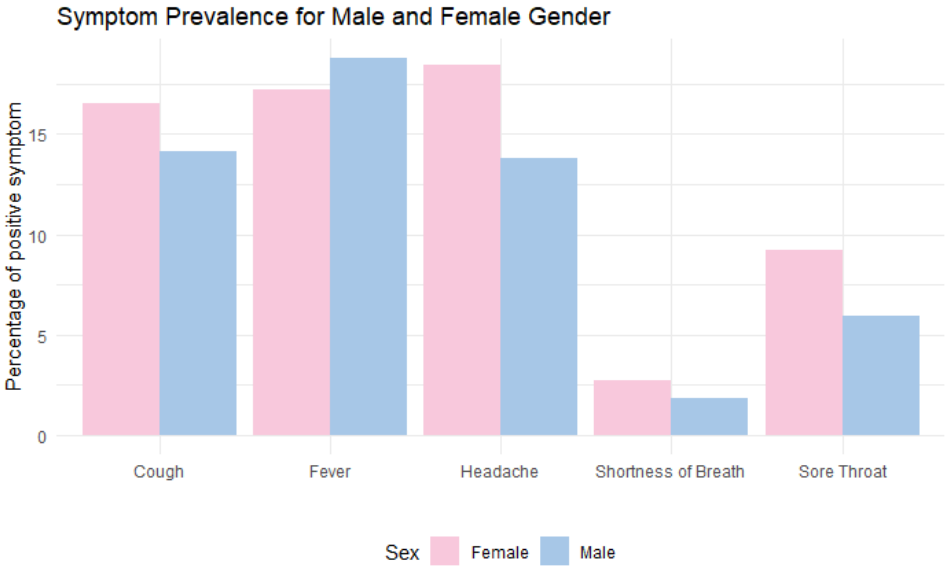
K-Mean Clusters	Cough	Fever	Sore_Throat	Shortness_Of_Breath	Headache	Age_Above_60	Male
1 (Female)	17%	17%	9%	3%	18%	12%	0
2 (Males)	14%	19%	6%	2%	14%	13%	1

Descriptive analytics results

The graph of symptom prevalence for different age groups indicates that people below the age of 60 who tested positive for Covid are more likely to experience headache and sore throat and the graph of symptom prevalence for different genders show that men who tested positive for Covid are more likely to experience fever.

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi





Evaluation

To reinstate, the goal of this analysis was to help the Israeli Ministry of Health to develop efficient, data-driven plans, to make informed decisions in their fight against COVID-19. By analyzing data on the main symptoms of COVID-19, especially with regards to gender, age and contact with a covid positive person, the Ministry could gain a better understand of how the disease affects different population groups. In doing so, the Ministry could implement prioritized testing for higher risk participants, guide early quarantine/hospitalization and provide better guidance on self-diagnosis, Moreover, business could better implement work from home policies, all the while uncovering the mysteries behind the transmission and spread mechanism of the virus.

The logistic regression model achieves our business objectives by answering all our business questions in detail. It predicts that people with a cough, fever, sore throat, shortness of breath and headache have 2.7, 6.0, 6.5, 7.7 and 13.2 times the odds of being covid positive respectively. Moreover, with regards to age, for people above 60, the model predicted an odds ratio of 1.13, meaning people above 60 have 1.13 times the odds of being covid positive. With regards to people who have any contact with a covid positive person, an odds ratio of 12.29 means that people have 12.29 times the odds of being covid positive. Finally, males have an odds ratio of 1.3, meaning, males have 1.3 times the odds of being covid positive over females. By attaining a sensitivity rate of 95.69%, the model is not only confident in maximizing the number of positive outcomes that are classified correctly but also in achieving the business objectives successfully.

Additionally, using K-means clustering and descriptive analytics, it became evident that symptoms for males and females as well as symptoms for people above and below 60 differ. With regards to gender, females who test positive for covid tend to have a higher percentage of cough, sore throat and headaches, while males who test positive for covid tend to have a higher percentage of fever. Moreover, with regards to age, people over 60 who test positive for covid tend to show a higher percentage of cough, fever and shortness of breath, while people under the age of 60 who test positive for covid tend to show a higher percentage of headache and sore throat. With this information, we are now capable of providing the Israeli Ministry of Health with recommendations for tacking this deadly virus.

Deployment

Recommendations for the Israeli Ministry of Health:

1. Publish the findings of this analysis on the Ministry of Health website and other public websites.
2. Prioritize testing for symptoms of headaches and shortness of breath over cough, fever and sore throat.
3. Make quarantining mandatory for at least 2 weeks after a positive covid result.
4. Reinforce vaccinations and masks in public spaces as contact with an infected person is the most important predictor of having a positive covid result.
5. Treat gender and age similarly when testing for covid but not after a person receives a positive test result.
6. For a positive covid test result, treat females for cough, sore throat and headaches more often and males for fever more often.
7. For a positive covid test result, treat people above 60 for cough, fever and shortness of breath more often and people under 60 for headaches and sore throat more often

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

Appendix (R Code)

Data Understanding

```
#Import dataset
Covid_Testing <- Big_Data_Files

#Data Preparation
myData <- Covid_Testing %>%
  drop_na() %>%
  mutate(
    Age_60_And_Above = ifelse(Age_60_And_Above == "No",0,1),
    Sex_Female = ifelse(Sex == "male",0,1),
    Result = ifelse(Result == "positive",1,0)
  )
head(myData)

#Correlation Plot between binary variables
library(ggplot2)
library(corrplot)
library(GGally)
cor_data <- myData[myData$Result == "1", ]
cor_data <- myData %>%
  select(-Sex, -Result)
cor_mat <- cor(cor_data)
ggcorr(cor_mat)


#Faceted bar chat showing symptom prevalence for negative and positive
# Filter data for negative results
CovidNegData <- myData[myData$Result == "0", ]
# Filter data for positive results
CovidPosData <- myData[myData$Result == "1", ]
# Function to calculate percentage of symptoms in data
```

```
calculate_symptom_percentage <- function(data) {
  data_len <- nrow(data)
  cough <- round(nrow(data[data$Cough == "1", ])/ data_len*100, 2)
  fever <- round(nrow(data[data$Fever == "1", ])/ data_len*100, 2)
  soreThroat <- round(nrow(data[data$Sore_Throat == "1", ])/ data_len*100, 2)
  shortnessOfBreath <- round(nrow(data[data$Shortness_Of_Breath == "1", ])/ data_len*100,
2)
  headache <- round(nrow(data[data$Headache == "1", ])/ data_len*100, 2)

  # Data frame with the values
  df <- data.frame(Category = c("Cough", "Fever", "Sore Throat", "Shortness of Breath",
"Headache"),
                   Percentage = c(cough, fever, soreThroat, shortnessOfBreath, headache))
  return(df)
}
# symptom percentage for negative results
CovidNegSymptoms <- calculate_symptom_percentage(CovidNegData)
# symptom percentage for positive results
CovidPosSymptoms <- calculate_symptom_percentage(CovidPosData)
# Combined data
combined_data <- rbind(cbind(CovidNegSymptoms, Result = "Negative"),
                       cbind(CovidPosSymptoms, Result = "Positive"))
# plot
ggplot(combined_data, aes(x = Category, y = Percentage, fill = Category)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  facet_wrap(~ Result, ncol = 1) +
  labs(x = "", y = "Percentage of positive symptom", title = "Symptom Prevalence for Negative
and Positive Covid Cases") +
  theme_minimal()

# Bar chart for plotting percentage of positive tests by contact
Contact_count <- table(myData$Contact)
filtered_data <- myData[myData$Result == "1", ]
grouped_count <- table(filtered_data$Contact)
percentages <- grouped_count / Contact_count * 100
df <- data.frame(Contact = names(percentages), Percentage = percentages)
ggplot(df, aes(x = Contact, y = Percentage.Freq, fill = Contact)) +
  geom_bar(stat = "identity") +
  xlab("Contact") +
  ylab("Percentage of Total Covid Positive") +
  ggtitle("Percentage of Positive Tests by Contact") +
  geom_text(aes(label = paste0(round(Percentage.Freq), "%")), vjust = -0.5)


#Graph for Covid positive symptom prevalence for different age groups
# Filter data for negative results
Below60Data <- CovidPosData[CovidPosData$Age_60_And_Above == "0", ]
```

```
# Filter data for positive results
Age_60_And_AboveData <- CovidPosData[CovidPosData$Age_60_And_Above == "1", ]
# symptom percentage for negative results
Below60Symptoms <- calculate_symptom_percentage(Below60Data)
# symptom percentage for positive results
Age_60_And_AboveSymptoms <- calculate_symptom_percentage(Age_60_And_AboveData)
# Combined data
combined_data <- rbind(cbind(Below60Symptoms, Age = "Below 60"),
                       cbind(Age_60_And_AboveSymptoms, Age = "60 And Above"))
ggplot(combined_data, aes(x = Category, y = Percentage, fill = Age)) +
  geom_col(position = "dodge") +
  labs(x = "", y = "Percentage of positive symptom", title = "Covid positive Symptom Prevalence for Different Age Groups") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

#Graph for Covid positive symptom prevalence for male and female gender

```
# Filter data for negative results
FemaleData <- CovidPosData[CovidPosData$Sex == "female", ]
# Filter data for positive results
MaleData <- CovidPosData[CovidPosData$Sex == "male", ]

# symptom percentage for negative results
FemaleDataSymptoms <- calculate_symptom_percentage(FemaleData)

# symptom percentage for positive results
MaleDataSymptoms <- calculate_symptom_percentage(MaleData)

# Combined data
combined_data <- rbind(cbind(FemaleDataSymptoms, Sex = "Female"),
                       cbind(MaleDataSymptoms, Sex = "Male"))

ggplot(combined_data, aes(x = Category, y = Percentage, fill = Sex)) +
  geom_col(position = "dodge") +
  labs(x = "", y = "Percentage of positive symptom", title = "Symptom Prevalence for Male and Female Gender") +
  scale_fill_manual(values = c("#F8C8DC", "#A7C7E7")) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

Decision Tree and Ensemble trees models

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

Same code was repeating considering data subset for different population groups (female, male, Age_Above_60 and Age_Below_60)to stand apon the difference in symptoms probability and if we can have better prediction model for each group separately

```
##################################### Male + Female
#Data loading and factorization
myData <- COVID_TESTING_DATASET
myData$Age_60_And_Above <- ifelse(myData$Age_60_And_Above =='Yes', 1,0)
myData$Female <- ifelse(myData$Sex =='female', 1,0)
myData$Result <- ifelse(myData$Result =='positive', 1,0)
myData <- myData %>%
  select(-Sex) %>%
  mutate_all(factor) %>%
na.omit()
myData <- data.frame(myData)
# Partitioning
suppressWarnings(RNGversion("3.5.3"))
set.seed(1)
myIndex <- createDataPartition(myData$Result, p=0.6, list=FALSE)
trainSet <- myData[myIndex,]
validationSet <- myData[-myIndex,]
########### Decision Tree
set.seed(1)
default_tree <- rpart(Result ~., data = trainSet, method = "class")
summary(default_tree)
prp(default_tree, type = 1, extra = 1, under = TRUE)
# Tree pruning
set.seed(1)
full_tree <- rpart(Result ~ ., data = trainSet, method = "class", cp = 0, minsplit = 2, minbucket
= 1)
prp(full_tree, type = 1, extra = 1, under = TRUE)
printcp(full_tree)
pruned_tree <- rpart::prune(full_tree, cp = 4.5293e-04)
prp(pruned_tree, type = 1, extra = 1, under = TRUE)
# Evaluation on validation dataset
predicted_class <- predict(pruned_tree, validationSet, type = "class")
confusionMatrix(predicted_class, validationSet$Result, positive = "1")
# updating confusion Matrix based on 0.10 cutoff Positive Covid19 cases ratio =
105434/(925290+105434 = 0.1023
predicted_prob <- predict(pruned_tree, validationSet, type= 'prob')
confusionMatrix(as.factor(ifelse(predicted_prob[,2]>0.10, '1', '0')), validationSet$Result,
positive = '1')
dplyr::arrange(varImp(pruned_tree, type = 1), desc(Overall))
# ROC-AUC curve
roc_object <- roc(validationSet$Result, predicted_prob[,2])
plot.roc(roc_object)
```

```
auc(roc_object)

########## Bagging Tree
set.seed(1)
bagging_tree <- randomForest(Result ~., data = trainSet, ntree = 100, mtry = 8, importance =
TRUE)
predicted_class <- predict(bagging_tree, validationSet, type = "class")
confusionMatrix(predicted_class, validationSet$Result, positive = "1")
# confusion Matrix based on 0.10 cutoff
predicted_prob <- predict(bagging_tree, validationSet, type= 'prob')
confusionMatrix(as.factor(ifelse(predicted_prob[,2]>0.10, '1', '0')), validationSet$Result,
positive = '1')
#Variable importance
varImpPlot(bagging_tree, type=1)
# ROC-AUC curve
roc_object <- roc(validationSet$Result, predicted_prob[,2])
plot.roc(roc_object)
auc(roc_object)

########### Random Forest Tree
set.seed(1)
randomforest_tree <- randomForest(Result ~., data = trainSet, ntree = 100, mtry = 3,
importance = TRUE)
predicted_class <- predict(randomforest_tree, validationSet, type = "class")
confusionMatrix(predicted_class, validationSet$Result, positive = "1")
# confusion Matrix based on 0.10 cutoff
predicted_prob <- predict(randomforest_tree, validationSet, type= 'prob')
confusionMatrix(as.factor(ifelse(predicted_prob[,2]>0.10, '1', '0')), validationSet$Result,
positive = '1')
#Variable importance
varImpPlot(randomforest_tree, type=1)
 # ROC-AUC curve
predicted_prob <- predict(randomforest_tree, validationSet, type= 'prob')
roc_object <- roc(validationSet$Result, predicted_prob[,2])
plot.roc(roc_object)
auc(roc_object)
```

Naïve Bayes:

```
#install.packages('caret')

library(tidyverse)  # data manipulation
library(ggplot2)
library(readr)
library(readxl)
library(klaR)
library(caret)
library(naivebayes)
library(dplyr)
```

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

```
library(ggplot2)
#library(psych)

df = read_excel("COVID_TESTING_DATASET.xlsx")

# Clean data
df = na.omit(df)
df$Age_60_And_Above <- ifelse(df$Age_60_And_Above == "Yes", 1, 0)
df$Male <- ifelse(df$Sex == "male", 1, 0)
df$Positive <- ifelse(df$Result == "positive", 1, 0)
df$Positive <- as.factor(df$Positive)
df = subset(df, select = -c(Sex,Result) )
head(df)

set.seed(1234)
ind <- sample(2, nrow(df), replace = T, prob = c(0.6, 0.4))
train <- df[ind == 1,]
test <- df[ind == 2,]

x_test<-test[, 1:8]
y_test<-test[,9]

model <- naive_bayes(Positive ~ ., data = train, usekernel = T)
predictions<-predict(model, x_test)
confusionMatrix(predictions, y_test$Positive)
```

# K Means Clustering:

```
install.packages('factoextra')

library(tidyverse)  # data manipulation
library(ggplot2)
library(cluster)    # clustering algorithms
library(factoextra)
library(readr)
library(readxl)
library(cluster)

"""# **Full Dataset**"""

df = read_excel("COVID_TESTING_DATASET.xlsx")
df = na.omit(df)
df$Age_60_And_Above <- ifelse(df$Age_60_And_Above == "Yes", 1, 0)
df$Male <- ifelse(df$Sex == "male", 1, 0)
df$Positive <- ifelse(df$Result == "positive", 1, 0)
df = subset(df, select = -c(Sex,Result) )
```

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

```
df = df[df$Positive == 1,]
df = subset(df, select = -c(Contact) )

wss <- function(k) {
  kmeans(df, k, nstart = 10)$tot.withinss
}
k.values <- 1:10

# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")

km <- kmeans(df, centers = 2, nstart = 25)
km[2]

km <- kmeans(df, centers = 4, nstart = 25)
km[2]

"""# **Male Subset:**"""

males = df[df$Male == 1,]

# Elbow method to find best k value
set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(males, k, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 1 to k = 15
k_values <- 1:10

# extract wss for 2-15 clusters
wss_values <- map_dbl(k_values, wss)

plot(k_values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")

km <- kmeans(males, centers = 4, nstart = 25)
km[2]
```

```
"""# **Females:**"""

females = df[df$Male == 0,]

# Elbow method to find best k value
set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(females, k, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 1 to k = 15
k_values <- 1:15

# extract wss for 2-15 clusters
wss_values <- map_dbl(k_values, wss)

plot(k_values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")

km <- kmeans(females, centers = 4, nstart = 25)
km[2]

"""# **Old:**"""

old = df[df$Age_60_And_Above == 1,]

# Elbow method to find best k value
set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(old, k, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 1 to k = 15
k_values <- 1:15

# extract wss for 2-15 clusters
wss_values <- map_dbl(k_values, wss)
```

```
plot(k_values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")

km = kmeans(old, centers = 4, nstart = 25)
km[2]

"""# **Young**"""

young = df[df$Age_60_And_Above == 0,]

# Elbow method to find best k value
set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(young, k, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 1 to k = 15
k_values <- 1:15

# extract wss for 2-15 clusters
wss_values <- map_dbl(k_values, wss)

plot(k_values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")

km = kmeans(young, 4, nstart = 25)
km[2]
```

Logistic Regression

```
# Covid Dataset
suppressWarnings(RNGversion("3.5.3"))
set.seed(1)
myIndex <- createDataPartition(df$Result, p=0.6, list=FALSE)
trainSet <- df[myIndex,]
validationSet <- df[-myIndex,]
set.seed(1)
```

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

```
model1 <- glm(Result ~ Cough + Fever + Sore_Throat + Shortness_Of_Breath + Headache +
Age_60_And_Above + Contact + Male, data = trainSet, family = binomial)
summary(model1)

vip(model1)

prediction <- predict(model1, validationSet, type = "response")
prediction = ifelse(prediction > 0.1, 1, 0)
prediction = as.factor(prediction)
confusionMatrix(prediction, validationSet$Result)

roc_object <- roc(validationSet$Result, as.numeric(prediction))
plot.roc(roc_object)
auc(roc_object)

#male
suppressWarnings(RNGversion("3.5.3"))
set.seed(1)
myIndex <- createDataPartition(male_df$Result, p=0.6, list=FALSE)
trainSet <- male_df[myIndex,]
validationSet <- male_df[-myIndex,]
set.seed(1)

model2 <- glm(Result ~ Cough + Fever + Sore_Throat + Shortness_Of_Breath + Headache +
Age_60_And_Above + Contact, data = trainSet, family = "binomial")
summary(model2)

vip(model2)

#female
suppressWarnings(RNGversion("3.5.3"))
set.seed(1)
myIndex <- createDataPartition(female_df$Result, p=0.6, list=FALSE)
trainSet <- female_df[myIndex,]
validationSet <- female_df[-myIndex,]
set.seed(1)

model3 <- glm(Result ~ Cough + Fever + Sore_Throat + Shortness_Of_Breath + Headache +
Age_60_And_Above + Contact, data = trainSet, family = "binomial")
summary(model3)

vip(model3)

#elder
```

Team: Ehab Abdo, Raghav Arora, Rasa Blortchi, Vibeesh Kamalakannan, Sneha Sabu, Max Sohl, Harshal Suryawanshi

```
suppressWarnings(RNGversion("3.5.3"))
set.seed(1)
myIndex <- createDataPartition(elderly_df$Result, p=0.6, list=FALSE)
trainSet <- elderly_df[myIndex,]
validationSet <- elderly_df[-myIndex,]
set.seed(1)

model4 <- glm(Result ~ Cough + Fever + Sore_Throat + Shortness_Of_Breath + Headache +
Male + Contact, data = trainSet, family = "binomial")
summary(model4)

vip(model4)

#non_elder
suppressWarnings(RNGversion("3.5.3"))
set.seed(1)
myIndex <- createDataPartition(nonelderly_df$Result, p=0.6, list=FALSE)
trainSet <- nonelderly_df[myIndex,]
validationSet <- nonelderly_df[-myIndex,]
set.seed(1)

model5 <- glm(Result ~ Cough + Fever + Sore_Throat + Shortness_Of_Breath + Headache +
Male + Contact, data = trainSet, family = "binomial")
summary(model5)

vip(model5)
```