DS 5100 Project Report: Analysis on COVID-19 Patients with Pre-existing Conditions

Group 5: Alex Bass, Connie Cui, Peumali Surani Withanage, Seth Galluzzi

Computing ID: ujb3bu, qqv3uu, upp2dh, vzw6yk

Introduction

As COVID-19 continues to impact the globe, it is imperative that we identify those around us who are the most vulnerable to death from the virus. Among those most vulnerable, are patients with pre-existing conditions. The aim of this project is to explore how pre-existing medical conditions impact COVID-19 patients in hospitals and to use our data to predict the probability of survival for a new patient. We will accomplish this by comparing mortality rates of patients with pre-existing conditions to mortality rates of patients without pre-existing conditions. We will explore visualizations with predictors such as pneumonia and age, and we will use the data to predict the probability of survival of a new patient. Once accomplished, we will share our knowledge in an engaging way by allowing users to explore how different medical conditions impact the probability of survival. The contents of this project can be found on the team's Github page.

Data

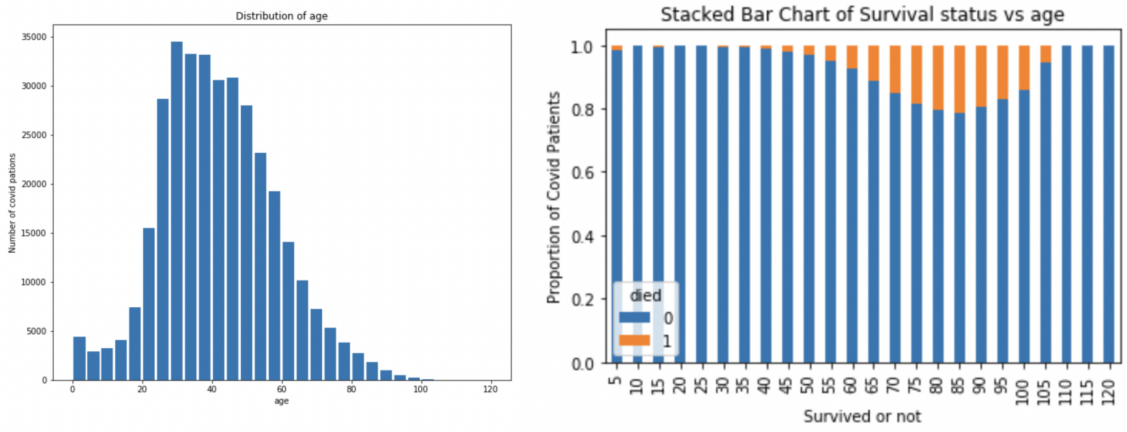
Our dataset was obtained from an online dataset released by the Mexican government with 100,000+ observations. For simplicity, we used the translated version of the data posted on kaggle. The dataset primarily consisted of information about hospitalized patients in Mexico. Information such as age and sex was collected, as well as other information regarding medical conditions (i.e. diabete, asthma, hypertension, etc.), and whether or not the patient died during hospitalization. We chose this dataset specifically due to the abundance of information regarding patients and their pre-existing medical conditions. The data contained a large number of categorical variables with values 0 and 1. To process the data, we first identified and removed the null values. We decided to remove the null values because they were not imperative in our analysis. We also removed an unnamed column that contained our index values, and then split our dataset into training and testing components. Below, Figure 1 illustrates the first few rows of our cleaned dataset.

	sex	patient_type	pneumonia	age	diabetes	copd	asthma	inmsupr	hypertension	other_disease	cardiovascular	obesity	renal_chronic	tobacco	contact_other_covid	covid_res
0	0.0	1.0	0.0	27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
1	1.0	1.0	0.0	56	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0
2	1.0	1.0	0.0	34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
3	1.0	1.0	0.0	34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0
4	1.0	1.0	0.0	49	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0

Figure 1: Header of Data

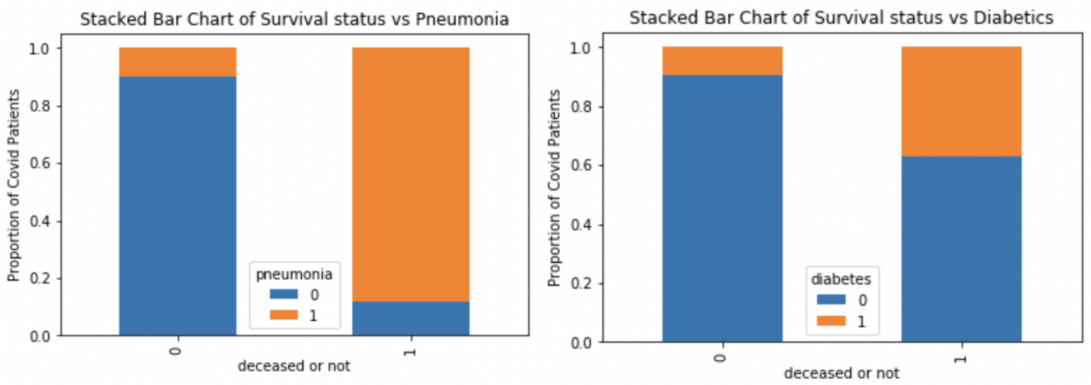
Analytical Design

During the exploration phase of this project, we considered many variables when trying to determine death. The first predictor that indicated a clear relationship with death was age. Thus, we decided to highlight this variable within our report. Below Figures 2 and 3, display the distribution of ages among covid patients as well as a stacked bar chart depicting the proportion of deaths by age. The distribution on the left indicates a large number of patients are between 20-60 years old, however, the visualization on the right indicates the largest proportions of deaths among patients have been between 60-100 years old. Thus, the data indicates there is a heightened risk of death among older patients with COVID-19.



Figures 2 and 3: Distribution of Age and Survival by Age Bar Chart

After exploring age, a multitude of pre-existing conditions were also explored such as obesity, diabetes, asthma, pneumonia, and hypertension. Pneumonia and diabetes both indicated a large proportion of deaths among patients. The bar charts in figures 4 and 5 depict the relationship pneumonia and diabetes share with death respectively.



Figures 4 and 5: Pneumonia and Diabetes Bar Charts

In order to help predict the survival rate of a hospitalized patient that has contracted COVID, we first needed to fit a model that could do so. To this end, we explored using machine learning classifiers such as logistic regression, decision trees, and random forest models to help both predict our survival rate probability. Before fitting our models, we noticed in our preliminary exploratory data analysis that there was a significant imbalance of values within our response variable, with a large majority of our patients in the data surviving versus dying (Figure 6).

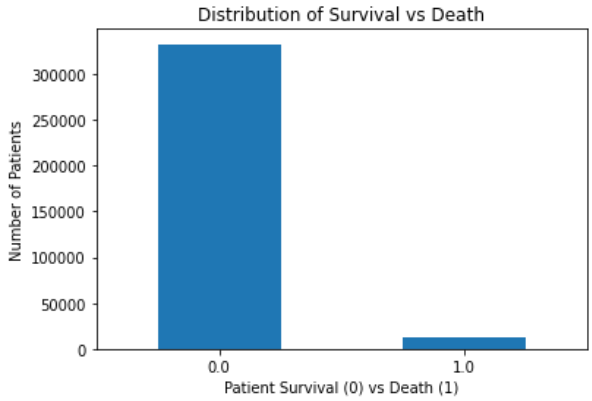


Figure 6: Distribution of Response Variable Patient Survival

While this is obviously preferable realistically speaking, datawise it is indicative of an imbalance dataset, one that will lead to Type II error if not addressed. Type II error can also be defined as a false negative, which in context with our project implies that a patient that is likely to die from COVID while hospitalized is predicted by our model to not die. We believe that this should be avoided, and in order to do so, we created a new dataset with the use of oversampling to duplicate rows of the minority group (patients that died) so that there were an equal number of

observations for each response variable value. The benefits and impact of balancing our data can be seen below in Figure 7, with metric scores for the model trained on the balanced dataset being overall much better than the one trained on the original dataset.

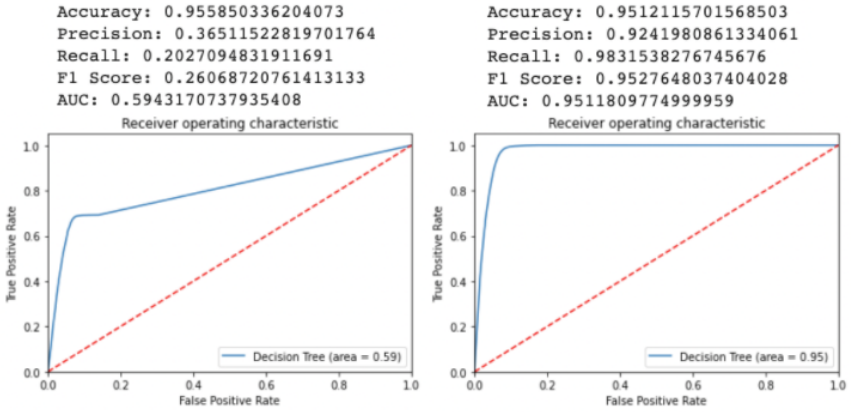


Figure 7: Side by Side Comparison of ROC Curves and Metric Scores of Decision Tree Models Trained by the Original Dataset (left) and Balanced Dataset (right)

In order to determine the best overall model to help predict survival rates for our hospitalized COVID patients, we trained three classifier models (logistic regression, decision tree, random forest) on both our original dataset and our balanced dataset then compared their metric scores, specifically accuracy ((TP+TN)/(TP+FP+FN+TN)) and F1 score (2*(Recall * Precision) / (Recall + Precision), where Recall = TP/TP+FN and Precision = TP/TP+FP) (metric scores displayed below Figure 8).

Original Dataset	**Logistic Regression**	**Decision Tree**	**Random Forest**
Accuracy	0.9613	0.9559	0.9569
F1 Score	0.2424	0.2607	0.2769

Figure 8: Table of Accuracy and F1 Scores from Three Classifiers Trained on the Original Dataset

As can be seen in Figure 9 above, while the accuracy scores for all three classifiers are quite high, the F1 scores, which take into account false negatives and false positives, is very low.

Balanced Dataset	**Logistic Regression**	**Decision Tree**	**Random Forest**
Accuracy	0.9148	0.9513	0.9520

F1 Score	0.9147	0.9513	0.9519

Figure 9: Table of Accuracy and F1 Scores from Three Classifiers Trained on the Balanced Dataset

As shown in Figure 9, we can see that while there isn't much of an improvement in the accuracy scores compared to the models trained on the original dataset, the F1 scores drastically improved, making it likely that we will be choosing one of these three models as our optimal model. After comparing the three models and their scores in the figure above, it's clear that the random forest model trained on the balanced dataset is our best model.

From this model, we can also glean insight into feature importance to try and determine which of our predictor variables seem to be the most influential in predicting the survival rate. Based on the results found in the figure below, it appears that patient type (whether the patient was hospitalized or simply sent home after the check up), age of patient, and whether or not the patient had pneumonia during the time were the most important predictor variables in our optimal random forest model trained on the balanced dataset.

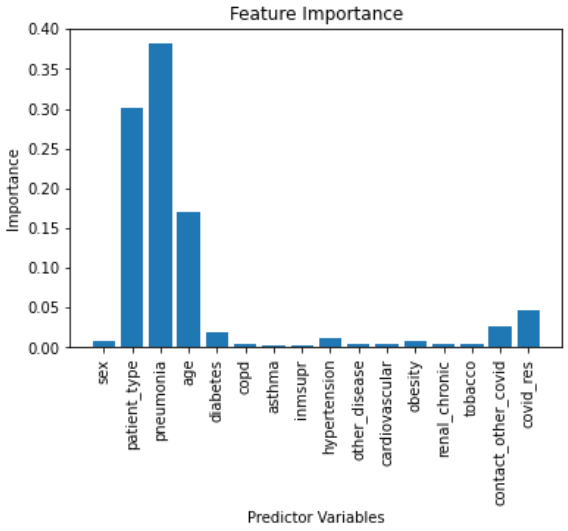


Figure 10: Feature Importance Based on Our Optimal Model

After completing our analysis on how pre-existing medical conditions impact patients with COVID-19. It was imperative that we shared our newfound understanding with others. When presenting knowledge, it is important to present it in a way that is easy to comprehend and encouraging to explore. Thus, along with our visualizations, we also chose to create a command line interface for the user to input different medical conditions and receive the probability of

survival based on those inputs. This code is easily downloaded via our instructions on our github repository. We expect users to interact with this interface in at least two ways:

1. We expect our users to input information about themselves and their family members to identify how COVID-19 impacts them personally.
2. We expect our users to explore various mock scenarios so the user can understand what factors contribute the most to survival rate from hospitalized COVID-19 patients accounting for various pre-existing conditions and important demographic information.

Below in Figure 11, is a portion of the code written to collect user data. Figure 12, depicts a portion of the interface the user would interact with.

```
class Predict_user():
    def __init__(self):
        #Gathering user data on input

        #printing intro text welcoming user
        print('\n\n\nHi! Please answer these questions to get a personalized probability of survival if hospitalized for COVID-19.\n\n')


        #Gathering user data for sex

        while True:
            Sex = input('\nWhat is your sex? M or F ')

            #Text Sanitization
            Sex = Sex.strip().capitalize()

            if Sex == "F" or Sex == "M":
                if Sex == "M":
                    self.Sex = 1
                else:
                    self.Sex = 0
                break

            print("\n\nPlease enter 'M' or 'F' to proceed.\n\n")


        #Gathering user data for age

        while True:
            Age = input('\nWhat is your age? Please enter a number:      ')

            #Text Sanitization
            Age = Age.strip()

            #Text Checking
            valid_age = bool(re.search('^[0-9]{1,2}$', Age))

            if valid_age:
                self.Age = int(Age)
                break

            print("\n\nPlease enter a number from 0 to 99.\n\n")
```

Figures 11 and 12: User Input Code and User Interface

```
Hi! Please answer these questions to get a personalized
 probability of survival if hospitalized for COVID-19.

What is your sex? M or F       F

What is your age? Please enter a number:       37

Do you have diabetes? Y or N       Y

Do you have athsma? Y or N       N

.......

Here are your results...

After loading YOUR data into our model, we predict you
have a 72.0% percent chance of survival if hospitalized
 for COVID-19
```

We hope that by creating this interface, we can encourage users to interact with our data, and to gain the understanding that pre-existing conditions can have a major impact on the probability of survival among COVID-19 patients.

Testing

Unit testing was used to ensure the quality of code, and to ensure user input was successfully collected. During testing, 14 tests were successfully run. Each test ensured that user input value's such as sex, age, and tobacco use would be collected correctly. A portion of the tests and output is displayed below in figures 13 and 14. The inputs for this specific test are a male smoker who suffers from chronic inflammatory lung disease and has been in contact with a person who has COVID-19.

```
z# import the unit test package
import unittest

# create a user with following attributes when prompted:
    # male, smokes, contact with other with covid, has copc
    # tests below to ensure the values in our user input cl

test_user = Predict_user()

#create a set of unit tests to ensure each user input value

class MyTest(unittest.TestCase):

    def test_the_sex(self):
        self.assertEqual(test_user.Sex, 1)

    def test_the_diabetes(self):
        self.assertEqual(test_user.diabetes, 0)

    def test_the_athsma(self):
        self.assertEqual(test_user.athsma, 0)

    def test_the_hypertension(self):
        self.assertEqual(test_user.hypertension, 0)

    def test_the_obesity(self):
        self.assertEqual(test_user.obese, 0)

    def test_the_pneumonia(self):
        self.assertEqual(test_user.pneumonia, 0)

    def test_the_otherdisease(self):
        self.assertEqual(test_user.other_disease, 0)
```

```
test_the_age (__main__.MyTest) ... ok
test_the_athsma (__main__.MyTest) ... ok
test_the_cardiovascular (__main__.MyTest) ... ok
test_the_contact (__main__.MyTest) ... ok
test_the_copd (__main__.MyTest) ... ok
test_the_diabetes (__main__.MyTest) ... ok
test_the_hypertension (__main__.MyTest) ... ok
test_the_obesity (__main__.MyTest) ... ok
test_the_otherdisease (__main__.MyTest) ... ok
test_the_patient_type (__main__.MyTest) ... ok
test_the_pneumonia (__main__.MyTest) ... ok
test_the_renalchronic (__main__.MyTest) ... ok
test_the_sex (__main__.MyTest) ... ok
test_the_tobacco (__main__.MyTest) ...

Here are your results...

After loading YOUR data into our model, we predict you


ok

----------------------------------------------------------
Ran 14 tests in 0.027s

OK
<unittest.main.TestProgram at 0x7f8c0fc355d0>
```

Figures 13 and 14: Unit Testing Code and Output

Beyond the Required Specifications

Several components of our project went beyond the required specifications of the assignment. Furthermore, some of the coolest and most insightful components of our project were created to purposefully dive deeper into the exploration. For example, we used machine learning to create our predictive model for the data. We also created user interaction that provided an effective and impactful way of conveying our findings by building a command line interface. We both gathered data from the user from this command line interface, and also implemented in real-time the user data to see personalized model results. Most importantly, we worked as a team, communicated effectively, and used our knowledge gained throughout this semester to produce a valuable exploration into COVID-19 patients with pre-existing conditions.

Result and Conclusion

The aim of this project was to explore how pre-existing medical conditions impact COVID-19 patients in hospitals, and to increase awareness by creating an interactive user experience that predicts the probability of survival for a new patient with certain pre-existing conditions.

We accomplished our aim by comparing mortality rates of patients with pre-existing conditions to mortality rates of patients without pre-existing conditions, and we explored visualizations with predictors such as pneumonia, age, and diabetes with data obtained from the Mexican government. Fortunately, we learned mortality rates for patients with COVID-19 were low. However, patients with pre-existing conditions such as pneumonia and diabetes had higher mortality rates than patients without those pre-existing conditions. We then used machine learning to train six models- three with unbalanced data and three with balanced data. The classifiers used were logistic regression, decision tree, and random forest. The balanced and unbalanced data both proved to have similar accuracy scores (0.95). However, the balanced model was much more effective with precision, recall, F1 score, and AUC(all greater than 0.9). The optimal model found was a balanced data set with a random forest classifier. Using the optimal model we found age, patient type and pneumonia were the most accurate predictors of mortality.

To make the presentation of our findings impactful, a program was designed and tested where users could input pre-existing conditions and receive an output displaying the probability of survival if admitted to the hospital with COVID-19. To further our exploration, we could compare our data from Mexico with data from other countries. We could also make our user experience more insightful by including appropriate visualizations along with the probability of survival within the output. Throughout this exploration we gained knowledge of how pre-existing conditions can impact COVID-19 patients, we displayed the results of our exploration in an impactful way, and we gained further understanding of tools used in data science.

References

AAT1iresh. (n.d.). *AAT1IRESH/ds5100_project_covid: This repository is created to facilitate DS5100 final project.* GitHub. Retrieved December 10, 2021, from https://github.com/AAT1iresh/DS5100_Project_Covid.

Datos Abiertos Dirección general de epidemiología - gob.mx. (n.d.). Retrieved December 10, 2021, from https://www.gob.mx/salud/documentos/datos-abiertos-152127.

Nikalje, M. (2020, July 28). *Covid-19 patient pre-condition dataset (cleaned)*. Kaggle. Retrieved December 10, 2021, from https://www.kaggle.com/madan44/covid19-patient-precondition-dataset-cleaned.