Coronavirus Mortality Prediction Using a Classification Prediction Model

Albina Cako, Colin Green, Lucy Zhang, Sean Zhang

Abstract

This project focuses on building a machine learning model on predicting probability of death during the Covid-19 pandemic using supervised learning. Data obtained by the Mexican government was used. The data was visualized, cleaned and prepared for modeling. Prediction of the model uses the patients sex, age and 12 medical preconditions. Four different classification models were created: decision tree, random forest, logistic regression and gradient boosting models. The models were evaluated on their accuracy, specificity, and sensitivity along with other metrics. The random forest model had the highest performance and accuracy, however, due to its large size, the logistic regression model was chosen for deployment. Deployment of the model was constructed using ShinyApp. An application was created that can be used on predicting probability of death based on age, sex and medical preconditions. The application can be used as a guide for medical professionals in decision making for resource distribution, as well as for patients who test positive with Covid-19 on whether to seek hospitalization. The app has it's limitations and it is advised to be used only as a guide, as it does not replace expert advice or government recommendations.

Background

Covid-19 is a viral infection caused by the novel virus called severe acute respiratory syndrome 2 (SARS-COV-2). Incubation period of Covid-19 is 2-14 days, which is the time a person shows symptoms after being exposed to the virus. The main symptoms are fever, cough, runny nose, headache, fatigue and muscle aches. The virus was first recorded in China in Fall 2019, however, around February 2020 it started rapidly spreading across the world. On March 11, 2020, the World Health Organization declared Covid-19 to be a global pandemic. One of the most tragic events of Covid-19 was the first wave of the virus in Italy, where around 44,000 deaths and 250,000 infections were recorded in just 3 months, from March to May 2020. Overwhelmed hospitals, shortage of medical supplies, hospital beds, icu units and medical staff, were a great challenge during the first wave of the virus. However, the fear of these factors is rising as the second wave of the pandemic is currently happening around the world. The virus has caused over 3 million deaths and has spread across 188 countries to date.

Governments all over the world have instructed people who tested positive for Covid-19 to quarantine for 14 days if they have mild symptoms or to visit the nearest hospital if they experience difficulty breathing, cyanosis or chest pain. People are considered at high risk of developing more serious symptoms or death if they have pre-existing medical conditions such as, but not limited to, chronic obstructive pulmonary disease (COPD), chronic kidney disease, obesity, smoking, immunocompromised, heart conditions, type two diabetes and cancer. Older age population is also at higher risk, as they are more likely to have a pre-existing medical condition and lower immunity.

As Covid-19 is still a threat to public health worldwide and the second wave of the virus is now growing globally, information is needed by medical professionals and people who have tested positive with Covid-19 when making decisions on whether hospitalization is needed. The use of machine learning is growing in the fight against Covid-19.

A death prediction app can assist medical professionals to make decisions on which patients are at high risk and require hospitalization. In addition, the app can also assist people infected with Covid-19 in deciding whether to self-quarantine or go to a hospital. We expect that our application, when used in conjunction with expert medical advice, will help guide resource allocation during the second wave.

Objective

The objective of this project is to create a supervised machine learning model that predicts probability of death due to Covid-19 based on age and pre-existing medical conditions. This app could be used by medical professionals to guide resource allocation for the treatment of Covid-19 patients. Having an app that predicts death probability can help hospitals decide which patients need hospitalization. In addition, the app can also be used by people who test positive with covid-19 as a guide to decide whether they should visit the hospital. Having this app can reduce unnecessary surges to the hospital of people who are at low risk. The app is simple to use and easily accessible by anyone.
Disclaimer: This application should only be used as a guide and is not a replacement for expert advice or government recommendations.

Data Analysis

The dataset was made publicly available from the Mexican government. It has over 560,000 anonymous records of patients who attended a hospital from March – July 2020 regarding Covid-19. Over 220,000 records were confirmed positive for Covid-19.

Data Dictionary

The table below defines each column in the dataset.

Label	Description
id	Case identifying number
sex	Sex of patient, where 1 means female, 2 means male, 99 means non-specified
patient_type	Describes the type of care the patient received, where 1 means discharged the same day, 2 means hospitalized, 99 means non-specified
entry_date	Hospital entry date
date_symptoms	The first date the patient started experiencing symptoms
date_died	The date the patient died if date is shown or 9999.99 if patient did not die or it is not-specified whether they died or not
intubed	Whether a patient was intubed in the hospital, where 1 means yes, 2 means no, 97 means not applicable. Our analysis showed that 97 was used for patients who were discharged the same day.
pneumonia	Whether the patient had pneumonia as a preexisting condition, where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
age	Patient's age
pregnancy	Whether a female patient was pregnant or not. Our analysis showed that 97 that means not applicable was used for male patients
diabetes	If the patient had diabetes as a preexisting condition, where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified)
copd	Whether the patient had copd - chronic obstructive pulmonary disease as a preexisting condition , where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
asthma	Whether the patient had asthma as a preexisting condition , where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
inmsupr	Whether the patient was immunocompromised, where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
hypertension	Whether the patient had hypertension as a preexisting condition , where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
other_disease	Whether the patient had another disease not listed as a preexisting condition, where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified

Label	Description
cardiovascular	Whether the patient had a cardiovascular disease as a preexisting condition, where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
obesity	Whether the patient was obese , where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
renal_chronic	Whether the patient had chronic renal disease as a preexisting condition , where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
tobacco	Whether the patient smoked tabacco, where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified
contact_other	Whether the patient had contact with apositive case prior to contacting the virus , where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified.
covid_res	Whether the patient tested positive or not , where 1 means yes, 2 means no, 97 means not applicable, 98 means ignored/not specified, 99 means not specified.
icu	Whether a patient was intubed in the hospital, where 1 means yes, 2 means no, 97 means not applicable. 97 was used for patients who were discharged the same day.

Data Exploration

The first step in our data exploration is to load in the data and then remove any cases that were not covid positive as they will have no affect on our analysis. We will then look at a summary of the data.

```
covid <- read_csv("covid.csv")
covid <- covid %>% filter(covid_res == 1)
```

	covid (N = 220,657)
sex	
male	120799
female	99858
patient_type	
hopsitalized	152361
discharged same day	68296
intubed	
yes (1)	6549
no (2)	61662
not applicable (97)	152361
ignored (98)	0
not specified (99)	85
pneumonia	
yes (1)	53031
no (2)	167623
not specified (99)	3
age	
mean	45.69
median	45
standard deviation	16.31
pregnancy	
yes (1)	1425
no (2)	97749
not applicable (97)	120799

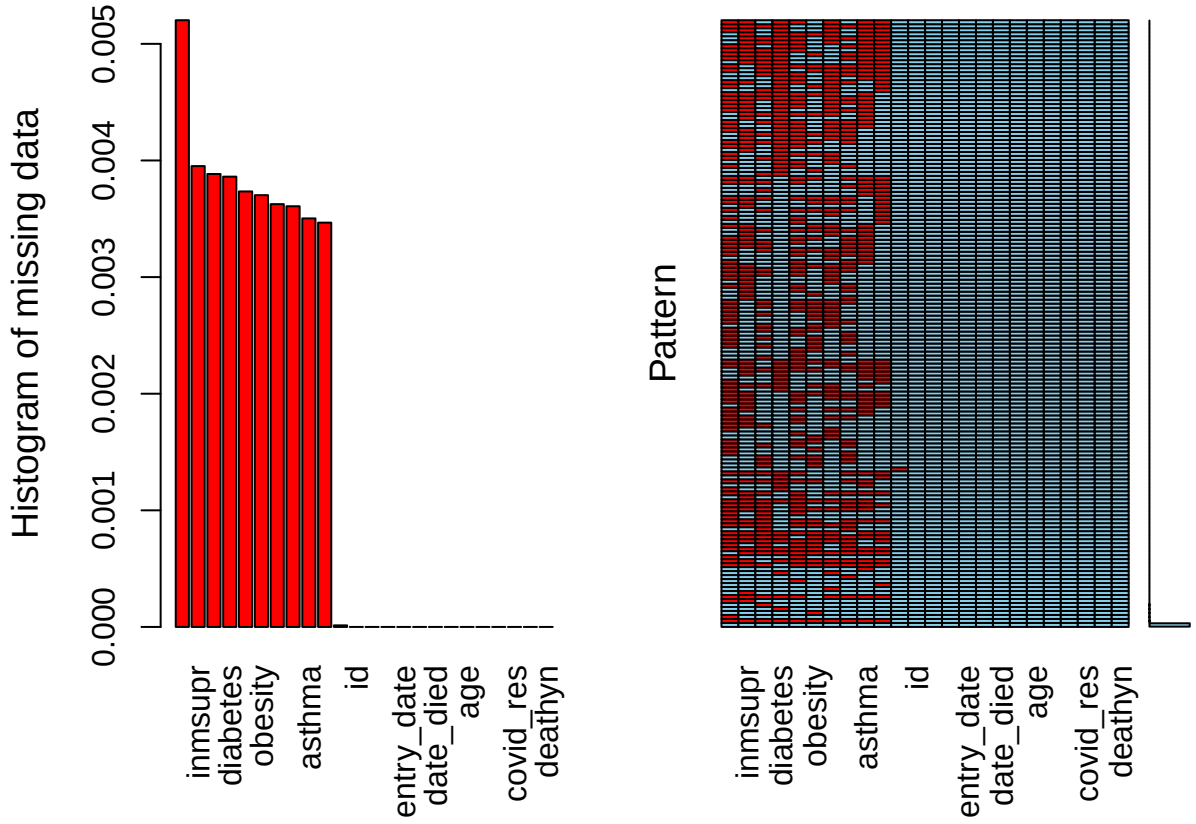
	covid (N = 220,657)
ignored (98)	684
diabetes	
yes (1)	36187
no (2)	183618
ignored (98)	852
copd	
yes (1)	3877
no (2)	216015
ignored (98)	765
asthma	
yes (1)	6063
no (2)	213821
ignored (98)	773
inmsupr	
yes (1)	3016
no (2)	216769
ignored (98)	872
hypertension	
yes (1)	44297
no (2)	175560
ignored (98)	800
other_disease	
yes (1)	6283
no (2)	213226
ignored (98)	1148
cardiovascular	
yes (1)	5162
no (2)	214671
ignored (98)	824
obesity	
yes (1)	43241
no (2)	176599
ignored (98)	817
renal_chronic	
yes (1)	4789
no (2)	215072
ignored (98)	796
tobacco	
yes (1)	17109
no (2)	202691
ignored (98)	857
contact_other_covid	
yes (1)	74280
no (2)	65352
not specified (99)	81025
icu	
yes (1)	5822
no (2)	62388
not applicable (97)	152361
not specified (99)	86

In order to explore the data further we will need to see how death is related to certain aspects of the data.

```r
covid<- covid %>%
  mutate(deathyn = ifelse(date_died == '9999-99-99', '0','1'))
```

Missing Data

First, we visualized the missing data.




```
##
##  Variables sorted by number of missings:
##             Variable        Count
##       other_disease 5.202645e-03
##              inmsupr 3.951835e-03
##              tobacco 3.883856e-03
##             diabetes 3.861196e-03
##       cardiovascular 3.734303e-03
##              obesity 3.702579e-03
##         hypertension 3.625536e-03
##        renal_chronic 3.607409e-03
##               asthma 3.503175e-03
##                 copd 3.466919e-03
##            pneumonia 1.359576e-05
##                   id 0.000000e+00
##                  sex 0.000000e+00
##         patient_type 0.000000e+00
##           entry_date 0.000000e+00
##        date_symptoms 0.000000e+00
```

```
##            date_died 0.000000e+00
##              intubed 0.000000e+00
##                  age 0.000000e+00
##            pregnancy 0.000000e+00
##  contact_other_covid 0.000000e+00
##            covid_res 0.000000e+00
##                  icu 0.000000e+00
##              deathyn 0.000000e+00
```

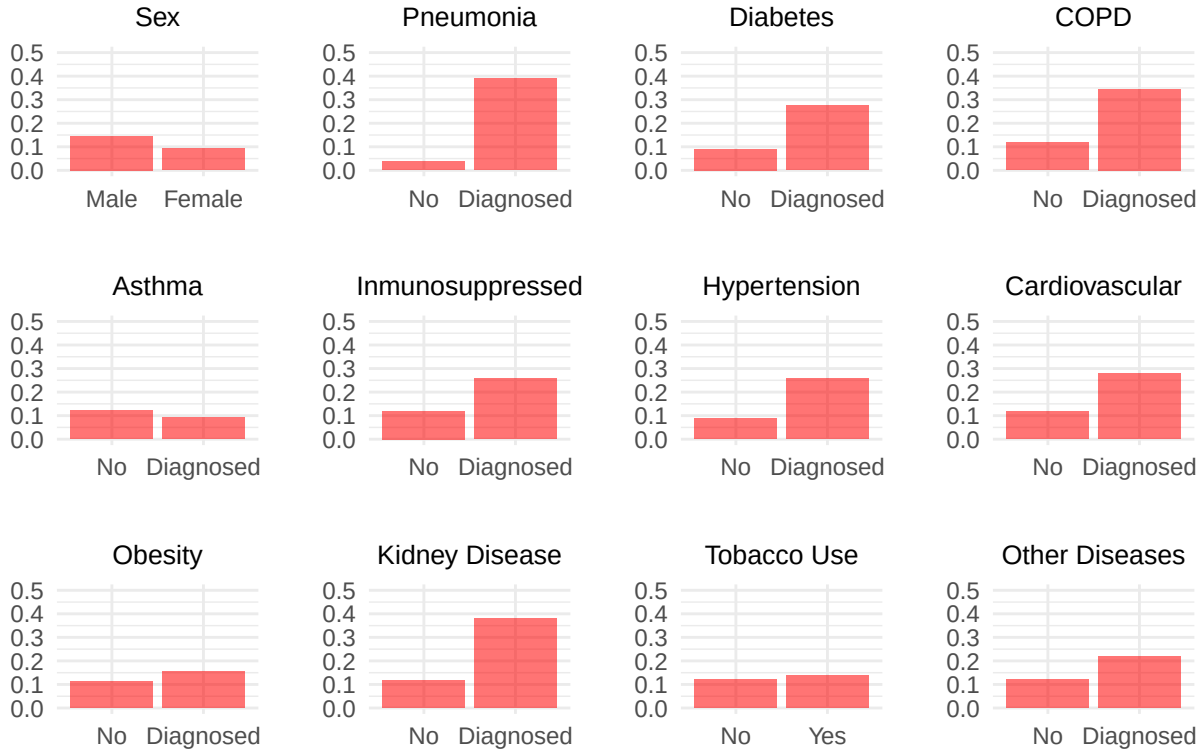
Missing data in the columns were characterized by 97 = not-applicable, 98 = ignored or not specified or 99 = not specified. In the precondition columns, missing data was proportionately low as seen in the visualization above.
We analyzed the data and realized that 97 was used in 3 columns: pregnancy, intubed or icu. Considering that pregnancy only applies to females, we checked whether 97 indicated that the person was male. Our analysis showed that 97 meant male patient. It was changed to zero in our table. We also explored 97 value in intubed and icu columns. We realized that it would be inapplicable to say a patient was intubed or on icu if they were discharged the same day. We ran an analysis and noticed that 97 was only used for patients who were not hospitalized in both cases. Since patient's were not hospitalized, that means that they did not need to be intubed or placed in icu, therefore 97 means "no" in these two columns. We substituted the 97 values with 0's (no) in our tables. The rows that contained the values 98 and 99, which mean ignored or not specified were removed from our data. These values were ignored in the icu and intubed columns which were not included in our analysis, as they are not a precondition. We removed any observation where a precondition column had a missing value (value = 98 or 99), under the assumption that data were Missing Completely At Random (MCAR). This still left us with a dataset of 218275 cases.
To clean the data, we converted all 2's that represent "no"s into zero's, since we are using a binary system for all preconditions. This excludes age as it is not a binary categorical variable.

Data Subsets

	cases	death%	mean_age	median_age	mean_age_death	male%
covid	220657	12.29	45.69	45	61.00	54.75
covid_complete_case	218275	12.22	45.66	45	61.03	54.83
covid_incomplete_cases	2382	18.39	48.52	48	58.98	47.02
covid_ignored_cases	2379	18.41	48.54	48	58.98	46.99
covid_blank_cases	536	17.91	49.17	47	60.26	64.93

In order to assess the extent missing variables we created subsets of the data. The first set is the entire covid positive dataset. Covid_complete_case contains all of the cases that had a response of yes or no in the precondition columns. Covid_incomplete_cases contains all of the cases that had at least one response of 97, 98, or 99 in the precondition columns. Covid_ignored_cases contains all of the cases that had a response of 98 in one of the precondition columns. Finally, covid_blank_cases contains all of the cases that had a response of 98 in all of the precondition columns.
The table clearly shows that the complete cases make up the majority of the cases. The cases with missing data were more likely to die and were often older patients.

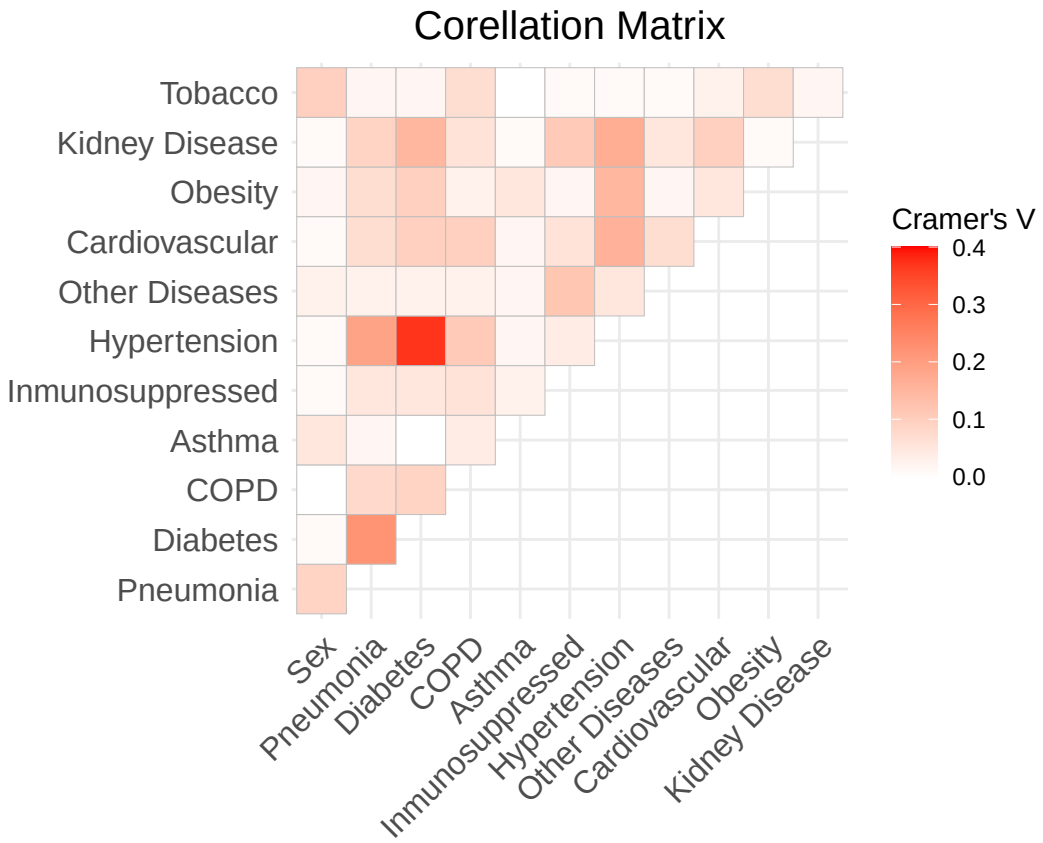
We plotted individual graphs to predict the relationship between our independent variables and our target variable death.



The graphs above show the comparison between our predictor variables and death. The proportion of deaths is represented by the y-axis. For example, in the "Sex" variable graph, we observe that the male sex has a greater proportion of individuals who died from Covid-19 compared to females. The highest proportion of death is seen in patients with pneumonia, kidney disease, and COPD.

Correlation of variables

To perform the correlation between our independent variables, we used the Chi-squared test, as a standard correlogram using Pearson's R does not accept categorical variables. However, we were unable to use the original p-value provided by the test as samples higher than a few hundred typically always return $p < 0.05$. To adjust for our large sample size, we used Cramér's V test instead. A higher Cramér's V value denotes stronger correlation.



The above correlogram shows correlations between the categorical predictor variables. A value greater than 0.5 typically represents collinearity; as all values fall below that threshold, we can assume independence.

Takeaways from Data Exploration

During data exploration we learned that there was a relatively low proportion of missing datapoints within our predictor variables. Additionally, our predictors are independent of each other, and that there appeared to be a visual correlation between our predictors and the target variable.

Data Imputation

Data exploration showed that there were patients' of age 0 up to 120. As 120 years old could be an error in imputation or an outlier, we decided to impute the upper age range to reduce error. Any age above 95 years old was replaced with 95. Age of 0 was left as it is, as there could be patients newborn or under the age of 1 years old.

```
mean_age <- mean(covid_complete_case$age)
sd_age <- sd(covid_complete_case$age)
imputed_age <- round(mean_age + 3*sd_age)
covid_complete_case$age[covid_complete_case$age > imputed_age] = imputed_age
```

Modeling and Evaluation

We have understood, cleaned and organized our data. We are now ready to create our prediction model. We have decided to create 4 models, evaluate them and then chose the model with the highest accuracy without making sacrifices to sensitivity or specificity.

Feature Selection

The data started with 23 columns. We have added the death column, which was obtained by the date_died column, as explained above. The death column will be used as our target variable. We excluded the icu, intubed and contact_other columns in our models as they are not medical preconditions. We chose 14 features in total to be used in our analysis, 12 which are the preconditions and 2 which are the patient's sex and age. We decided to keep sex as a factor in our analysis, as data exploration showed that difference in sex was significant in covid death rates. The 12 precondition features chosen were: pneumonia, pregnancy, diabetes, copd, asthma, inmsupr (immunosupressed), hypertension, other_disease, cardiovascular, obesity, renal_chronic and tobacco. Age was the other feature chosen and it was used as a separate numerical feature.

Cross Validation

The data was split into k = 10 folds. During each training iteration, one of the k subsets was used as the test and the other k-1 subsets were used as the training set. This method was used instead of the simple train-test-split as it gives a more valid estimation of model effectiveness.

Logistic Regression Model

Logistic regression is a common model used in binary classification problems. One of the benefits of logistic regression is its interpretability. The model produces an easy to read equation that shows the variable importance. For example, the coeficient for pneumonia is 2.25, this means that someone with pneumonia has 2.25% higher chance of dying than someone who doesn't have pneumonia (holding all other factors constant). Logistic regression models are often used to classify binary dependent variables. In our case we did not want to classify someone as dead or alive so we will only be interpretting the value provided by the equation as a specific case risk. The accuracy for the logisitc regression model was 88.66%. While it was very good at predicting if someone would live, it struggled to predict death. Running the model through an upsampling method did not improve the results.

```
##     (Intercept)            sex1      pneumonia1             age      pregnancy1
##     -5.80930786     -0.45302764      2.25528335      0.05246929      0.07019434
##     pregnancy97       diabetes1           copd1         asthma1        inmsupr1
##              NA      0.32907552      0.15899211     -0.12368372      0.30847526
##   hypertension1  other_disease1 cardiovascular1        obesity1  renal_chronic1
##      0.18531198      0.48283785     -0.09276075      0.24359063      0.74618758
##        tobacco1
##     -0.07552174
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    0    1
##          0 84.7  8.2
##          1  3.1  4.0
##
##  Accuracy (average) : 0.8866
```

Decision Tree Model

The decision tree model was chosen as one of the classification models as it is fast, accurate and can handle a large amount of data. It is also easily interpreted visually and does not take a lot of data preparation. The decision tree model yielded a 89.03 % total accuracy. The model had a specificity of 91.08 % and a sensitivity of 58.77 %. The model very is very accurate at predicting non-death, but it's not accurate at predicting death. 'This is more likely due to the data being unbalanced, as most of the data had non-death.

For this reason, the model was not chosen to make the application for this project.' Running the model through an upsampling method did not improve the results.

```
##     pneumonia1            age  hypertension1 renal_chronic1      diabetes1
##    9828.048878    2253.679953     231.048502      74.896075      58.411769
##     pregnancy1       obesity1    pregnancy97           sex1        asthma1
##      17.397825      10.096798       8.878814       8.878814       5.530805
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    0    1
##          0 85.0  8.5
##          1  2.8  3.7
##
##  Accuracy (average) : 0.8868
```

Random Forest Model

Random Forest is an 'ensemble' model that fits based on majority voting from numerous decision trees which corrects for overfitting. As Random Forest is a non-parametric algorithm, it requires little data preparation beforehand. Variables can be ranked according to importance based on Gini index, though how a variable affects final output is less interpretable than logistic regression. The first iteration of training the Random Forest Classifier yielded 88.3% overall Accuracy, but a low Sensitivity of 27.96%, meaning it could correctly predict fewer than one-third of the cases where death occurred. We hypothesized that the low sensitivity might be due to an imbalance within the dataset (deaths occurred in <5% of all cases), and that undersampling the majority class (no death) or oversampling the minority class (death) might increase accuracy. Using the caret package, the Random Forest model was trained additionally using the following sampling techniques: down (simple random undersampling an equivalent number of cases where patients did not die), ROSE (Random Over-Sampling Examples), and SMOTE (Synthetic Minority Oversampling Technique). The results are shown below:

model	Accuracy	Specificity	Sensitivity	Balanced.Accuracy
original	88.29%	96.70%	27.96%	62.44%
down	81.10%	79.16%	83.05%	81.11%
ROSE	91.39%	90.33%	92.45%	91.39%
SMOTE	84.06%	84.99%	82.82%	83.90%

The different sampling methods to balance the data all resulted in an increase in sensitivity as hoped, though overall accuracy fell for down-sampling and SMOTE due to a decrease in detecting true negative cases. However, the ROSE method was chosen as the final model, as it greatly increased sensitivity while preserving specificity, thus leading to a higher overall accuracy and much higher balanced accuracy.

```
##                MeanDecreaseGini
## sex                    2050.195
## pneumonia             35298.290
## age                   13258.784
## pregnancy             10605.882
## diabetes               5081.880
## copd                   8700.317
```

```
## asthma                 1837.096
## inmsupr                4693.982
## hypertension           3125.686
## other_disease          3389.804
## cardiovascular         5289.382
## obesity                2015.949
## renal_chronic         11908.067
## tobacco                1881.653
```

```
##       0      1
## 0 98609  10553
## 1  8237 100876
```

Gradient Boosting

Gradient boosting is a method of converting weak learners into strong learners. The model begins by training a decision tree with equal weight, and then increasing the weights of those observations that are difficult to classify and then lowering the weights for easier ones to create a new tree. Our model is therefore a combination of tree 1 and tree 2. Gradient boosting identifies the shortcomings by using gradients in the loss function (y=ax+b+e, e needs a special mention as it is the error term). The loss function is a measure indication of how good the model's coefficients are at fitting the underlying data. We used the package "gbm" to train the gbm model and used k-fold cross-validation method to determine the best iteration. There are several methods we can choose for distribution of our response variable, for example-"bernoulli" (logistic regression for 0-1 outcome), "gaussian"(squared errors), "tdist" (t-distribution loss), we used bernoulli for our gbm model. The overall accuracy for this model is 88.67%, however, like the logistic regression and decision tree models, the gradient boosting model struggled to predict death, resulting in a low sensitivity.

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##            Reference
## Prediction    0    1
##          0 84.8  8.3
##          1  3.0  3.9
##
##  Accuracy (average) : 0.8867
```

Model Selection

As you can observe from the table, the random forest model has the best performance, as its accuracy and F1 score are 91.4% and 91.5% respectively. The other three models performed well, but all have some limitations, as described above. Unfortunately we are unable to use the random forest model for deployment due to the size of the model file. We chose the logistic regression model for deployment as it was still fairly accurate, had the 2nd best F1 score, and the equation made deployment simpler.

model	accuracy	sensitivity	specificity	precision	kappa	bal_accuracy	f1
logistic_regression	0.887	0.563	0.912	0.328	0.355	0.738	0.415
decision_tree	0.887	0.569	0.909	0.303	0.340	0.739	0.396
random_forest	0.914	0.925	0.903	0.905	0.119	0.914	0.915
gradient_boosting	0.887	0.565	0.911	0.320	0.351	0.738	0.408

Deployment

We created an application that used the logistic regression model with R Shiny. The user can choose from a checklist of pre-set parameters based on the model's predictor variables (e.g. age, sex, various preconditions) in the UI, in which the server will then output a probability of death. The application then compares the predicted probability with the distribution of all cases predicted by the model and returns a risk score (low, middle, high) based on how the predicted probability compared to the entire dataset.

While the Random Forest model with ROSE sampling was the most accurate, it was too large to be deployed with the Shinyapps.io cloud. Thus, the logistic regression equation, with its high interpretability and low memory usage, was chosen for the application back-end instead. The application can be found here https://sean-z.shinyapps.io/covid_analysis/.

Conclusion

In this project we used supervised machine learning to create an application that predicts probability of death from Covid-19 based on age, sex, and pre-existing medical conditions. The dataset was obtained from Kaggle and prepared by the Mexican government. The data was cleaned and a total of 13 features were included in the models. Eleven features were medical preconditions as binary data. Four different machine learning models were created: decision tree model, logistic regression model, random forest model and gradient boosting model. They were evaluated based on accuracy from which the random forest model was the one with the highest accuracy. However, due to the large size of the random forest model, the logistic regression model was selected for building our application. Our application is displayed on shiny app. It can be used by medical professionals as a guidance in decision making for the hospitalization of patients and use of medical resources during the Covid-19 pandemic. It can also be used by people who have tested positive with Covid-19 as a guide on whether to seek hospitalization. The application is not to replace expert advice or governmental recommendations.

There are limitations to be considered for this project. First of all, the dataset was obtained from the Mexican government and it only contains Mexican population. This means that the data does not represent the whole world population diversity. Furthermore, the data is from the first wave of the virus when less was known about the virus. Viruses are known to change or mutate, so the data might not reflect all types of Covid-19 strains. A few different strains are known of Covid-19, thus this data might not reflect all of them, especially the ones that have been from a recent mutation. In addition, the preconditions severity is unknown. Depending on how severe a condition is, it can affect death probability. This application is build on the assumptions that all preconditions are in equal severity. Also, the Covid-19 test is not 100% accurate. Our model does not account for the accuracy of the Covid-19 test, which could affect the prediction results. We also acknowledge the fact that some of the cases listed may have died after the data was collected, though we account for censoring by Kaplan-Meier survival analysis (see Supplementary). Finally, we do not know whether it is a Covid-19 related death. As many patients had other preconditions, we cannot know for certain that each of them died from Covid-19.

Overall, we were able to build an accurate model that predicts the probability of death based on preconditions for confirmed-positive Covid-19 patients. However, our model does have limitations, as it was built mainly to demonstrate a proof-of-concept. Further tuning would be necessary to improve model accuracy and stakeholder decision-making.

Bibliography

Certain Medical Conditions and Risk for Severe COVID-19 Illness. (n.d.). Retrieved October 13, 2020, from https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-with-medical-conditions.html/ Coronavirus Cases:. (n.d.). Retrieved October 13, 2020, from https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1%3F Li, H., Liu, S. M., Yu, X. H., Tang, S. L., & Tang, C. K. (2020). Coronavirus disease 2019 (COVID-19): current status and future perspectives. International journal of antimicrobial agents, 55(5), 105951. https://doi.org/10.1016/j.ijantimicag.2020.105951/ Meng, L., Qiu, H., Wan, L., Ai, Y., Xue, Z., Guo, Q., Deshpande, R., Zhang, L., Meng, J., Tong, C., Liu, H., & Xiong, L. (2020). Intubation and Ventilation amid the COVID-19 Outbreak: Wuhan's Experience. Anesthesiology, 132(6), 1317–1332. https://doi.org/10.1097/ALN.0000000000003296/

Supplementary

To account for censoring of deaths due to loss of follow-up, we calculated survival times as difference between date of symptom onset and date of death. We considered July 1, 2020 as the last date of follow-up and imputed it in instances where date of death was missing. Our analysis found that the median follow-up time was 30 days, while the mean follow-up time was 34.9 days. The median and mean survival times were fond to be 10 and 11.5 days, respectively with an interquartile range of 9 days and standard deviation of 7.4 days. Thus, we are likely to have captured the majority of deaths within our analysis. The following figure shows predicted Kaplan-Meier survival curves, segmented by our predictor values. We did not predict survival duration using machine-learning based regression algorithms, as it was beyond the scope of this project.

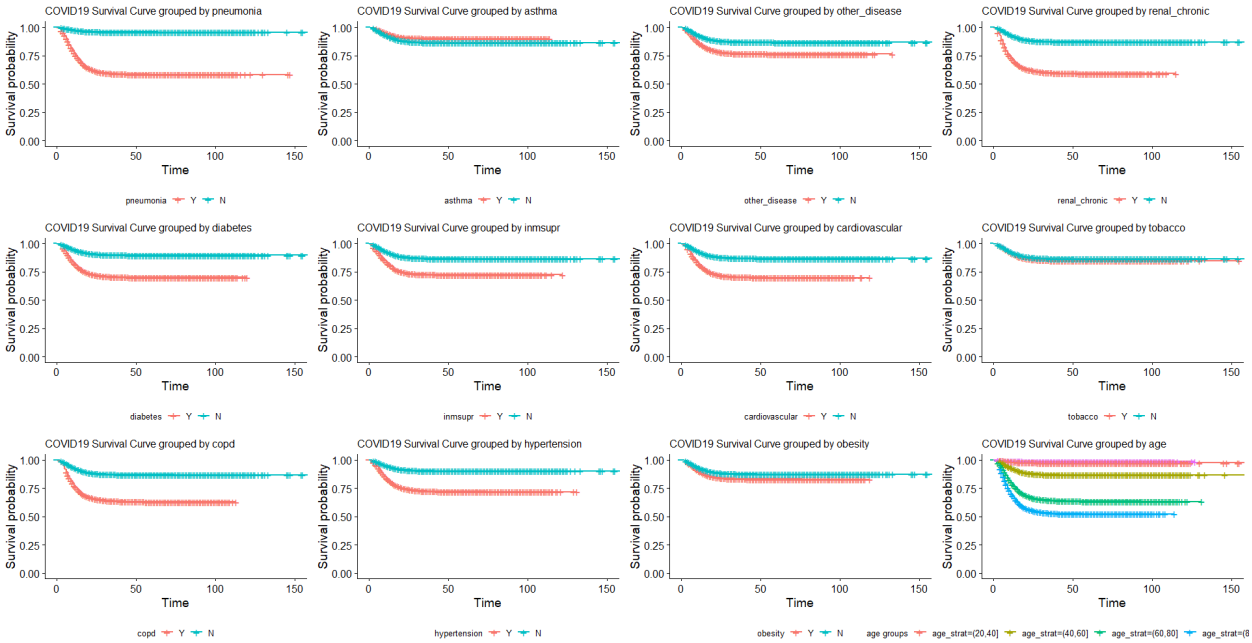


Figure 1: Supplementary Figure of Kaplan-Meier Survival Curves