PREDICTING THE SURVIVAL RATE OF COVID-19 PATIENTS

Antone Evans Jr., Abilash Vanam

May 3rd, 2020

ABSTRACT

Coronavirus disease 2019 also known as **COVID-19** is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [1]. The disease was first identified in December 2019 in Wuhan, the capital of China's Hubei province, and has since spread globally, resulting in the 2019–20 coronavirus pandemic. The first confirmed case of what was then an unknown coronavirus was traced back to November 2019 in Hubei. It plays an increasingly significant role to predict the survival rate efficiently and precisely for critically ill COVID-19 patients as more fatal cases can be targeted and interfered in advanced. Therefore, as the death rates increase, and resources begin to shorten, patients with a higher chance of mortality based on age, gender, medical and travel history should be at the top of the list to receive aggressive treatment. As a remedy, we have developed multiple machine learning models that precisely predicts the survival of a COVID-19 patient.

In this project, we used the Novel Corona Virus 2019 dataset with 1,085 observations and 21 variables to predict the survival rate of patients. Pre-processing of the dataset included removing the variables which are negligible to predict the outcome variable and imputation using missForest method. To predict the outcome variable, this project applies Regularized Logistic Regression such as Lasso, Ridge and Elastic Net methods; Tree Ensemble Models such as Random Forest Model, Gradient Boosting Tree Model, along with K-Nearest Neighbor, and Artificial Neural Networks. We also use Anomaly Detection methods such as One-Class Support Vector Machines and Isolation Forest because our data was severely imbalance.

PROBLEM DEFINITION AND PROJECT GOALS

The main goal of this project is to use various Machine Learning Models to predict the death of a COVID-19 patient given the input data.

The dataset was obtained from Kaggle.com. The link to the dataset: https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset#COVID19_line_list_data.csv

This dataset initially holds 1,085 observations of 21 variables each with its unique id as seen in figure 1. This project uses the death variable as its outcome variable.

NO.	VARIABLE	TYPE	DESCRIPTION
1	ï..id	Integer	Unique id for each positive patient
2	case_in_country	Integer	Number of cases in the patient's country
3	reporting.date	Factor	Date on which the patient first reported
4	X	Logical	This column has all NA's
5	Summary	Factor	Summary of the patient
6	Location	Factor	City of the patient

7	Country	Factor	Country of the patient
8	Gender	Factor	Gender of the patient
9	Age	Numerical	Age of the patient
10	symptom_onset	Factor	Date on which patient first showed symptoms
11	If_onset_approximated	Integer	If the patient contacted with someone positive
12	hosp_visit_date	Factor	Date on which the patient first visited hospital
13	exposure_start	Factor	Start date of exposure of the patient
14	exposure_end	Factor	End date of exposure of the patient
15	visiting.Wuhan	Integer	If the patient visited Wuhan or not
16	from.Wuhan	Integer	If the patient is from Wuhan
17	Death	Factor	Death date of the patient/ if the patient is dead
18	Recovered	Factor	Whether the patient recovered/ recovery date
19	Symptom	Factor	Symptoms seen in patient
20	Source	Factor	Source of all the data
21	Link	Factor	Web link of the information

Table 1. List of all variables

The dataset had a lot of NA's to deal with therefore, instead of removing all the NA's this project uses the missForest model to replace the NA's with the average of the column's observations. A lot of pre-processing is done to this dataset. There are a few variables which are negligible to predict the death variable and have been removed. After all the pre-processing is done this project trains the pre-processed dataset on different machine learning models and compares each accuracy, AUC, kappa and FNR to find the best model.

RELATED WORK

With this pandemic being an ongoing process along with the lack of accurate datasets there are not much related works to our topic. However, we were able to find two interesting papers. It is good to know that currently none of these papers have been peer reviewed. The first is entitled, "Prediction of survival for severe Covid-19 patients with three clinical features: development of a machine learning-based prognostic model with clinical data in Wuhan". In this study, they collected 3,129 patients' electronic records who were confirmed or suspected of COVID-19 from January 10th to February 18th, 2020 at Tongji Hospital in Wuhan, China. The method used on this dataset is the XGBoost which reported back an accuracy of 93% on their training dataset [2].

The other paper is entitled, "Predicting Mortality Risk in Patients with COVID-19 Using Artificial Intelligence to Help Medical Decision-Making". This paper uses a dataset of more than 117,000 laboratory-confirmed COVID-19 patients from 76 countries around the world. This paper used more similar algorithms to our paper such as Support Vector Machine (SVM), Neural Networks, Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbor. They received an accuracy over 10-fold cross validation as 90.36%, 93.75%, 91.88%, 90.63%, 90.00% and 83.12% respectively [3].

Our paper is more similar to [3] as we also used 10-fold cross validation on our algorithms, and we used similar algorithms. However, unlike this paper we use more algorithms and we report the AUC, kappa and FNR of these algorithms. Similar to [2] we had to do a lot of preprocessing with our data however, we used different algorithms to train, test and validate our data.

DATA EXPLORATION AND PRE-PROCESSING

Exploring Data

We started by getting a summary of our data as seen in table 1. This was important because when searching for our dataset we noticed that there was no complete dataset. However, there were many with missing rows and unnecessary columns. Therefore, followed by summarizing our data we got the NAs from each variable as seen in table 2. This was important in deciding which variables could be cleaned up and others to removed quickly. Along with considering the amount of NAs in the variable we decided to remove variables which would not help us when training our dataset because of the lack of information gained such as case_in_country, summary, X, location, exposure_start, exposure_end, link, and source.

No.	VARIABLE	PERCENTAGE OF NA's (%)
1.	ï..id	0.000
2.	case_in_country	18.15
3.	reporting.date	0.092
4.	X	100.0
5.	Summary	0.460
6.	Location	0.000
7.	Country	0.000
8.	Gender	16.86
9.	Age	22.30
10.	symptom_onset	48.11
11.	If_onset_approximated	48.38
12.	hosp_visit_date	53.17
13.	exposure_start	88.20
14.	exposure_end	68.57
15.	visiting.Wuhan	0.000
16.	from.Wuhan	0.368
17.	Death	0.000
18.	Recovered	0.000
19.	Symptom	0.000
20.	Source	0.000
21	Link	0.0

Table 2. Percentage of all NA's

Pre-Processing Data

We then converted the symptom_onset, hosp_visit_date and reporting.date to date columns. We did this so we could compute the time from an individual first presented a symptom to the reporting date along with computing the time they visited a hospital and their reporting date.

Symptom Length = Reporting Date – Symptom Onset Date

Reporting Length = Reporting Date – Hospital Visit Date

We believed this information would be useful as it could be used in the debate for mass testing as much countries around the world are currently not conducting it. This information could also allow individuals to understand the importance of reporting these symptoms quickly. The symptom_onset and hosp_visit_date variables were then removed from the dataset.

We then converted the if_onset_approximated, death and recovered variables to binary. In the if_onset_approximated is 0, it indicates the individuals did not encounter a COVID positive individual and 1 indicates the reverse. In the death variable 0 indicates survival and 1 indicates death. In the recovered variable 0 indicates not recovered and 1 indicates recovered.

Handling Missing Values

Following this step, we performed missForest which inputs values into all our NAs. MissForest does not work with variables with more than 53 categorical levels, therefore; we removed the symptom column along with the outcome variable which is death. We used missForest to find missing observations for reporting.date with 1 NA, gender with 183 NAs, age with 242 NAs, from.Wuhan with 4 NAs, symptomLength with 522 NAs, and reportingLength with 579 NAs. The out of bag error on this data is reported in table 3.

NRMSE	PFC
0.02708344	0.20288248

Table 3. OOB Error

Next, we added the symptom and death column back to the data. We then separated the symptom variable into 32 unique columns based on the number of unique symptoms in the dataset as seen in table 4. The columns' produced were binary where 0 indicates the patient did not have the symptom and 1 indicates the reverse. We then removed the symptom variable along with the ID variable as no meaningful data would be gained from this variable. Table 4 shows the final list of the 42 variables we used when training, testing and validating data.

No.	VARIABLES AFTER PRE-PROCESSING
1	Country
2	Gender
3	If_onset_approximated
4	visiting.Wuhan
5	from.Wuhan
6	Recovered
7	Symptom_none
8	Symptom_fever..mild.to.sever.
9	Symptom_coughing..mild.to.sever.
10	Symptom_difficulty.breathing
11	Symptom_chills
12	Symptom_joint.pain
13	Symptom_throat.pain
14	Symptom_nasal.discharge
15	Symptom_fatigue

16	Symptom_abdominal.pain
17	Symptom_diarrhea
18	Symptom_cold
19	Symptom_pneumonia
20	Symptom_vomiting
21	Symptom_loss.of.appetite
22	Symptom_malaise
23	Symptom_headache
24	Symptom_sputum
25	Symptom_myalgia
26	Symptom_sore.throat
27	Symptom_dyspnea
28	Symptom_nausea
29	Symptom_respiratory.distress
30	Symptom_throat.discomfort
31	Symptom_sneeze
32	Symptom_chest.discomfort
33	Symptom_thirst
34	Symptom_flu.symptoms
35	Symptom_muscle.cramps
36	Symptom_reflux
37	Symptom_physical.discomfort
38	Symptom_itchy.throat
39	Age
40	SymptomLength
41	ReportingLength
42	Death

Table 4. Variables after preprocessing

Correlation between the outcome and categorical variables

After completing the tedious preprocessing step, we made another dataframe to handle our data exploration. We did this so that we would not ruin our main dataframe. Working with binary variables proved extremely hard when trying to plot them on scatterplots or to get information from a side-by-side boxplot. Therefore, we began by converting our binary variables – death, if_onset_approximated, visiting.Wuhan, from.Wuhan, recovered and all of the symptom columns to yes and no columns where 1 is equivalent to Yes and 0 as No. Our outcome variable death was renamed survivalOrnot in this section. We started comparing our outcome variable with our categorical variables. As seen in table 5 we get a list of p values which were produced from doing a crosstable and chi square test on each variable. With a significance value of 0.01 which means there is a relationship between the outcome and the other variable. At this point we would remove the non-associated variables, however, due to our dataset being severely bias as seen in figure 12 we will not remove any of the variables as we will balance the training data later on.

Bar graphs in figures 1-8 represents a clear picture of how different variables are correlated with the death variable. Below are a few bar charts which are correlated with the death variable. Those variables selected display a p-value less than the significant value when performed a chi-squared test. The y-axis shows the number of positive tested patients and the x-axis represents the variable.

No.	VARIABLE ~ DEATH	P-VALUE
1	Country	5.841393e-10
2	Gender	0.0007997044
3	If_onset_approximated	0.5924556
4	visiting.Wuhan	0.0005564678
5	from.Wuhan	2.756613e-20
6	Recovered	0.0007018556
7	Symptom_none	0.0449717
8	Symptom_fever..mild.to.sever.	0.008410168
9	Symptom_coughing..mild.to.sever.	0.2504773
10	Symptom_difficulty.breathing	0.1737565
11	Symptom_chills	0.349769
12	Symptom_joint.pain	0.454488
13	Symptom_throat.pain	0.7252526
14	Symptom_nasal.discharge	0.3871106
15	Symptom_fatigue	0.7066357
16	Symptom_abdominal.pain	0.8038278
17	Symptom_diarrhea	0.4302411
18	Symptom_cold	0.00747502
19	Symptom_pneumonia	6.371671e-06
20	Symptom_vomiting	0.5419597
21	Symptom_loss.of.appetite	0.7252526
22	Symptom_malaise	0.1831657
23	Symptom_headache	0.3170533
24	Symptom_sputum	0.407861
25	Symptom_myalgia	0.6396606
26	Symptom_sore.throat	0.6084014
27	Symptom_dyspnea	0.8859324
28	Symptom_nausea	0.7252526
29	Symptom_respiratory.distress	0.8038278
30	Symptom_throat.discomfort	0.6188496
31	Symptom_sneeze	0.8038278
32	Symptom_chest.discomfort	0.6667357
33	Symptom_thirst	0.8038278
34	Symptom_flu.symptoms	0.7252526
35	Symptom_muscle.cramps	0.8038278
36	Symptom_reflux	0.8038278
37	Symptom_physical.discomfort	0.8038278
38	Symptom_itchy.throat	0.8038278

Table 5. P-value after chi-squared test

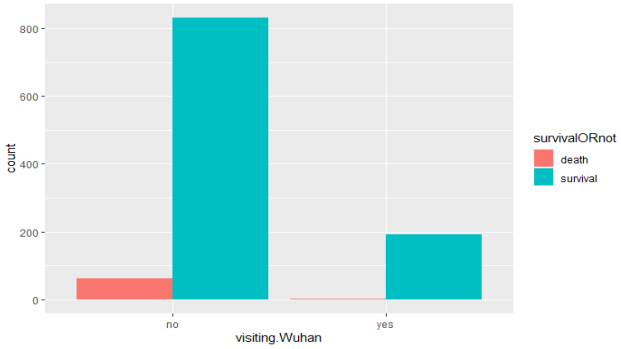


Figure 1. Death/Survival VS Visiting Wuhan

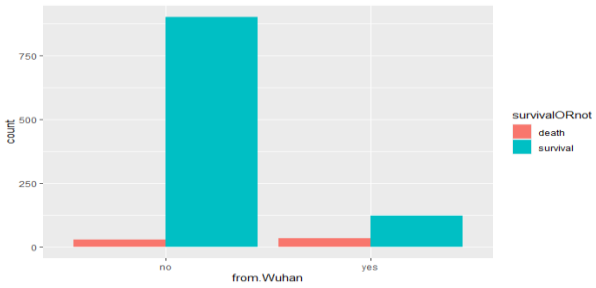


Figure 2. Death/Survival VS From Wuhan

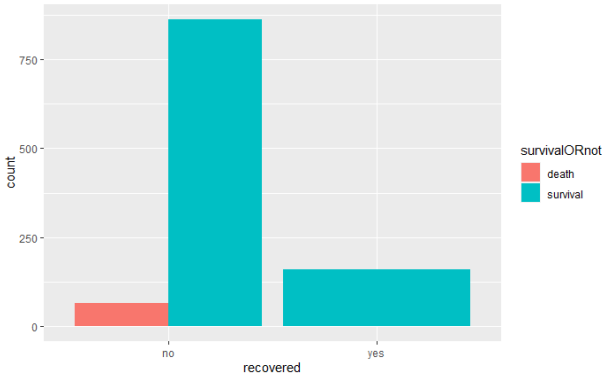


Figure 3. Death/Survival VS Recovered

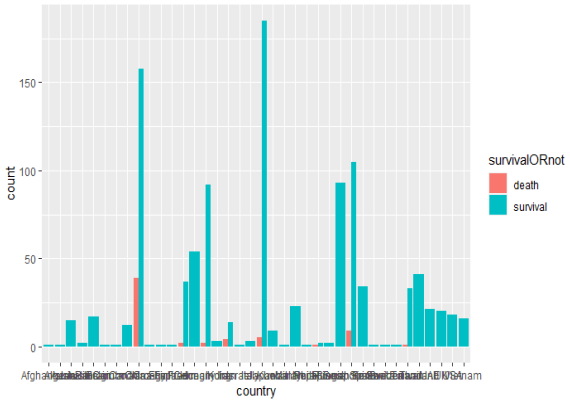


Figure 4. Death/Survival VS Country

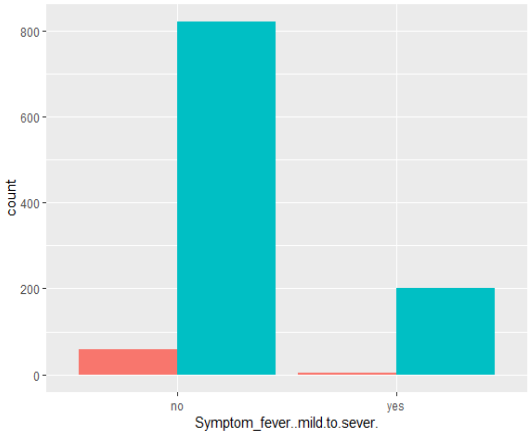


Figure 5. Death/Survival VS Fever Mild to Sever

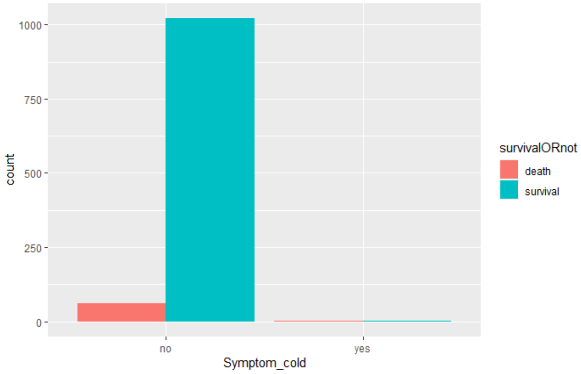


Figure 6. Death/Survival VS Cold

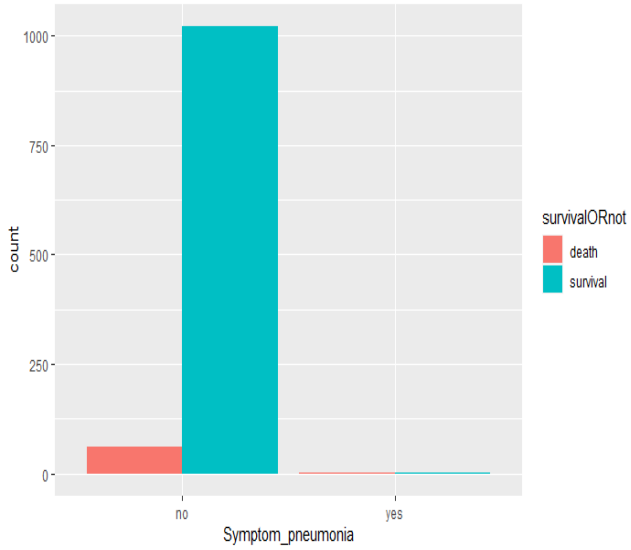


Figure 7. Death/Survival VS Pneumonia

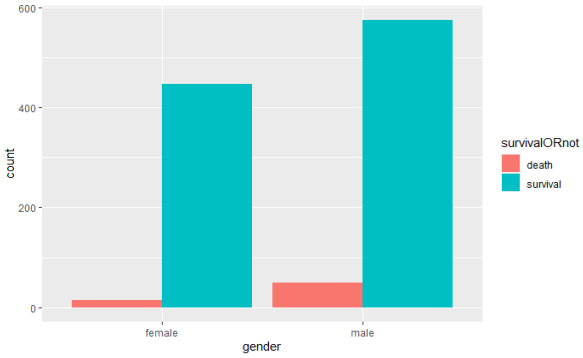


Figure 8. Death/Survival VS Gender

Correlation between the outcome and numeric variables

We then compared our outcome variable with the numeric variables in the dataset. We conducted the two-sample t-test followed by producing side-by-side boxplots as seen in figures 9-11 with p-value less than the significance level of 0.01. All these variables would be lower and is seen as strongly associated with the outcome variable. In addition, we can clearly notice by the side-by-side boxplots that the means are noticeably different.

NO	VARIABLE	P-VALUE
1	Age	2.423e-16
2	SymptomLength	2.195e-09
3	ReportingLength	3.164e-07

Table 5. P-value for Welch Two Sample t-test

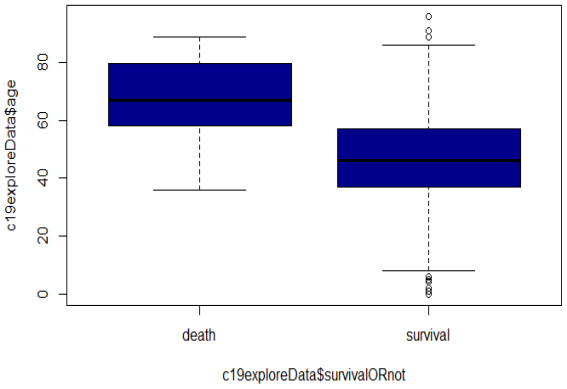


Figure 9. Survival/Death and Age

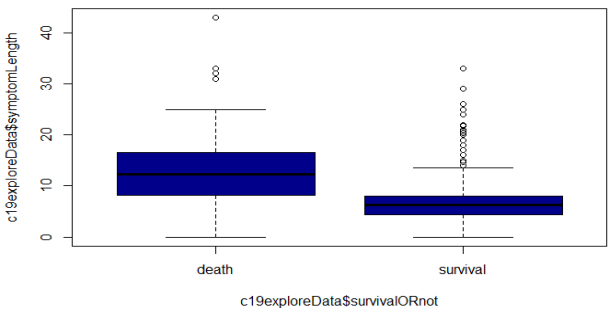


Figure 10. Survival/Death and Length of Symptom

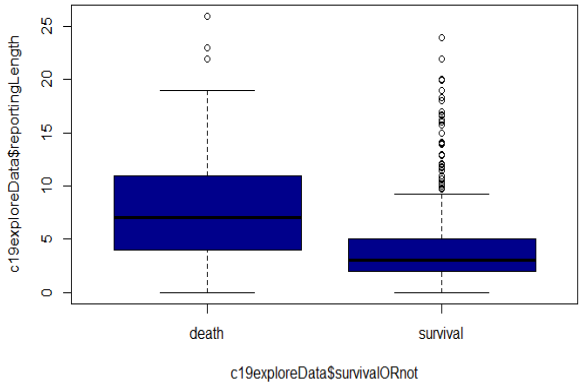


Figure 11. Survival/Death and Reporting Wait Time

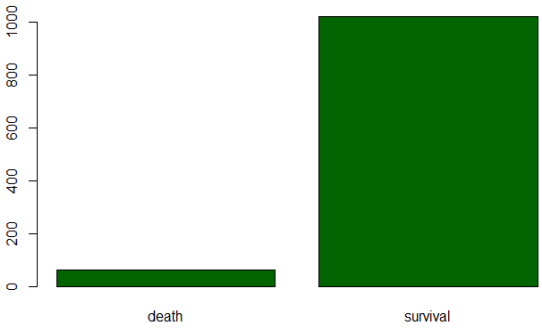


Figure 12. Death vs Survival

Balancing Data using SMOTE

Our dataset was almost ready to be applied to our algorithms, however; we faced a big problem. Our data was severely imbalanced, so we had to balance our data. First, we split our dataset into 80% training and 20% testing. We decided to apply the SMOTE method to our training data set as it is preferred by many over under sampling and over sampling. In Figure 13 you can see our data before SMOTE and figure 14 represents the data after applying SMOTE.

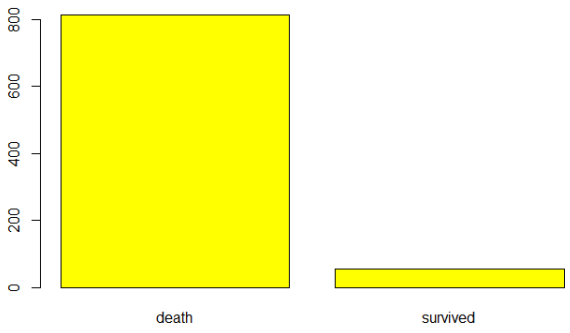


Figure 13. Death VS Survived after splitting data

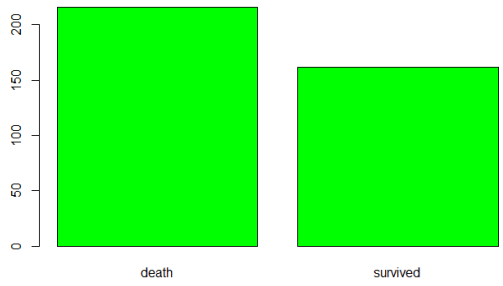


Figure 14. Death VS Survived after applying SMOTE on training data

Feature scaling and encoding on ANN and One-Class SVM

ANN

Our dataset was almost ready to be applied on our algorithms however, ANN and SVM required scaling and categorical feature encoding of data further. ANN would require us to apply these methods. We split our training data into 90% training and 10% validation. Then converted our categorical variables – gender and country to numeric. We then had to scale all our numeric variables -country, gender, age symptomLength and reportingLength. Next, calculated the mean

and standard deviation of the numeric data and applied this on the validation numeric data. We then repeated this step for our original train and test data.

One-Class SVM

One-Class SVM required us to convert our categorical features to numeric and scale all our numeric features. We then converted our outcome variable to TRUE and FALSE where TRUE represents the majority which is survival and FALSE represents death. We then split our data into 90% train and 10% testing.

DATA ANALYSIS AND EXPERIMENTAL RESULTS

We used the following algorithms:

- Regularized Logistic Regression
  - Lasso Regression
  - Ridge Regression
  - Elastic Net Regression
- Tree Ensemble Models
  - Random Forest Model
  - Gradient Boosted Tree Model
- K Nearest Neighbor
- Artificial Neural Network
- Anomaly Detection methods
  - Isolation Forest
  - One-Class Support Vector Machines

Regularized Logistic Regression, Tree Ensemble Models and K Nearest Neighbor

For our regularized logistic regression, tree ensemble models, and KNN we used carets package which allows for auto tuning. We used 10-fold cross validation for each of these models to give consistency in answers. In Lasso and Ridge, we tuned the lambda against 100 different values between 10 to the power of -3 and 3 to get the best lambda. Of the 78 variables used in Lasso, the algorithm shrunk 50 variables down to 0. The remaining variables are listed in table 6. In Elastic Net; in addition to tuning lambda on 100 different values as we did in Lasso and Ridge, we tuned alpha on 10 different values between 0 and 1. In KNN, we tried to get the best accuracy over 20 different numbers for K. We notice in figure 15 that K of 7 gives us the best accuracy in this model. In table 7, we see the list of importance produced by Random Forest Model over all the variables.

VARIABLES	
countryChina	4.049555
countryFrance	3.332274
countryHong Kong	0.812082
countryIran	4.011311
countryJapan	-0.80942
countryPhillipines	5.64017

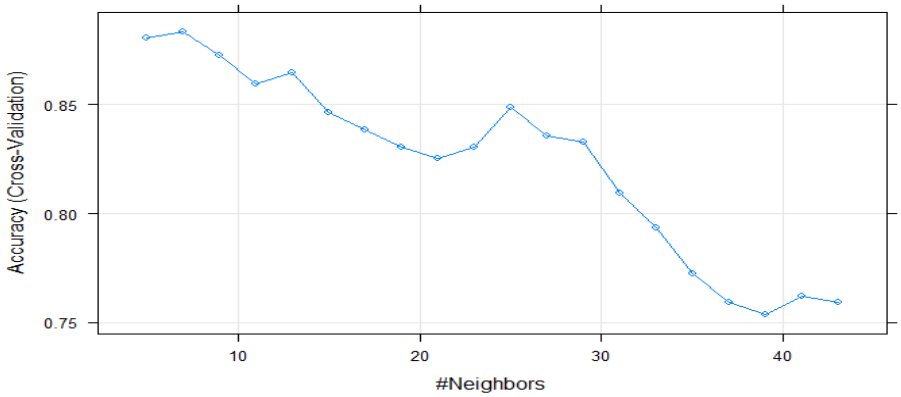
countrySingapore	-5.62323
countrySouth Korea	4.577487
countrySwitzerland	-2.1996
countryTaiwan	-1.7392
countryThailand	-0.18687
countryUAE	-0.96953
gendermale	2.151515
age	0.111793
If_onset_approximated	0.110745
visiting.Wuhan	-3.32628
recovered	-1.64861
symptomLength	0.108225
reportingLength	0.431281
Symptom_fever..mild.to.sever.	-2.90801
Symptom_coughing..mild.to.sever.	3.251137
Symptom_chills	-9.76062
Symptom_diarrhea	-6.93578
Symptom_pneumonia	7.385025
Symptom_sputum	-6.71851
Symptom_myalgia	14.04022
Symptom_sore.throat	-0.02269
Symptom_dyspnea	0.005529

Table 6. Variables not made 0 by Lasso

VARIABLE	IMPORTANCE
Age	100.000000
from.Wuhan	77.649735
reportingLength	65.659321
symptomLength	60.060861
countryMalaysia	33.507065
countryChina	33.025892
Recovered	30.924717
countrySouth Korea	26.961547
Gendermale	25.829086
countryThailand	20.196759
Symptom_sputum	16.282432
Symptom_pneumonia	14.808265
Symptom_sore.throat	12.582844
countrySingapore	11.867318
Symptom_fever..mild.to.sever.	10.106045
countryJapan	8.506078
Symptom_diarrhea	7.601071
countryTaiwan	7.588307
countryUAE	7.446168

visiting.Wuhan	7.420449

Table 7. Number of Importance in Random Forest Model



Figured 15. K-Nearest Neighbors and Accuracy

Artificial Neural Network

When we create our ANN, we train it on our training and validation data which we mentioned above. We created two flags in our model with two dropout levels. We had 2,592 combinations of that, 2% was sampled which left us with 51 different combinations to run our dataset over. From those we get the max metric_val_accuracy which gave us the best combination of flags - node1 392, batch_size 500, activation relu, learning_rate 0.001, epochs 30, dropout 0.5 and nodes2 392. There was no overfitting as val_loss was not always higher than our loss. We then ran this combination over our original training and testing data set. Figure 16 shows a plot of our test and training data.

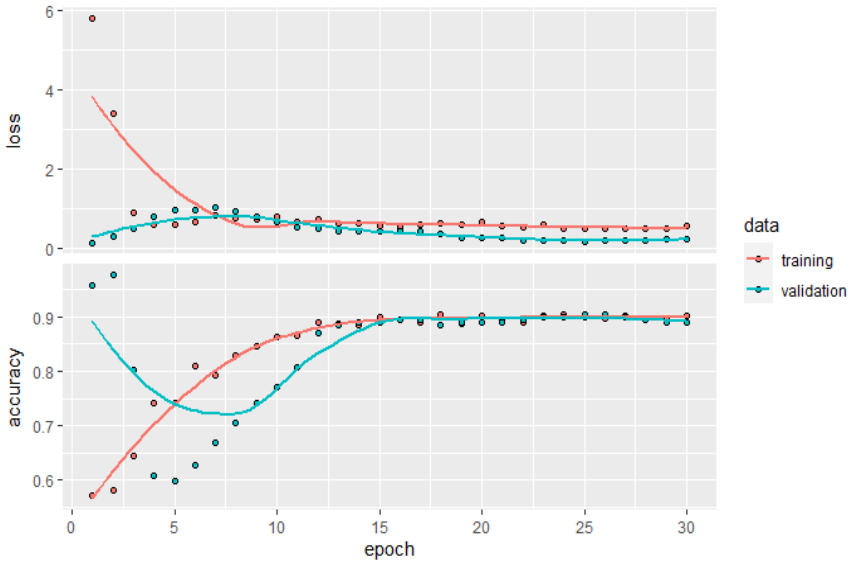


Figure 16. Epochs and Loss/Accuracy on training data

Isolation Forest

Due to our data being very bias we used Anomaly Detection methods such as Isolation Forest and One-Class SVM. Isolation Forest is similar to decision trees as it breaks down further into branches. Isolation Forest works by finding the datapoints which are far from others and isolating it. We built multiple trees as we try to find the average number of splits required to isolate a sample, by doing this we provide stability to the process. The Isolation Forest uses this method and produces values for different data points that indicates the 'level of anomaly or abnormality' [4]. In figure 17, we have compared the death and survival anomaly score. We notice that the means have a noticeable difference. Also, we notice that survival has many outliers whereas death has none.

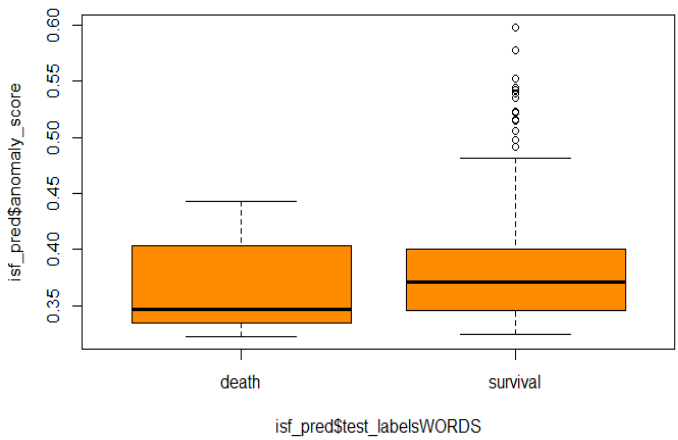


Figure 17. Death/Survival and Anomaly Score

One-Class Support Vector Machines

Unlike Isolation Forest, One-Class SVM produces a confusion matrix which we have compared to the outcome of our other methods. As mentioned earlier unlike our other algorithms where 1 indicated death, in One-Class SVM 0 indicates death. This is because SVM interprets the superior observation in the variable as TRUE and minority as FALSE. Therefore, a TRUE/FALSE confusion matrix is produced by One-Class SVM which can easily be interpreted into 0 and 1.

Goal of Paper

The goal of this paper was to predict the survival rate of patients who were tested as COVID-19 positive. After exploring these algorithms and looking at the outcomes, we noticed some algorithms produce particularly good accuracy; however, we receive extremely poor false negative rates. However, **random forest** is relatively the best choice as it gives the highest accuracy, kappa, AUC and a low FNR compared to other algorithms. The confusion matrix of Random Forest is shown in figure 23. SVM also gives an extremely low FNR, a low AUC and accuracy and an average kappa based on the other algorithms. With a better dataset, I believe we could improve the outcome of all algorithms as we would not have to deal with imbalanced data.

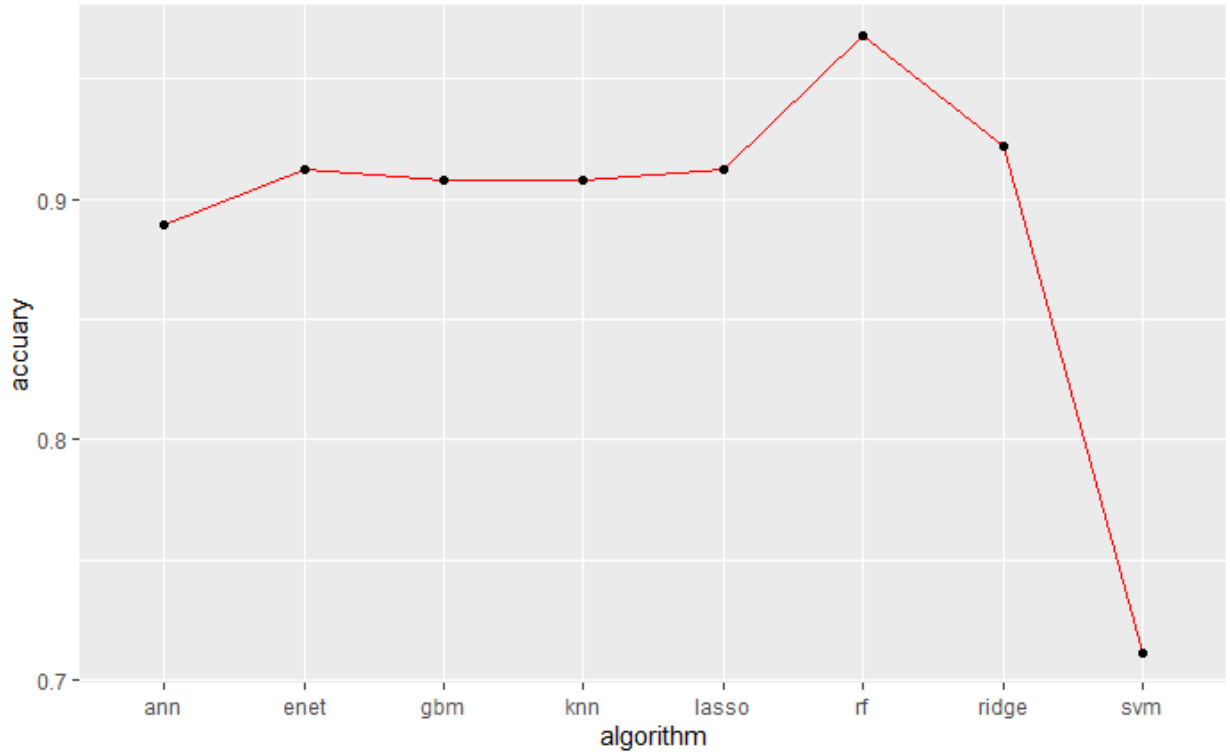


Figure 18. Accuracy of Algorithms

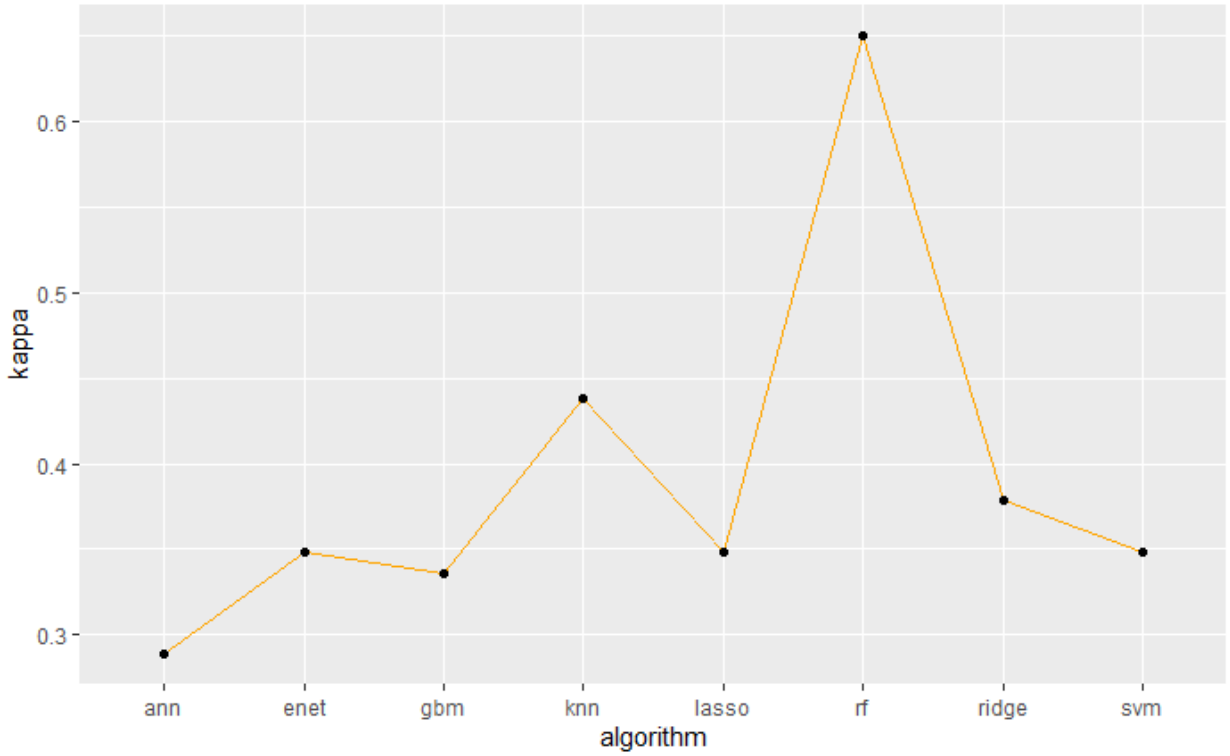


Figure 19. Kappa of Algorithms

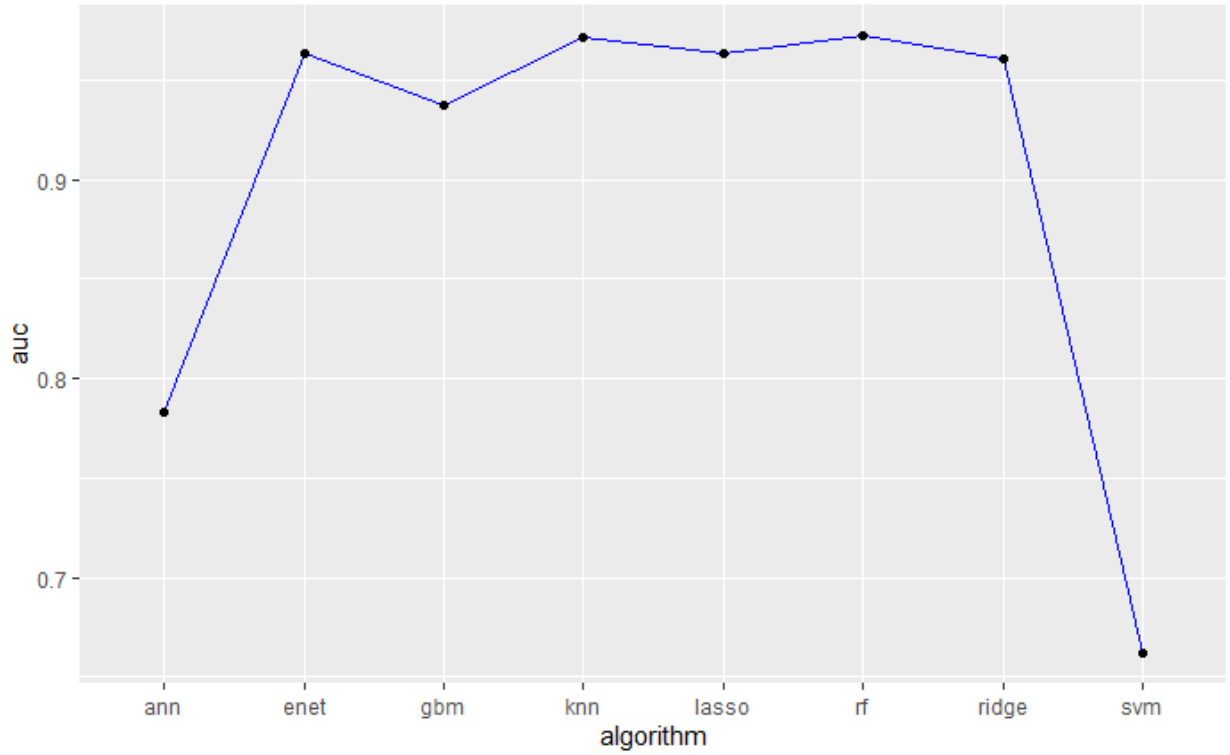


Figure 20. AUC of Algorithms

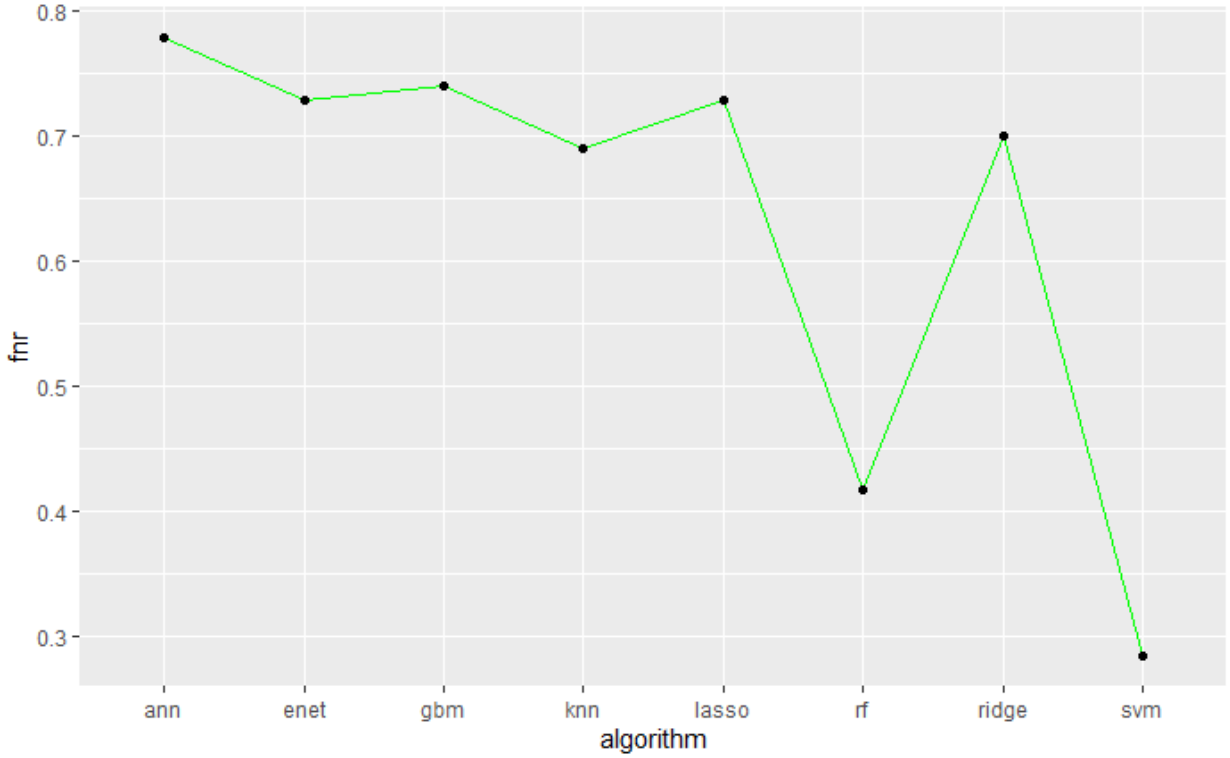
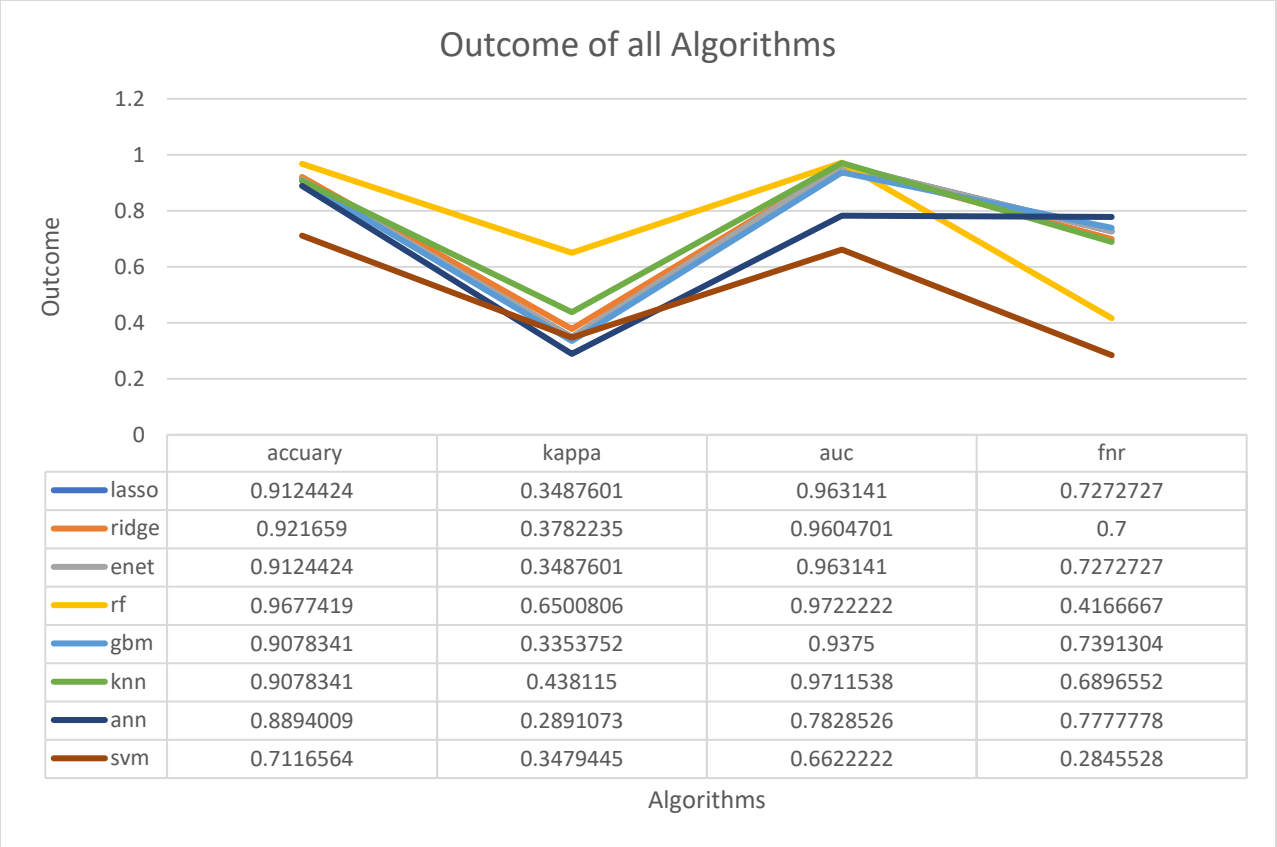


Figure 21. FNR of Algorithms



	accuary	kappa	auc	fnr
lasso	0.9124424	0.3487601	0.963141	0.7272727
ridge	0.921659	0.3782235	0.9604701	0.7
enet	0.9124424	0.3487601	0.963141	0.7272727
rf	0.9677419	0.6500806	0.9722222	0.4166667
gbm	0.9078341	0.3353752	0.9375	0.7391304
knn	0.9078341	0.438115	0.9711538	0.6896552
ann	0.8894009	0.2891073	0.7828526	0.7777778
svm	0.7116564	0.3479445	0.6622222	0.2845528

Figure 22. Outcome of all Algorithms

	Reference	
Prediction	0	1
0	203	2
1	5	7

Figure 23. Random Forest Confusion Matrix

CONCLUSION

This project has given us a wealth of knowledge. We have been able to understand that datasets with binary variables as the outcome variable must be handled differently, as the use of side-by-side boxplot and scatterplots will not supply useful data when trying to conduct data exploration. We also understand that ANN has a binary loss function which must be used, or the algorithm would not run. In addition, we learned how to rename, and split variables into binary columns in R along with learning the use of Anomaly Detection methods and the difference between One-Class SVM and the other types of SVM which exists. Our knowledge of the use of kappa and AUC were fined tuned. Most importantly we understand the difficulty and importance of a good dataset because without one your research may not yield satisfactory results.

Regarding our dataset, we noticed that some of the symptoms which Lasso did not shrink because of importance were mentioned in the updated list of symptoms which COVID-19 patients may present with.

A future research aspect may be the prediction of being positive for COVID-19 based on CT scans. In addition, countries in the Caribbean have been producing their own modeling because the modeling done by developed countries were not to the population scale, as most Caribbean Islands have hundreds of thousands of residents and developed countries have millions. The exploration of the difference between models can also be an interesting topic.

REFERENCES

[1] Naming the coronavirus disease (COVID-19) and the virus that causes it. (n.d.). Retrieved May 1, 2020, from https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it

[2] Li Yan, H.-T. Z., Goncalves, J., Xiao, Y., Wang, M., Guo, Y., Sun, C., … Yuan, Y. (2020, January 1). A machine learning-based model for survival prediction in patients with severe COVID-19 infection. Retrieved May 1, 2020, from https://www.medrxiv.org/content/10.1101/2020.02.27.20028027v3

[3] Pourhomayoun, M., & Shakibi, M. (2020, January 1). Predicting Mortality Risk in Patients with COVID-19 Using Artificial Intelligence to Help Medical Decision-Making. Retrieved May 1, 2020, from https://www.medrxiv.org/content/10.1101/2020.03.30.20047308v1

[4] DataVedas, & *, N. (n.d.). UNSUPERVISED ANOMALY DETECTION. Retrieved May 1, 2020, from https://www.datavedas.com/unsupervised-anomaly-detection/