Data 670 Data Analytics

Theodore Fitch

Professor Hany Saleeb

Assignment 6

December 10th 2024

Executive Summary

This project integrated two COVID-19 patient datasets from the Mexican Ministry of Health and analyzed the joint, unified dataset to develop predictive models that forecast patient outcomes, such as ICU admission and mortality rates. The variables included age, sex, pre-existing health conditions, hospital treatment data, and comorbidities. The datasets were cleaned using data engineering techniques to address inconsistencies, standardize variables, and delete any missing values, ensuring data integrity. Two features were engineered to aid in analysis. The project's primary business objective is to identify key risk factors influencing severe patient outcomes and translate these findings into actionable insights for healthcare providers and policymakers to enhance resource allocation and patient care strategies. Key performance indicators (KPIs) included achieving a model accuracy (achieved), sensitivity (achieved), and F1 score (not achieved) of over 90%, successfully integrating datasets with over 95% completeness (achieved), and ensuring that the top five features explain more than 80% of outcome variance (achieved). Exploratory data analysis (EDA) was conducted and key visualizations were generated to identify patterns and trends, such as correlations between comorbidities and mortality rates. Predictive modeling used 14 machine learning algorithms such as Random Forest, Logistic Regression, and Gradient Boosting to forecast patient outcomes. The champion models were a decision tree to predict mortality and a random forest for ICU admission. A risk matrix was generated to quantify the impact of each variable on outcomes, bridging the gap between raw data analysis and actionable healthcare strategies. In summary, the expected outcomes were achieved: to make a data-driven foundation for optimizing pandemic response and healthcare policies.

Table of Contents

Project Scope

Problem Description

This project focuses on integrating and analyzing two datasets containing COVID-19 patient data from multiple sources on Kaggle to gain a comprehensive understanding of patient demographics, health conditions, and treatment outcomes. Given the immense volume of data generated during the pandemic, this integration is crucial for building a reliable foundation for analysis. By employing advanced data engineering and modeling techniques, I aim to create a unified dataset that can effectively predict critical patient outcomes, such as ICU admissions and mortality rates. The datasets, which provide insights into various patient factors, will undergo a thorough process of integration, cleaning, exploratory analysis, modeling, and interpretation. This systematic approach will facilitate a deeper understanding of the underlying data. It will also highlight the correlations between specific health conditions and patient outcomes. Ultimately, the goal of this analysis is to identify the patient factors that most significantly influence outcomes like ICU admission and mortality, thereby equipping healthcare workers and policymakers with actionable insights.

The project will focus on two predictive modeling objectives: first, to accurately predict patient mortality and second, to forecast patient admission to the ICU. Both outcomes are critical, especially in emergency situations where timely and informed decisions can make a significant difference in patient care. Understanding mortality risk is essential for developing targeted interventions, while predicting ICU admissions allows healthcare systems to allocate resources more effectively during surges in patient volume. In the context of the ongoing COVID-19 pandemic, these predictive models can provide

invaluable guidance to healthcare professionals as they navigate the complexities of patient management in crisis situations. By quantifying the risks associated with various patient demographics and health conditions, this project seeks to contribute to a more nuanced understanding of how to triage care effectively, ensuring that high-risk patients receive the necessary interventions promptly.

This problem is of paramount importance, particularly in light of the challenges faced during the pandemic when healthcare resources were stretched thin. The COVID-19 pandemic effects were felt worldwide and the consequences are still being felt today. Integrating datasets from various sources is inherently complex due to differences in data structure, variable definitions, and potential inconsistencies. Therefore, a robust integration process is necessary to create a comprehensive dataset that accurately reflects the reality of patient experiences during COVID-19. Moreover, developing predictive models requires not only technical skills in data engineering and machine learning but also a deep understanding of the clinical context to ensure that the models are relevant and actionable. By addressing this problem, the project aims to contribute meaningful insights that can enhance patient care strategies and improve decision-making processes. This will ultimately reduce mortality and optimize the use of critical resources. The data analytics problem that I am analyzing is how to integrate diverse COVID-19 datasets effectively and develop predictive models to identify the critical factors influencing patient outcomes, such as mortality and the need for ICU admission.

Business Understanding

This project is rooted in the healthcare and public health management industry, focusing specifically on the application of data analytics to pandemic response and patient outcome optimization. The industry has faced unprecedented challenges due to COVID-19, leading to a greater reliance on data-driven insights to make informed decisions. As healthcare systems were overwhelmed by the influx of patients, accurate prediction models became critical for resource allocation, patient management, and policymaking. This project leverages advanced machine learning techniques and data engineering processes to analyze patient-level data and provide actionable insights for stakeholders within the healthcare sector. The integration of multiple datasets, each with hundreds of thousands of patient records, allows for a comprehensive understanding of how various factors influence patient outcomes. Ultimately, the project aims to enhance healthcare resilience by supporting evidence-based decision-making.

The organization primarily involved in this project is the Mexican Ministry of Health, which provided the datasets used for analysis. The Ministry has a vested interest in understanding patient demographics, comorbidities, and treatment outcomes to inform future public health strategies. However, while transparently applicable to itself, most of the findings of this analysis should be broadly applicable to other government run health organizations, as well as industry-based healthcare providers. By identifying key predictors of severe patient outcomes, such as ICU admissions and mortality, the project helps stakeholders prioritize healthcare resources and tailor interventions to at-risk populations. This initiative also has implications for international health organizations and research bodies focused on epidemiology, as findings could contribute to a global

understanding of COVID-19's impact. The expected outcomes of the project include not only improved patient care within Mexico but also a broader framework that can be applied across similar healthcare settings worldwide.

Organization

The project is most closely aligned with the Mexican Ministry of Health, a government body responsible for public health management, epidemiological research, and healthcare policy. This is necessary to study in particular because Mexico was ranked 3rd in the world in number of total COVID-19 deaths, totally 192,488 (comprising 7.33% of all global deaths) as on March 12th, 2021 (García-Guerrero et al., 2021). The Ministry is therefore motivated to understand its own trends in order to regulate health within Mexico. This organization oversees data collection and management of COVID-19 patient records, making it a crucial entity for understanding how different variables contribute to patient outcomes. Given the scale and complexity of the datasets, the Ministry of Health is focused on leveraging data analytics to enhance healthcare strategies and optimize patient management. Their role includes coordinating with hospitals and clinics across the country to ensure accurate reporting and effective utilization of healthcare resources. The findings of this project will provide the Ministry with insights into which patient populations are most vulnerable to severe outcomes, guiding targeted interventions and healthcare policies.

Additionally, the scope of this project extends to other types of organizations, such as public health agencies and research institutions. Public health agencies at the local and international levels can use the results to inform strategies for mitigating the

effects of pandemics on healthcare systems. Research institutions will benefit from the methodologies employed, as they can replicate or build upon this research to enhance their own studies. The collaboration of these organizations with the Mexican Ministry of Health ensures that the findings are disseminated widely, benefiting not only the healthcare system in Mexico but also contributing to global research on pandemic preparedness and response.

Stakeholders

The primary stakeholders for this project include healthcare providers, such as doctors, nurses, and clinical staff, who are directly involved in patient care. These stakeholders rely on accurate predictions of patient outcomes to make critical decisions about patient treatment and resource prioritization. For example, identifying which patients are most likely to need intensive care can help doctors prioritize resources like ICU beds and ventilators for those who need them most urgently. This not only optimizes patient care for administrators but also reduces the burden on frontline healthcare workers. Additionally, accurate forecasts of patient deterioration enable healthcare providers to intervene earlier, potentially reducing mortality rates and improving recovery outcomes. As such, healthcare providers will benefit greatly from the insights derived through predictive modeling and risk assessment.

Another significant group of stakeholders consists of hospital administrators and healthcare facility managers. These individuals are responsible for operational planning, resource allocation, and ensuring that healthcare facilities run efficiently. For them, the data-driven insights generated from this project are invaluable for forecasting demand,

managing supply chains, and planning staff schedules. By understanding patterns of ICU admission and patient outcomes, administrators can make informed decisions about resource distribution and capacity planning. This helps to prevent overcrowding and ensures that facilities are not overwhelmed during periods of high patient inflow, such as during a pandemic surge. Administrators will use these insights to balance patient care needs with operational constraints, thereby improving the overall performance of healthcare institutions.

Public health officials and policymakers form another critical stakeholder group. They are responsible for shaping health policies, implementing public health initiatives, and allocating resources at the community or national level. The findings from this project will enable them to identify high-risk patient populations, evaluate the effectiveness of health interventions, and design targeted public health strategies. For example, if the analysis reveals that certain comorbidities or demographic factors significantly increase the risk of severe COVID-19 outcomes, policymakers can prioritize vaccination or resource allocation for those populations. Additionally, by using the data to inform public health campaigns and preventive measures, they can reduce the spread of disease and lessen the overall burden on healthcare systems. This group's decisions have far-reaching impacts, making the insights generated from this project crucial for effective public health management and policy development.

Define Business Area

The business area for this project is healthcare analytics and public health management, which focuses on leveraging data-driven insights to optimize patient care,

resource allocation, and healthcare policy development. Healthcare analytics involves using advanced data processing, statistical analysis, and machine learning techniques to understand complex health trends and predict patient outcomes. This project specifically aims to support pandemic response efforts by analyzing patient-level data to forecast the likelihood of ICU admissions and mortality, thereby enabling more effective management of healthcare resources. In addition, public health management is a critical component of this project, as the insights derived will inform strategies for handling future health crises and improving overall public health outcomes. By integrating large-scale datasets, this project provides a holistic view of patient demographics, comorbidities, and treatment responses, making it possible to identify at-risk populations and address health disparities. The business area also encompasses operational planning within healthcare facilities, where predictive modeling can support efficient staffing, supply management, and capacity planning. Overall, this project contributes to both the clinical and administrative dimensions of healthcare, making it a valuable tool for enhancing the performance of health systems and improving patient outcomes during pandemics or other public health emergencies.

Business Objectives

The first business objective of this project is to enhance patient outcome predictions using advanced machine learning models. By leveraging a unified dataset containing patient demographics, comorbidities, and clinical data, the project aims to develop predictive models that achieve high accuracy in forecasting ICU admissions and mortality rates. These models will help healthcare providers identify high-risk patients and prioritize their care accordingly. The goal is to achieve performance metrics, such as

accuracy, sensitivity, and F1 scores exceeding 90%, which will ensure that the models are reliable and effective in real-world scenarios. By providing early warnings about patients who are likely to experience severe outcomes, these models can enable healthcare providers to intervene sooner, optimize treatment plans, and reduce the overall mortality rate.

Another key business objective is to optimize resource allocation within the healthcare system by forecasting demand based on patient risk profiles. Accurate predictions of ICU admissions and resource needs, such as ventilators and staff, are critical for ensuring that hospitals are not overwhelmed during peaks in patient inflow. By having accurate models which can forecast rates of ICU admission and mortality, it can be determined which factors contribute most to these outcomes. This will be the basis on which to build a risk-matrix which would allow both healthcare workers and administrators alike to allocate care most efficiently. For example, if the models predict a surge in ICU admissions in a particular region based upon demographics and pre-existing condition rates of the general population, hospital administrators can proactively prepare by increasing staffing levels and securing additional medical supplies. This proactive approach will not only improve patient care but also reduce costs associated with emergency resource procurement and overtime labor.

A third business objective is to inform public health strategies and policy development by utilizing the aforementioned risk-matrix. The analysis will reveal which comorbidities, demographic factors, and clinical characteristics have the highest impact on patient outcomes, providing a data-backed foundation for targeted public health

interventions. Policymakers can use these insights to design focused health campaigns, prioritize vaccination or treatment for vulnerable groups, and allocate resources more effectively across different communities. For example, if certain age groups or comorbidities are identified as having a higher risk, public health campaigns can be tailored to those populations to promote early testing and preventive care. This will not only enhance the effectiveness of public health initiatives but also contribute to better overall health outcomes at the population level.

The final business objective is to enhance operational efficiency and reduce healthcare costs through improved inventory and capacity management. By integrating and analyzing data on hospital admissions, patient flow, and treatment protocols, the project aims to identify inefficiencies in current operations and suggest areas for improvement. For instance, understanding the patterns of patient admissions and discharges can help administrators optimize bed utilization and minimize patient wait times. Additionally, by accurately forecasting demand, the project can help hospitals reduce waste associated with over-preparation or under-utilization of resources. In the long run, this will result in cost savings for healthcare institutions, allowing them to reinvest in patient care and quality improvement initiatives. With data-driven resource management, hospitals can maintain high-quality care standards while operating more efficiently, even under the stress of pandemic conditions or other public health crises.

While this project encompasses multiple business objectives, such as enhancing resource allocation, optimizing healthcare resources and processes, and contributing to public health strategies, its ultimate aim is to save patients from severe healthcare

outcomes. By developing predictive models focused on ICU admission and mortality, the project seeks to proactively identify high-risk individuals and guide interventions that can prevent the progression of severe cases. These insights can support timely clinical decisions, better allocation of medical resources, and improved patient care protocols. Although optimizing logistics and improving public health infrastructure are important parallel goals, they all converge on the primary objective of preventing critical outcomes and saving lives through data-driven, patient-centered strategies.

Business Success Criteria

A robust data integration and preprocessing phase is essential to ensure that the final dataset is accurate, consistent, and free of errors. This involves carefully aligning variables across datasets, addressing missing values, standardizing formats, and resolving any discrepancies between the datasets, which can often be a time-consuming but critical process. Achieving this outcome requires a deep understanding of the dataset structures and content, as well as insight into potential data quirks and inconsistencies that may arise during integration. Tools such as Python's pandas and NumPy will be instrumental for performing complex data manipulation tasks, enabling efficient handling of large datasets and intricate transformations. Throughout the integration process, regular data validation and quality checks must be conducted to ensure data integrity is maintained at every stage, minimizing the risk of propagating errors into the modeling phase. The team must also possess strong technical skills in data engineering and data cleaning techniques, as these skills are foundational to producing a reliable dataset. By prioritizing this step, the project sets a strong foundation for the subsequent analysis, increasing the likelihood of generating valid and actionable insights.

To deliver high-performing predictive models, the project team must possess expertise in machine learning and statistical modeling. This includes not only selecting the right algorithms but also understanding their underlying assumptions and suitability for the problem at hand. Fine-tuning hyperparameters and implementing cross-validation strategies are crucial steps to avoid overfitting, ensuring that the models can generalize well to new data rather than just performing well on the training set. The desired outcome is to achieve high model performance metrics, such as accuracy, sensitivity, and F1 scores above 90%, which will ensure that the models are effective in predicting critical patient outcomes like mortality and ICU admission. Utilizing tools like Python's scikit-learn and XGBoost allows for leveraging a wide range of algorithms, from simpler models like logistic regression to more complex ones like gradient boosting, each tailored to different aspects of the analysis (Wade and Glynn, 2020). Additionally, the ability to interpret model results and understand feature importance is key to refining models and making data-driven adjustments. Team members need strong analytical skills to scrutinize these insights, making informed decisions that ensure the models are both predictive and robust, ultimately guiding effective healthcare interventions.

Clear and compelling communication of findings is crucial for this project to influence decision-making and support healthcare strategies. The value of advanced analytics is significantly diminished if insights cannot be translated into actionable strategies that resonate with healthcare providers, policymakers, and other stakeholders. This requires the ability to distill complex analytical results into clear, meaningful insights that are easily understood by stakeholders with varying levels of technical expertise, from clinicians to administrators. To achieve this, proficiency in data

visualization tools like Python's matplotlib, seaborn, and plotly is necessary, allowing for the creation of graphs and charts that succinctly capture trends and key findings. Experience in creating interactive dashboards using platforms like Tableau will further enable dynamic data exploration, making the insights more accessible and tailored to specific needs. Additionally, strong presentation and communication skills are essential to effectively convey results, emphasizing the practical implications of the findings and recommending evidence-based solutions. By doing so, the project's outcomes will be more likely to drive meaningful changes in patient care and resource allocation, ensuring that the analytical work translates into tangible improvements in healthcare management.

Background

Healthcare analytics has played a pivotal role in understanding and managing the impact of COVID-19 on global health systems. Since the outbreak of the pandemic, data scientists and healthcare professionals have collaborated to analyze large-scale datasets containing patient demographics, comorbidities, and clinical outcomes to gain deeper insights into the disease's progression and its effects on various populations. These analyses have helped identify critical risk factors associated with severe outcomes, such as age, gender, pre-existing health conditions, and geographic location. Early studies focused on understanding patterns of disease transmission and identifying high-risk groups, which contributed to the development of strategies for controlling the spread of the virus. As the pandemic progressed, research shifted towards evaluating treatment effectiveness, predicting healthcare resource needs, and developing predictive models to anticipate patient outcomes, such as ICU admissions and mortality rates. Advanced machine learning and statistical techniques have been employed to create models that

guide clinical decision-making and optimize resource allocation within overwhelmed healthcare facilities. Additionally, healthcare analytics has been instrumental in evaluating the efficacy of interventions like social distancing, vaccination rollouts, and public health campaigns. Despite these advancements, there remain some gaps in integrating diverse datasets and applying predictive insights to inform healthcare policy and operational strategies.

Research

There has been extensive research and analysis using similar COVID-19 datasets across various platforms, with Kaggle alone hosting 37 entries for the first dataset (Mukherjee, 2020) and 102 entries for the second dataset (Nizri, 2023). Many data scientists have leveraged these datasets to perform EDA and predictive modeling, often focusing on understanding patient demographics, health conditions, and their correlations with COVID-19 outcomes. Visualizations such as correlation heatmaps, distributions of comorbidities, and mortality rates are commonly used to provide an initial understanding of the data. A significant portion of research has focused on building predictive models to forecast patient outcomes, namely mortality, using machine learning techniques like Random Forest, Logistic Regression, and Gradient Boosting. The predictive models developed from these datasets typically achieve accuracy scores ranging from 80% to 92%. While these studies offer valuable insights, their primary focus has been on model development and performance, rather than on the application of findings to inform healthcare policies or actions. The vast majority of these however, only perform a basic analysis of the data and do not offer multiple models for comparability or EDA providing depth of insight into data trends. One of the most comprehensive studies on these datasets

is discussed in a paper published by the National Center for Biotechnology Information (Almustafa, 2022), which achieved a model accuracy slightly exceeding 94%. This study utilized advanced machine learning techniques and provided a thorough discussion on the predictive capabilities of various models.

Gaps in this Problem Resolution

While numerous studies have focused on predictive modeling using these datasets, there are several critical gaps that this project aims to fill. First, many existing analyses do not provide a comprehensive examination of which specific variables contribute most to patient risk, and they fail to build a risk matrix that evaluates the impact of each variable individually. This project will address this gap by not only identifying the top risk factors for severe COVID-19 outcomes but also by constructing a risk matrix to quantify the influence of each variable on patient outcomes.

Second, most studies have not utilized the predictive insights to recommend actionable strategies or healthcare policies. This project will translate model findings into concrete recommendations, providing a framework for healthcare providers and policymakers to prioritize high-risk patients, allocate resources effectively, and develop targeted interventions. By focusing on the practical applications of predictive modeling, this project aims to bridge the gap between data analysis and real-world healthcare decision-making. For the few studies that exist which do contain recommendations for the Mexican Ministry of Health and healthcare providers like performed in Martínez-Martínez (2022), comparison can be made between results and concluding policy recommendations.

Like much of the other research, the previous comprehensive study on the dataset (Almustafa, 2022) fell short in addressing the translation of these predictions into actionable healthcare strategies. The research largely focused on achieving high predictive accuracy without delving into the implications of these predictions for healthcare policy, patient prioritization, or resource allocation. There are a handful of other studies in existence also analyzing similar data (COVID-19 patient data in Mexico) using different datasets. For example, Parra-Bracamonte et al., developed a comprehensive study using logistic regression from data released in 2020 to determine what variables were most correlated with mortality (2020). This along with the body of other relevant literature will be critical in comparing results and recommending resultant policy.

The last significant gap in previous research is the lack of focus on data integration techniques. While many studies have utilized individual datasets to conduct EDA and build predictive models, the do not address the complexities and challenges involved in combining multiple sources of data. This project will fill that gap by demonstrating robust data integration methods to merge multiple large-scale COVID-19 datasets. Through advanced data engineering processes, such as aligning different variables, handling discrepancies, and ensuring data consistency, the project will create a unified dataset that provides a more comprehensive view of patient outcomes and health conditions, enabling deeper insights that are not achievable with isolated datasets.

Proposed Project

This project was selected because COVID-19 has had a significant impact on public health globally, creating a pressing need for data-driven insights to better understand the factors influencing patient outcomes (Berlin et al., 2020). The 2020 pandemic quickly exposed the vulnerability of healthcare systems and the inadequacy of existing protocols to handle large-scale emergencies. Policymakers and healthcare workers gave their best efforts to contain the outbreak of COVID-19; yet, it quickly escalated from a regional, to national, to global crisis, culminating in over 14 million documented positive cases and more than 597,000 deaths worldwide (WHO, 2020). While the quantitative toll of COVID-19 is undeniable and felt by all, its qualitative impact is also profound - supply chains faltered, economies stalled, and daily life was profoundly disrupted. My personal motivation for everything I accomplish professionally is to the benefit of patients' well-being and outcomes, driving me toward projects where I can make a tangible difference. This passion for patient-centered outcomes has guided my career into both research and the manufacturing of high-fidelity pharmaceuticals, particularly in times when rapid and effective solutions are most critical. The availability of extensive patient data related to COVID-19 provides a unique opportunity to apply advanced data engineering and predictive modeling techniques, allowing for the identification of critical patterns that can improve healthcare responses during future crises.

The importance of this project lies in its potential to provide actionable insights that can directly shape the development of targeted interventions, optimize resource allocation, and refine patient care strategies. The healthcare community learned that

during emergencies, quick and precise decisions are necessary to save lives, especially when resources are limited. It is too costly and inefficient to rely on educated guesses for triage in the middle of a high-stress disaster when data-backed strategies can guide decisions with greater accuracy. By analyzing patient outcomes and identifying variables that contribute to severe cases, this project aims to equip policymakers with the knowledge they need to implement preventive measures and to refine healthcare protocols. For example, understanding how comorbidities like hypertension or diabetes impact COVID-19 outcomes can guide vaccination or treatment priorities. Moreover, this knowledge can serve as a basis for building more resilient healthcare strategies, ensuring that hospitals are better prepared for surges in demand. Armed with these insights, healthcare providers can shift from reactive to proactive care models, leading to more effective responses not only in future pandemics but in other public health emergencies as well.

Key Performance Indicators

Key Performance Indicators (KPIs) are essential for evaluating the success and effectiveness of any data analytics project. In this healthcare analytics project, the KPIs are designed to ensure that the predictive models developed are both accurate and reliable in forecasting critical patient outcomes such as mortality and ICU admissions. These indicators not only help gauge the performance of the models but also ensure that the data integration and feature engineering processes contribute meaningfully to the predictions. Given the importance of patient outcomes, achieving high accuracy and sensitivity is crucial to minimize errors, particularly false negatives, which could lead to delayed or inadequate treatment. By setting specific benchmarks for each KPI, the project can

systematically track progress and adjust as needed to achieve the desired performance levels.

1. **Model Accuracy ≥ 90%** - Achieve an overall model accuracy greater than 90% in predicting patient outcomes in mortality (primary focus).
2. **Model Sensitivity (Recall) ≥ 90%** - Ensure that the model correctly identifies at least 90% of positive cases (mortality) reducing the likelihood of false negatives. This is especially important since the datasets are imbalanced with a ratio near 10:1.
3. **F1 Score ≥ 90%** - Attain an F1 score greater than 90%, balancing precision and recall to ensure robust model performance in predicting mortality.
4. **Data Integration Completeness ≥ 95%** - Successfully integrate the 2 datasets with at least 95% completeness (measured by the percentage of non-missing records across the combined datasets).
5. **Feature Importance Contribution ≥ 80%** - Ensure that the top 5 features explain more than 80% of the variance in the prediction outcome, showing that key factors are effectively contributing to model predictions.
6. **Model Accuracy, Sensitivity, and F1 Score ≥ 90%** - Achieve an overall model accuracy, sensitivity, and F1 score of equal to or greater than 90% in predicting patient admission to the ICU (secondary focus).

KPI 1: Achieving an accuracy rate above 90% is crucial to ensure the reliability of predictions made by the model. Accuracy is the fraction of correctly predicted cases (Erickson et al., 2021). In the context of predicting patient outcomes, high accuracy means that the model can correctly classify the majority of cases, leading to greater trust in its outputs. This level of accuracy is particularly important when integrating findings

into clinical decision-making, where incorrect predictions could impact patient care strategies leading ultimately to life or death. However, while accuracy is a fundamental metric, it is not sufficient on its own, especially in cases of imbalanced data. A high accuracy could still result in a model that overlooks many true positive cases if the model tends to predict the majority class. Therefore, maintaining a balance between accuracy and other metrics like sensitivity (recall) is critical to ensure the model performs effectively across different classes. Thus, KPI1 acts as a foundation upon which other measures of model performance are built.

KPI 2: High sensitivity is vital in the context of predicting mortality because it directly addresses the model's ability to identify true positive cases. In healthcare analytics, a false negative - where a high-risk patient is incorrectly classified as low-risk - can lead to a delay in necessary interventions, potentially worsening patient outcomes (Erickson et al., 2021). Therefore, ensuring that the model captures at least 90% of the true positive cases minimizes these risks, making the predictions more actionable in clinical settings. Since the dataset is imbalanced with a low occurrence of mortality compared to survival, focusing on recall helps to ensure that these rarer, but critical, cases are not overlooked. A high sensitivity threshold also highlights the model's robustness in identifying patients who may otherwise be missed. This metric ensures that the model remains sensitive to the high stakes associated with patient outcomes, providing a safeguard against under-detection of critical cases.

KPI 3: The F1 score balances both precision and sensitivity, making it a comprehensive measure of the model's performance, especially when dealing with imbalanced data (Erickson et al., 2021). A high F1 score above 90% indicates that the

model is not only effective at identifying positive cases but also maintains a low rate of false positives, ensuring overall robustness. In a healthcare context, this means that the model can accurately differentiate between patients who are truly at risk and those who are not, reducing unnecessary interventions for those who are falsely predicted to be high-risk. While the cost of a false negative is far higher (meaning the patient is likely to die), the cost of a false positive is also significant towards the healthcare system and towards the patient (as they would be told they are high-risk and would need significant intervention). The F1 score becomes particularly valuable when there is a trade-off between sensitivity and precision, as it ensures that neither aspect is sacrificed for the other. By focusing on this metric, the analysis aims to deliver a model that is balanced and reliable, providing stakeholders with confidence in its predictions. This KPI ensures that the predictive model is suitable for practical use, where both missing critical cases and over-alerting can have significant consequences.

KPI 4: The success of the project heavily relies on the seamless integration of the Covid1 and Covid2 datasets, aiming for a completeness level of over 95%. This metric ensures that the datasets are merged with minimal loss of data, thereby preserving the richness of information across both datasets. High completeness directly impacts the model's training, as missing data can skew results or reduce the diversity of cases the model is exposed to, which could hinder generalization. By achieving this KPI, the analysis ensures that valuable records are retained, leading to a more comprehensive understanding of the data landscape and patient trends. Additionally, a well-integrated dataset enables more effective feature engineering, as relationships across datasets can be explored without the complications introduced by missing entries. It also facilitates a

more accurate comparison of variables that are shared between the two datasets. Ultimately, this KPI guarantees that the data used in modeling is as representative and reliable as possible, strengthening the overall validity of the analysis.

KPI 5: Focusing on the contribution of the top 5 features to explain more than 80% of the variance is critical for understanding which variables are driving the model's predictions. This KPI allows the analysis to prioritize the most influential predictors, which not only aids in model interpretability but also helps in streamlining data collection in future studies. In the context of predicting outcomes like mortality, knowing which features most heavily influence predictions (e.g., age, comorbidities, ICU admission) can inform clinical focus areas and resource allocation. By concentrating on a few high-impact variables, the model becomes more actionable, offering healthcare providers clearer insights into the key factors that contribute to severe outcomes. This also means that the model is less likely to rely on spurious correlations or noise, increasing the robustness of its predictions. This KPI will feed directly into generating the risk-matrix as well. Achieving this KPI helps ensure that the analysis is not only predictive but also offers meaningful insights into patient health, thus bridging the gap between analytics and real-world application.

KPI 6: Achieving these performance metrics for predicting ICU admission ensures that the model is robust and reliable across each different aspects of performance as previously discussed. This will provide a comprehensive view of the model’s strengths and weaknesses. In the context of ICU admission, high accuracy ensures that the majority of predictions are correct, while high sensitivity ensures that those who truly need

intensive care are identified. An F1 score greater than 90% indicates a balanced approach between precision and recall, ensuring that both false positives and false negatives are minimized. This is crucial for ICU predictions where the cost of a false negative (missed ICU admission) could lead to severe outcomes for patients. Moreover, ensuring that the model performs consistently well across all three metrics helps in making the model suitable for practical use, such as in assisting triage decisions during surges in COVID-19 cases. This KPI reflects the goal of creating a predictive tool that is not only theoretically sound but also practically applicable in high-stakes environments like healthcare.

The established KPIs will serve as benchmarks to ensure that the project meets its objectives and delivers actionable insights. By achieving high model performance metrics and maintaining data integrity, these KPIs will confirm the robustness and reliability of the models in predicting patient outcomes. Ultimately, they will demonstrate the project's success in providing valuable data-driven insights to healthcare providers and policymakers, enabling more informed decision-making and better patient management strategies.

Project Insights of your Data Analysis

From the analysis, I expect to build a risk-matrix: a set of key factors that are strongly correlated with severe COVID-19 outcomes, such as ICU admission and mortality. These factors will include comorbidities like cardiovascular disease, diabetes, renal chronic issues, and obesity, along with demographic variables such as age, sex, and pregnancy status. This matrix should include whatever variables most strongly affect the negative patient outcomes. In addition, the matrix should be able to add quantifiable risk scores to those top drivers thus giving real numbers to what conditions are most likely to

put patients at risk of severe outcomes. By integrating multiple datasets, I aim to build a more comprehensive understanding of patient profiles, thereby revealing insights that go beyond isolated variables. For example, it is likely that interactions between certain pre-existing conditions and patient demographics – such as the combination of hypertension and advanced age – will be more predictive of severe outcomes than any single variable alone. I expect that I will build a quantitative approach to understanding which variables most heavily affect negative patient outcomes.

Once that risk-matrix is built, I expect that it will be used to make general process and policy recommendations. For example, healthcare providers and administrators should be able to take it to build a risk profile for each patient to understand how risk they carry for negative COVID-19 outcomes. This will therefore help guide operational process flow and triaging patients into different risk categories (high, medium, low for example). Likewise, the same risk-matrix should be helpful to policymakers to build generalizable strategies which are effective across multiple regions of the same country (or even globally). While each region and country will need some tailoring, the primary objective health outcomes should be generalizable across populations.

The integrated analysis is expected to reveal disparities in patient outcomes based on sociodemographic factors, medical history, and health behavior patterns, such as tobacco use. I anticipate the results will show that specific patient subgroups are disproportionately affected by severe COVID-19 complications. For instance, patients with immunosuppressive conditions or multiple comorbidities will exhibit a higher likelihood of adverse outcomes, which could be quantitatively captured through

predictive modeling. Such findings would highlight the need for tailored healthcare strategies, especially for at-risk populations. In addition, this project may also identify gaps in current treatment protocols or resource allocation that, if addressed, could significantly improve patient care and reduce strain on healthcare systems during future pandemics.

I expect that the project will achieve most, if not all, of the defined KPIs and be executed on time, adhering closely to the milestones and class schedule. This involves not only meeting deadlines but also ensuring that each phase, from data preparation to model deployment, is completed with high quality and precision. If the project does not meet its KPIs or milestones, I expect to be able to explain why that is – whether due to failure in analysis, need for more resources, difficulty with the data, etc. I anticipate showcasing efficient data engineering techniques, particularly in the integration of the Covid1 and Covid2 datasets, which involves overcoming challenges like aligning shared variables, handling potential inconsistencies, and standardizing data formats. These steps are critical in ensuring that the combined dataset is robust and ready for subsequent analysis, minimizing any risks of data quality issues that could skew the findings.

Moreover, I expect to demonstrate advanced exploratory data analysis (EDA) and modeling techniques, which will help uncover meaningful patterns within the data. This includes using visualization tools like Python and Tableau to better understand the relationships between variables and applying machine learning algorithms to identify trends that might not be immediately apparent. For instance, I aim to explore feature importance and model explainability helping to clarify which factors most influence

patient outcomes. My analysis will strive to surpass the existing body of literature on COVID-19 patient data by leveraging a significantly larger number of records, enabling a more comprehensive view of patient demographics and outcomes. Additionally, I plan to develop a model that not only achieves higher predictive accuracy but also is more interpretable and applicable in real-world settings. This could involve assessing the model's generalizability across different patient groups and ensuring that the findings can be translated into actionable insights for public health strategies. Ultimately, I aim for the project to serve as a valuable reference for future research by providing a detailed methodology and insights that could inform better clinical decision-making during similar health crises.

Project Milestones

Milestone 1: Data Review and Initial EDA

- Gather and review the two COVID-19 datasets to understand their structure and characteristics.
- Perform an initial exploratory data analysis (EDA) to identify potential inconsistencies, missing values, and anomalies.

Milestone 2: Data Integration and Cleaning

- Merge the datasets using appropriate data integration techniques.
- Perform data cleaning to address missing values, standardize date formats, address outliers, and ensure consistent labeling of categorical variables. Eliminate columns unique only to one dataset or evaluate potential usage of them.
- Achieve KPI 4.

Milestone 3: Feature Engineering and Selection

- Create new features such as patient risk scores, age brackets, etc.
- Select the most relevant features using correlation analysis and feature importance evaluation.

Milestone 4: Final Integrated Dataset EDA

- Perform a final EDA of the integrated dataset in order to show trends, class balances, and understand the prepared data before modeling.

Milestone 5: Predictive Model Development and Training

- Develop multiple predictive models (e.g. Random Forest, Logistic Regression, Gradient Boosting, and Support Vector Machine) to predict mortality outcomes.
- Optimize model parameters using techniques such as cross-validation.

Milestone 6: Model Evaluation and Validation

- Evaluate model performance using accuracy, sensitivity, F1 score, and ROC-AUC thus achieving KPI 1-3, and 6.
- Validate the model with a hold-out test set or through cross-validation to ensure robustness and generalizability.

Milestone 7: Data Visualization and Insights Generation

- Create insightful visualizations such as heatmaps, distribution plots, and correlation matrices.
- Achieve KPI 5.
- Present key findings and actionable insights through clear and concise visual summaries.

Milestone 8: Final Report and Presentation

- This milestone will occur concurrently with all other steps as outline via the DATA 670 classrooms due dates
- Compile a comprehensive report detailing the project's scope, methodology, results, and conclusions.
- Prepare several presentations to communicate the project scope, methods, findings, and recommendations to stakeholders.

Completion History

Week 1	I completed searching through many datasets attempting to find the right project mainly focusing on topics of chemistry, biology, and healthcare. I downloaded several potential datasets but most did not fit the course requirements.
Week 2	I completed finding 2 datasets pertaining to COVID-19 provided by the Ministry of Health in Mexico. I created the project charter describing from a high-level point of view the scope of this project including information on the business background, the datasets to be analyzed, KPIs, milestones, and critical success factors.

Week 3	I converted the information from the project charter into a full-text document (this report). I adapted the Project Scope document from Assignment 1 to this document adjusting each section to be the appropriate level of length/detail. I performed EDA on the datasets to fully understand and describe them before processing.
Week 4	I refined the full report further adding deeper research from relevant sources (in the field of healthcare analytics and public health) and further framed the business background more robustly. I created a graphical representation of the datasets I used and how they related to one another. I created 2 bar chart matrices, counts, and histograms using Python to show variable distribution in each dataset.
Week 5	I created a 10-minute presentation which was communicated live to my group of 3 other students. The PPT had 17 slides containing nearly all topics discussed in the written assignment thus far. The audience then had 10 minutes with which they provided feedback. I recorded this to be able to adjust the presentation as this project develops further. I made a written report documenting the feedback I also gave to my cohort for each of their presentations.
Week 6	I wrote the Python code to clean and integrate the datasets. I troubleshot many issues along the way until the code was fully functional. I achieved KPI 4.
Week 7	I used the Python code to author the report required for this week included the approach towards data preparation, cleaning, transformation, and the analysis which would follow.
Week 8	I created 4 visualizations from the final dataframe before analysis in order to highlight trends found within the data. The models will help confirm these trends (such as those seen in the correlation matrices).
Week 9	I wrote code to fix the data integration error and for the models.
Week 10	I continued to write code for the models continuously debugging errors. I summarized their results into tables with heatmap visual coding and their parameters.
Week 11	I wrote the last pieces of code to interpret the best mortality model. I also made the final presentation of the project to summarize the project, results, and outcomes, as well as writing the rest of the final report.

Lessons Learned

Week 1	I learned that the original field I wanted to perform this project in did not have nearly robust enough datasets in order to fulfil this course's requirements. I learned many datasets don't fit the parameters set out of having many variables (>20) and many tuples (>100,000).

Week 2	I learned that the field which I would have the most interest in with the highest likelihood of high-quality datasets would be in the field of COVID-19 analytics. I learned about the two datasets I eventually chose which fulfilled the course requirements and would be interesting to work with (while not presenting an overwhelming challenge not feasible within the timeframe of this course).
Week 3	I learned from feedback from the professor that scope of my project was acceptable. Furthermore, the primary area of improvement from the original project charter was to demonstrate further background on the business metrics that made this project necessary.
Week 4	I learned from feedback that the wording of the scope pertaining to the Business Objectives needed to be tailored. I learned ways I can effectively demonstrate the two datasets and their relationship using a Venn diagram.
Week 5	I learned from feedback my presentation excellently captured the story of my analysis thus far. Areas for improvement included explaining my target variables more thoroughly and simplifying the slides to only contain the necessary information.
Week 6	I learned while coding that 1. it won't be worth keeping the unique variables for the long-term and 2. while there appear to be a handful of potential duplicates in the data, they must be kept in as there is no way of detecting which entries are true duplicates and which are not.
Week 7	I learned the best path forward in utilizing visualizations to tell the story of my analysis as well as the modeling.
Week 8	I learned what the most valuable variables are to each target variable (described in this report) based upon Chi-Square analysis and correlation matrices. These should be later confirmed by modeling.
Week 9	I learned there was an error in the dataset integration where the COVID-19 testing variable was previously merged without any adjustment. This was incorrect as values meant different outcomes in Covid1 and Covid2. This was corrected and resultant visualizations were updated.
Week 10	I learned achieving all KPIs for this analysis will not be possible as no models achieved an F1 score >54%. F1 is heavily impacted by class imbalance. F1 is strongly affected by techniques like class weights and K-fold cross validation since they keep the imbalance in the dataset (unlike oversampling).
Week 11	I learned how to build a framework for risk matrixing. I learned that despite the chi square analysis showing somewhat equal values between the top 3 variables, the feature importance analysis showed an overwhelmingly primary dependency on ICU as a variable.
Week 12	I learned that the primary drivers of mortality and ICU admission were mostly consistent across feature importance of the champion models and

	the chi-square analysis. They also are consistent with current research and intuitive logic.

Data Set Description

The project will utilize two datasets, each containing extensive information on COVID-19 patient demographics, health conditions, and medical treatment details found on Kaggle. The data from both sources were extracted from the Ministry of Health of Mexico. The first dataset contains approximately 566,603 records and includes variables 23 such as patient demographics, pre-existing health conditions, dates of symptom onset and hospital entry, and critical outcomes like ICU admission and mortality (Mukherjee, 2020). This dataset provides a comprehensive view of patient characteristics and health histories. Thus, it is ideal for exploring how different factors contribute to the severity of COVID-19 cases. Some of the key features include patient age, gender, and presence of health conditions like diabetes, hypertension, and cardiovascular issues (all variables are categorical with the exception of age, ID, and 3 date variables).

The second dataset is even more extensive, with 1,048,567 records but only 21 variables (Nizri, 2023). This dataset also includes variables such as medical unit assignment, patient type (e.g., outpatient or hospitalized), and the final classification of the patient's COVID-19 condition. It also provides a detailed view of various comorbidities like asthma, obesity, and renal chronic disease, along with patient behaviors such as tobacco use. In contrast to the previous dataset, it does not contain a patient ID, date of entry, date of symptom onset, having contact with another patient infected with COVID-19, nor testing result (positive/negative). However, it does contain two unique fields: a classification of testing of presence of COVID-19 (on a scale) and how many medical units were used to treat patients. Thus, these two datasets have

enough similarity that they should be able to be combined to be used for the same type of predictive modeling towards the same target variable.

While there is significant overlap between the two datasets in terms of health condition indicators and patient demographics, they differ in their focus and size. The first dataset emphasizes individual patient outcomes and treatment processes, whereas the second dataset offers more granular hospital-level data and classification details, and it has many more entries. Integrating these two datasets will create a unified view that encompasses both patient-level and hospital-level factors, enabling a more holistic analysis of how various factors impact patient outcomes during the COVID-19 pandemic. An overview of all variables of each dataset, and how the variables overlap is observed in Table 1. This overview gives a visual reference for what each variable indicates and for potential data quality issues.

The process of integration will entail joining the two tables based upon the unified variables observed in both tables as seen in Table 2. Of the 25 variables, 19 are found in both tables leaving 6 variables only found in 1 table or the other (4 unique to dataset 1 and 2 unique to dataset 2). Because these are both extracted from the same source (the Ministry of Health of Mexico) and are several years apart (with the larger dataset being posted later), it needs to be understood if there are potential identical entries in the datasets. If there are identical entries, then those tuples will be highlighted such that duplicates can be removed. All subsequent further analysis will then depend on analyzing data from dataset 1, dataset 2, or both (for example, modeling could be conducted using date of entry and date of first symptoms from dataset 1; or how many medical units were

used to treat patients from dataset 2). If there are no overlapping entries between the datasets, then can be combined directly without concern of removing duplicates. From henceforth, the datasets will be referred to as “Covid1” and “Covid2” respectively for ease of discussion.

High-Level Data Diagram

A Venn diagram was generated to present the relationship between two datasets (Figure 6). Again, the Covid1 dataset comprises 23 variables and includes 566,603 records, while the Covid2 dataset consists of 21 variables with a larger scope of 1,048,567 records. Despite some differences, the datasets share a significant overlap, with 19 common variables. These shared variables include essential information like patient demographics (e.g., age and sex), medical conditions (e.g., diabetes, hypertension, asthma), and clinical outcomes (e.g., ICU admission, intubation status, and COVID-19 test results). However, each dataset also includes unique variables that contribute to their distinct focus. The Covid1 dataset has the specific variables of an anonymized patient ID, patient entry date, date of symptom onset, and details about contact with other COVID cases, emphasizing individual patient timelines and exposure risks. In contrast, Covid2 includes variables such as USMER and Medical Unit, which are more focused on the hospital or healthcare facility context.

Each dataset has its strengths and limitations. The Covid1 dataset has its emphasis on individual patient details like the onset of symptoms and contact tracing. It offers insights into the timelines and progression of the disease at the individual level. However, its smaller sample size limits the generalizability of its findings. Meanwhile, Covid2 has a larger number of records making it better suited for statistical analyses/modeling. The

downside is that its lack of certain individual-level data, such as symptom onset dates, could make it less precise when tracking disease progression. By integrating these datasets, analysts can capitalize on their respective strengths. The shared variables enable seamless merging, providing a comprehensive dataset that combines the breadth of Covid2 with the depth of detailed patient-level data from Covid1. This integration should lead to a more holistic understanding of COVID-19 outcomes across both individual and institutional contexts. However, merging the data will introduce challenges, such as aligning the timing-related variables or managing the differences in the data collection methods between datasets. However, these will be addressed during the next stage of analysis once it is discovered if there are repeated entries or not. It is possible to use Covid1, Covid2, and the integrated dataset in modeling in order to demonstrate the differences between analyses using each of the datasets.

Data Definition/Data Profile

A table was created in order to quickly understand the logistics of the two datasets and how they will be integrated (Table 1). Where dataset 1 has 23 variables, and dataset 2 has 21 variables, combined they will have 25 variables due to overlap. It is critical to highlight that of the 25 total variables 10 variables have no missing values. The other 15 variables have missing values denoted in the form of “98/99” (not specified), and “97” (not applicable). In the case of “97” values, examples include ICU (where a patient who was outpatient would not have been applicable to being in the ICU) or pregnant (where a male patient could not be pregnant). All dates fall within expected ranges (within Jan to Dec of 2020). The patient ID variable will be discarded as it offers no benefit to analysis. In the age variable, there are a total of 85 and 138 tuples with age value >100. In a

population of >500,000 and <1,500,000, having values above 100 years old are possible. But with values like 120 years, these are not likely valid since the oldest person alive now is 111 years old (Gerontology Research Group, n.d.). There is no way to check if these values are accurate and thus it will be assumed all age values above 100 are inaccurate and will be removed.

Two bar chart matrices for the Covid1 and Covid2 datasets were generated in Python in order to understand the distributions that both datasets have separately (Figure 7, Figure 8). These reveal several important trends to understand about both datasets before moving on with further analysis. Both datasets show that most patients did not suffer from severe conditions like Pneumonia, Diabetes, or likewise have severe negative COVID-19 outcomes like requiring ICU admission or dying. This suggests that while COVID-19 can be life-threatening, many cases are not associated with these critical conditions. However, the minority of patients who did experience these conditions likely required significant medical intervention, making these variables key predictors for poor outcomes. The relatively low numbers for severe comorbidities such as COPD, Cardiovascular Disease, and Renal Chronic Disease across both datasets likely will have the outcome that all of these conditions are causal factors in COVID-19 severe outcomes.

There are several variables where a “97” value stands out in both datasets like Pregnant, Intubated, and ICU. This is because there are many patients who these conditions are not applicable to like males not being pregnant, and outpatients not being in the ICU or intubated. This suggests a clear differentiation between the majority of patients, who likely managed with less invasive treatments being outpatient, and a

smaller subset who required more aggressive respiratory support being inpatient. Additionally, the skewed Age distribution in both datasets points to a higher number of younger patients, although older patients, who likely experienced more severe outcomes, represent a significant tail in the distribution. This will be explored further in future steps in the project like generating an age distribution for those who died and those who needed ICU intervention. Understanding the relationship between age and other clinical variables such as comorbidities and ICU admission could yield deeper insights into the factors driving severe COVID-19 outcomes.

Furthermore, in the Covid2 dataset, the DATE_DIED distribution reveals a clear elevated concentration of deaths in 2020 with a tapering off in 2021. This potentially corresponds with a major wave of the pandemic earlier in 2020 with decreases happening due to awareness and vaccination. This temporal pattern is not seen in the Covid1 data because it only spans within 2020. Finally, it is key to note that the vast majority of missing values are relatively low. Many of the “97” values can be kept in and will be treated as a “no” in cases where it indicates “not applicable”. Therefore, in the next section it will be explored to determine the impact of removing all missing value tuples from the dataset or if another method is needed to address the missing values. On rough approximation, if the missing value tuples add up to around 50,000 per dataset, then it will likely be appropriate to remove them as this is around 1% of the overall dataset. This was tested by writing some Python script and it was determined the total counts of tuples with missing values in each in Covid1 and Covid2 were respectively 5,506 and 28,909 (Figure 11). Thus, the total number of tuples with any missing values in them are 34,415. It's important to note that many of the records which contain missing values, are missing

multiple values from different variables. In addition, there are no traditional outliers in this dataset which need to be addressed as most of the variables are categorical or dates. While some of the categorical variables are imbalanced, none of them contain traditional outliers. The dates likewise don't contain any outliers. The age variable, as was previously discussed, contained a handful of values which may appear as outliers and will be removed.

Two histograms were made to depict the age distribution of deceased patients in the Covid1 and Covid2 datasets, using 10-year age bins. Both distributions reveal a similar pattern, with the highest concentration of deaths occurring in patients aged between 30 to 50 years, followed by a gradual decline in frequencies as age increases beyond 60. In both datasets, there is a significant number of deceased patients in the younger age brackets (10-40 years), indicating that COVID-19 affected a substantial proportion of younger populations as well. The frequency of deceased patients diminishes beyond the 60-year mark, though there are still notable counts up to age 100, reflecting the broad vulnerability range of the virus. Additionally, the Covid2 dataset shows a higher frequency across most age bins compared to the Covid1 dataset reflecting how the Covid2 dataset has a larger sample size. Since the shape of the histograms are similar in both cases, and similar to the overall population histograms previously observed, we conclude that these datasets represent deaths across all age groups and that age is likely less of a contributing factor than some of the other factors to be studied.

Addressing the imbalance in target variables, such as mortality and ICU admission, is crucial for the accuracy and reliability of the predictive models. The

datasets present a significant disparity, with a relatively small number of ICU patients, 26,970 in total, compared to the much larger number of deceased patients, with 113,111 total deaths (36,177 in Covid1 and 76,934 in Covid2). Traditional methods like oversampling and undersampling could be considered to balance these classes, but they are not ideal in this scenario due to the substantial gap between the class sizes, which could lead to overfitting or loss of important information (Amin et al., 2016). Instead, more advanced techniques will be implemented to handle this imbalance effectively. Applying class weights during model training allows the models to give more importance to the minority classes without artificially altering the data distribution. Additionally, using k-fold cross-validation ensures that each subset of the data is well-represented in training and testing, providing a robust way to evaluate the model's performance across different segments of the data (Amin et al., 2016). These methods will help ensure that the models are sensitive to identifying severe cases, such as those requiring ICU care or having higher mortality risks, despite the imbalance in the target variables.

All variables in the integrated dataset will be considered appropriate for modeling, with the exception of the ID variable from Covid1, which will be dropped as it serves no predictive purpose. The integration process will retain all variables, including those that are exclusive to either the Covid1 or Covid2 dataset, ensuring that unique information from both sources is preserved. This approach allows for a broader initial analysis where insights can be drawn from the entire spectrum of available data. However, to enhance the robustness of the modeling phase, a second version of the integrated dataset will be created that focuses solely on the 19 common variables between the two datasets. In this version, the six unique variables (unique to only Covid1 or Covid2) will be removed to

ensure consistency and reduce noise in the data, facilitating more high-fidelity predictive modeling. This dual-dataset strategy enables flexibility in exploring the broader dataset for potential insights while ensuring that the core modeling efforts are based on a standardized variable set, thus balancing comprehensiveness and precision. Finally, it's critical to note that there are no biases or critical outliers not already addressed in the dataset. The datasets have their limitations in that they don't contain more granular data. For example, it's assumed that the trends observed in these datasets will be applicable to all demographics and ethnic groups. However, this must be cross-validated to ensure that this assumption is correct. It is also limited where the COVID-19 vaccine has been out for more than 2 years now. Data should also be analyzed with this variable to understand how vaccination changes outcomes. However, those results should be compared with this analysis to ensure the outcomes are still in agreement with one another.

Data Preparation/Cleansing/Transformation

Data Preparation

The process of preparing the data will involve a series of methodical steps to ensure that it is clean, consistent, and ready for analysis. Ultimately, the goal of preparation is to prepare the data for modeling. Therefore, the resultant dataframe will be clean, consistent, and fully available for the model. The initial phase will include loading the datasets into Python, specifically within the Spyder 3.3.6 environment. It is an integrated development environment (IDE) ideal for data exploration and manipulation and is a longstanding personal favorite IDE (Figure *12*). Data will be read using Pandas, which offers powerful functions for data ingestion, allowing for efficient loading of large

datasets. Variable alignment will be performed first, ensuring that column names between the Covid1 and Covid2 datasets are standardized. This step is crucial to ensure consistency when merging, as inconsistent variable names can cause errors or mismatches. Once columns are aligned, merging the datasets into a unified dataframe will follow, using Pandas' concat function.

Once the data has been integrated, quality control checks will be conducted to validate the integrity of the merging process. This is a pivotal step in data engineering to ensure data integrity before moving forward. This will involve printing the shape of the dataframe and examining a sample of the data to confirm that all expected columns and rows are present. Duplicate entries will be identified using the duplicated() method to assess the extent of redundancy in the dataset. While duplicates will be flagged, they will need to be investigated to determine the correct path forward prior to analysis. Usually, duplicates need to be removed but the duplicates may be retained to analyze any potential trends or patterns in repeated data entries. Columns that are not needed, such as unique identifiers or unrelated variables, will be removed using the drop() function in Pandas to create a cleaner and more focused dataset. These removals will help streamline the dataset, making subsequent analysis more efficient and reducing noise that could affect modeling outcomes.

Following structural cleaning, data preparation will continue with handling missing values and standardizing variable formats. Specific codes like 99 and 98, which are used in this dataset to denote missing data, will be targeted and removed. Python functions will also be used to replace non-standard categorical values, such as converting

placeholder values of 2 to the more conventional 0 across categorical variables. Finally, age data will be reviewed, and any records with values over 100 will be removed to exclude unrealistic outliers that could skew the analysis. The overall preparation process, leveraging Python's Pandas and other libraries within the Spyder environment, will ensure that the data is clean, standardized, and ready for robust exploratory data analysis and model development. As a last note, some previous sections in this report were updated based upon feedback received in order to make this analysis more thorough.

Feature Examination

The selection and justification of features are crucial in building predictive models and meaningful visualizations. Each feature used should have a clear rationale based on its predictive power and interpretability. For example, the AGE_GROUP variable was specifically engineered to replace the continuous AGE variable, allowing for better segmentation of patient data into discrete categories. This choice helps capture non-linear relationships between age and outcomes, facilitating both interpretability and visualization. Categorical variables, such as AGE_GROUP, can make trends more apparent when plotting data, aiding in the identification of risk profiles within specific age brackets. Similarly, the COMORBIDITY_COUNT variable aggregates binary indicators for chronic conditions into a single metric, which simplifies the assessment of overall patient risk. This variable is not only helpful in predictive modeling by providing a straightforward measure of health complexity, but it is also valuable in visualizations, as it allows stakeholders to quickly see how the number of comorbidities correlates with different outcomes. The inclusion of comorbidities such as DIABETES, COPD, CARDIOVASCULAR, and all other chronic conditions is justified by their well-

documented impact on patient outcomes and disease severity. These variables provide critical insights into individual patient risk profiles, enabling predictive models to account for the cumulative effect of health conditions on outcomes. Hospital data, such as INTUBED, ICU, and OUTPATIENT status, are essential for understanding the level of care and severity of cases, making them highly relevant for predicting patient trajectories and visualizing treatment patterns and healthcare resource utilization. In total, all variables left in the dataset are critical to the analysis as well and visualization and will be kept as part of the analysis.

A well-explained target variable is essential for setting clear expectations and maintaining focus throughout the analysis. In this context, it's critical to detail what the target represents and why it is chosen over others. The selected targets, ICU and mortality, were chosen because they represent severe endpoints that are essential for healthcare decision-making and resource allocation. Fully understanding and defining these variables ensures that the predictive models are aligned with real-world healthcare priorities, such as identifying high-risk patients who may require intensive care or are at higher risk of mortality. ICU admission and mortality provide meaningful, actionable outputs that guide clinical practices and policy decisions. Ensuring that these target variables are accurately captured and consistently defined is crucial for developing models that are both reliable and effective in predicting and visualizing severe outcomes. They have both already been discussed in-depth, however as a new note, ICU will be used as a predictive variable for mortality but not vice versa. This is because ICU can predate mortality but mortality cannot predate ICU admission.

An essential aspect of modeling is determining whether the target variable is imbalanced, as this can significantly affect model training and predictions. An imbalanced target, where one class greatly outnumbers others, may cause models to become biased, predicting the majority class more frequently and underrepresenting the minority class. As the target variables in this dataset show signs of imbalance (more imbalance in ICU), strategies will be implemented to address it. One approach is to apply weighted loss functions that give more importance to underrepresented classes during training, ensuring the model doesn't overlook critical cases. Additionally, k-fold cross-validation can be used to provide a robust evaluation of the model by training and testing on multiple subsets of the data (Amin et al., 2016). This technique ensures that the model is exposed to diverse data distributions and reduces variance in performance metrics. While oversampling/undersampling are also viable options, they would severely limit the dataset especially when exploring the ICU variable (since the ratio of positive cases are much smaller than Mortality). The goal of the analysis should be to increase information, whereas these methods could cause the analysis to lose information from the dataset. Thus, instead of these method, weighed loss and/or k-fold cross validation will be utilized instead. By outlining these methods, the approach to handling class imbalance is justified and explained, emphasizing the goal of creating a balanced, fair, and reliable predictive model.

Data Cleansing

The initial phase of the data cleaning process focused on aligning the variable names between Covid1 and Covid2. This step was essential due to inconsistencies in how column names were structured in each dataset. By standardizing the variable names, data

integration became a seamless task, reducing the risk of errors during merging and analysis. The names chosen were previously discussed and again can be found in Table 2. This foundational step set the stage for integrating the datasets into a unified dataframe.

Following variable alignment, the integration of the two datasets was performed. This integration combined all records while maintaining data integrity and handling shared and unique data points across the two sources. A comprehensive integration process required checks to ensure that no data was lost and that both common and unique columns were appropriately managed. Once merged, the integrated dataset was subject to thorough data engineering quality control checks. These checks included verifying row and column counts, confirming data type consistency, and ensuring that variable alignment persisted post-integration. This stage was crucial for maintaining data reliability, as any oversight at this point could cascade into issues during analysis or modeling phases. Checks included: 1. Observing the total number of tuples in each of the original 2 datasets versus the integrated dataset (Covid1 = 566,602, Covid2 = 1,048,575, integrated = 1,615,177); 2. Checking the number of variables in the integrated dataset (25); 3. Using an if/else statement to check if the expected number of rows were correct and if so issuing a written statement (and if not as well); 4. Displaying all of the column names in the integrated dataset; 5. Showing all of the data types of the columns (Figure *13*). No issues were observed across all five checks and thus cleaning could progress to the next stage.

After ensuring that the integration was successful and consistent, a thorough review for duplicate rows was conducted. Identifying duplicates helped highlight

redundant data that could skew results or add noise to predictive models. Interestingly, while duplicates were flagged, they were intentionally retained for further examination and potential analysis with the reasoning described below. First, the duplicate function was used to detect potential duplicates in the integrated dataset with no edits which showed 812,049 values (Figure *14*). This seemed unreasonably high since this would indicate that if there were duplicates, it would be more than the entire number of Covid1 plus 300,000 entries. To investigate further, the unique variables were dropped from the dataframe and the duplicate check was performed again (Figure *15*, Figure *16*). This jumped the number to 1,385,270 entries with a total leftover of 229,907 unique values. The original datasets were also checked for potential duplicates (after removing the ID variable from Covid1 to ensure the function would work) resulting in respectively 93,858 and 812,049 entries for Covid1 and Covid2 (Figure *17*). The staggering difference between the values is due to Covid1 having 3 date-related variables giving more uniqueness to the tuples. With that exploration completed, it was deemed important to keep all entries as it was unlikely that Covid2 contained >800,000 accidental duplicates. Even if the tuples appear as duplicates due to having similar values, it is important to keep them in the model as they represent real-world patient data. If there are multiple patients within one dataset that have the same values, then the model certainly will face these values again while performing on further real-world data in production usage. Thus, comparing duplicate checks across the original datasets and the integrated dataframe provided a deeper understanding of data overlaps, especially in patient records. Retaining duplicates allowed the data to be more comprehensive and was justified by the need for completeness in this analysis phase.

Again, the cleaning process moved on to refining the dataset structure by removing variables deemed unnecessary (Figure *15*). These were columns that were unique to only one of the original datasets and did not add value to the combined analysis. By dropping these extraneous variables, the dataset became more focused, streamlining subsequent data handling and model training processes. This step helped reduce noise and ensured that computational resources were directed toward processing relevant information. After streamlining the dataset, another duplicate check was performed to validate that the structural changes did not introduce inconsistencies.

To ensure data accuracy and usability, the next steps included handling missing values and standardizing categorical variables. All tuples containing placeholder codes indicating missing data (such as 99 and 98) were removed, resulting in a dataset free from incomplete information that could mislead model predictions (Figure *18*). Following this, a standardization effort was conducted to replace categorical values marked as 2 with the conventional 0, ensuring uniformity across variables and improving data quality (Figure *19*). Finally, any tuples where the age exceeded 100 were removed to exclude unrealistic outliers that could distort analysis (Figure *20*). There were no further skewed variables, outliers, or other issues which would require further data cleaning. This comprehensive, multi-step cleaning process ensured that the dataset was accurate, reliable, and well-prepared for the next stages of exploratory data analysis and predictive modeling. Each step was methodically applied to maximize data integrity and analytical effectiveness. With this, KPI 4 has been successfully achieved where total number of rows in the final dataframe are 1,580,762 (where there were originally 1,615,177 total tuples in the

integrated dataset, minus 203 from >100 age tuples, minus 34,212 tuples from missing values, for a 97.9% total data integration completeness[= 1,580,762 / 1,615,177]).

Data Transformation

In the process of preparing the dataset for predictive modeling, a range of feature engineering options were considered. Initial thoughts included developing complex interaction terms between certain comorbidities (like describing chronic conditions or conditions related to the same systems of the body) or adding temporal features like time-of-year admission to capture potential seasonal trends in patient outcomes. While potentially valuable, these features were ultimately not pursued due to data limitations and the need for more straightforward, interpretable features that align with the available information and desired outputs. Interaction terms, for instance, can add complexity that might be challenging to interpret and explain to stakeholders. Additionally, seasonal or time-based features were less relevant to the immediate scope of understanding patient severity, as the primary focus was to assess individual patient characteristics rather than external factors. As a result, the focus was narrowed to more direct and informative variables that could be derived from existing data columns. Also, the features engineered needed to either simplify the data or highlight a trend. With this in mind, the following approaches were taken.

One significant transformation implemented was the creation of the AGE_GROUP variable (Figure *21*). This feature involved segmenting the continuous AGE column into distinct categorical buckets such as "0-20," "21-40," and so on, up to "81-100." Bucketing age into groups simplifies analysis by allowing models to capture

non-linear relationships between age and patient outcomes. This approach was chosen because certain age ranges are known to correlate differently with health outcomes, and this segmentation helps models learn those variations more effectively. Additionally, age bucketing provides more interpretability than a raw continuous variable. This makes it easier to identify which age categories are more at risk without overfitting to specific age values. While it is possible that this could obfuscate information since it makes the lines blurrier in this dataset, it is thought that adding these age buckets should be an overall net positive. They should add more streamlining to the analysis that information is taken away from the dataset. Most of the 20-year buckets act mostly similar to one another physiologically. While there may be outliers to this, the overall trend should follow that patients within each of these buckets will be similar to one another making it easier for the model to predict their outcomes.

Another key transformation was the creation of the COMORBIDITY_COUNT variable (Figure *21*). This feature sums the binary indicators of various conditions present in a patient's record, such as diabetes, COPD, and cardiovascular disease, to provide a single, aggregated measure of comorbidity burden. The rationale for this feature lies in its ability to condense multiple related columns into a single variable that reflects overall patient risk. Patients with higher counts are likely to face more severe health challenges, which can be highly predictive of adverse outcomes. This simplification allows for a nuanced yet straightforward input for machine learning models, highlighting the cumulative effect of having multiple health issues without overcomplicating the data structure. Both AGE_GROUP and COMORBIDITY_COUNT were chosen for their

clarity, interpretability, and ability to capture significant patient risk factors that align well with predictive modeling objectives.

The creation of the MORTALITY variable is an essential transformation in the dataset, converting the DATE_DIED column into a binary indicator where valid dates signify a mortality event (represented by 1) and placeholder dates like '9999-99-99' indicate survival (represented by 0)(Figure *22*). This new variable simplifies the data structure, making it more suitable for predictive modeling and analysis by converting potentially complex date information into a straightforward, interpretable feature. Justifying this transformation lies in its ability to enhance model training by providing a clear target variable that aligns with real-world healthcare outcomes. Additionally, the binary nature of the MORTALITY variable allows for streamlined visualizations and decision-making, enabling quick identification of mortality trends and patterns across other features in the dataset. This step ensures that the model can focus on significant patient outcomes, thereby improving the interpretability and effectiveness of predictions.

Further feature engineering was deemed unnecessary as there were no additional variables within the dataset that could be generated to add significant value or improve analytical outcomes. The existing features already capture key aspects of patient health, comorbidities, and treatment variables, providing sufficient depth for predictive modeling. Introducing more engineered variables could lead to overfitting or unnecessary complexity without contributing meaningful insights. Thus, the current set of features was determined to be comprehensive and appropriate for robust analysis.

Data Analysis

To conduct thorough data analysis and build effective predictive models, a mix of Python libraries and Tableau will be employed. Visualizations will first be developed using Matplotlib and Seaborn within the Python environment, specifically in Spyder 3.3.6, which offers an efficient and familiar interface for Python programming. Seaborn, known for its ability to generate informative and visually appealing plots, will be utilized for initial data exploration. Visuals such as bar charts, histograms, and correlation heatmaps will be created to examine the distribution of key features, understand relationships between comorbidities and outcomes, and detect potential multicollinearity. These preliminary visualizations are essential for providing a strong foundation for feature selection and model development, as they allow for a detailed understanding of data trends and outliers that may influence model performance. The primary visualization to make before modeling will be a correlation heatmap using Seaborn to understand which variables are correlated with one another. This will accomplish multiple goals at once like showing which variables move more closely and less closely in lockstep with the target variables, and which variables may be tied to one another. Also the same bar chart matrix made for the Covid1 and Covid2 datasets will be recreated for the integrated dataset for comparison.

In addition to Python visualizations, Tableau will be used to create interactive and dynamic dashboards, showcasing a variety of visual types that offer greater flexibility and stakeholder engagement. Tableau is particularly valuable for presenting complex data in an accessible format, allowing non-technical stakeholders to interact with the data through filters and drill-down capabilities. For this dataset, interactive heatmaps could be

designed to reveal how combinations of comorbidities relate to patient outcomes such as ICU admission and mortality (especially for comparison against a Seaborn version). Additionally, line graphs depicting trends over time, segmented by age group or other key features, will help uncover temporal patterns and their potential influence on severe outcomes. In particular, Tableau will be helpful to make histograms of the comorbidities over ICU admission and mortality as well as box plot distributions of age versus the comorbidities and the target variables in order to understand how age affects these factors. The use of Tableau ensures that the analysis is not only comprehensive but also actionable, providing a means to visualize complex relationships and support data-driven decision-making at all levels.

For predictive modeling, Python's Scikit-learn library will be the primary tool due to its extensive range of algorithms and ease of use for both prototyping and performance tuning. The project will leverage models including logistic regression, decision trees, random forests, gradient boosting, and support vector machines (SVM). Logistic regression will serve as a baseline model, offering insights into the linear relationships between independent variables and the outcomes, which can be particularly valuable for understanding the impact of individual predictors. Decision trees, known for their straightforward interpretation, will map out decision paths that can be visualized to show how variables like age group or comorbidity counts contribute to predicting outcomes. Random forests and gradient boosting models, as ensemble techniques, will help capture more complex interactions among variables, reducing overfitting and improving the robustness of the models. SVMs will add depth by handling non-linear relationships

between features and outcomes, which could be pivotal in accurately classifying cases with intricate patterns.

The use of these specific models is justified by their balance of interpretability and predictive power. In the context of healthcare, transparency is key; stakeholders must understand why a model is making certain predictions, especially when patient care and resource allocation are involved. Decision trees and ensemble methods like random forests and gradient boosting allow for feature importance to be evaluated, highlighting which factors contribute most to severe outcomes. This is essential for building trust and ensuring that insights can inform clinical or operational strategies. Pairing these models with visual analysis tools like Seaborn and Tableau further enhances the project by enabling comprehensive storytelling through data. This combination of detailed, static plots and dynamic, interactive dashboards ensures that insights are clear, thorough, and backed by rigorous analysis, aligning with the project's goal of producing actionable and interpretable results.

Data Visualization

Data Visualization 1

Correlation matrices provide a comprehensive overview of the relationships between variables in the dataset, offering critical insights into feature interactions. It is critical to note that the age variable was left as a continuous variable (instead of categorical) for the sake of this visualization only. The first visualization, a standard heatmap generated using Seaborn in Python, displays the Pearson correlation coefficients

between all pairs of variables. The Pearson coefficient of correlation is a statistical measure that quantifies the linear relationship between two continuous variables, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation)(Sedgwick, 2012). A diverging color scheme ranging from deep red (indicating strong positive correlations) to blue (representing strong negative correlations) was used to highlight these relationships. The heatmap includes exact correlation values within each cell, allowing for precise interpretation of the strength and direction of these associations. For instance, the correlation between ICU and INTUBED is a robust 0.65, which is expected given the clinical need for intubation in critically ill patients. Meanwhile, weaker correlations, such as AGE and ASTHMA (-0.03), suggest minimal relationships that are unlikely to drive predictive power. This visualization is particularly useful for identifying redundant features and those with strong predictive potential.

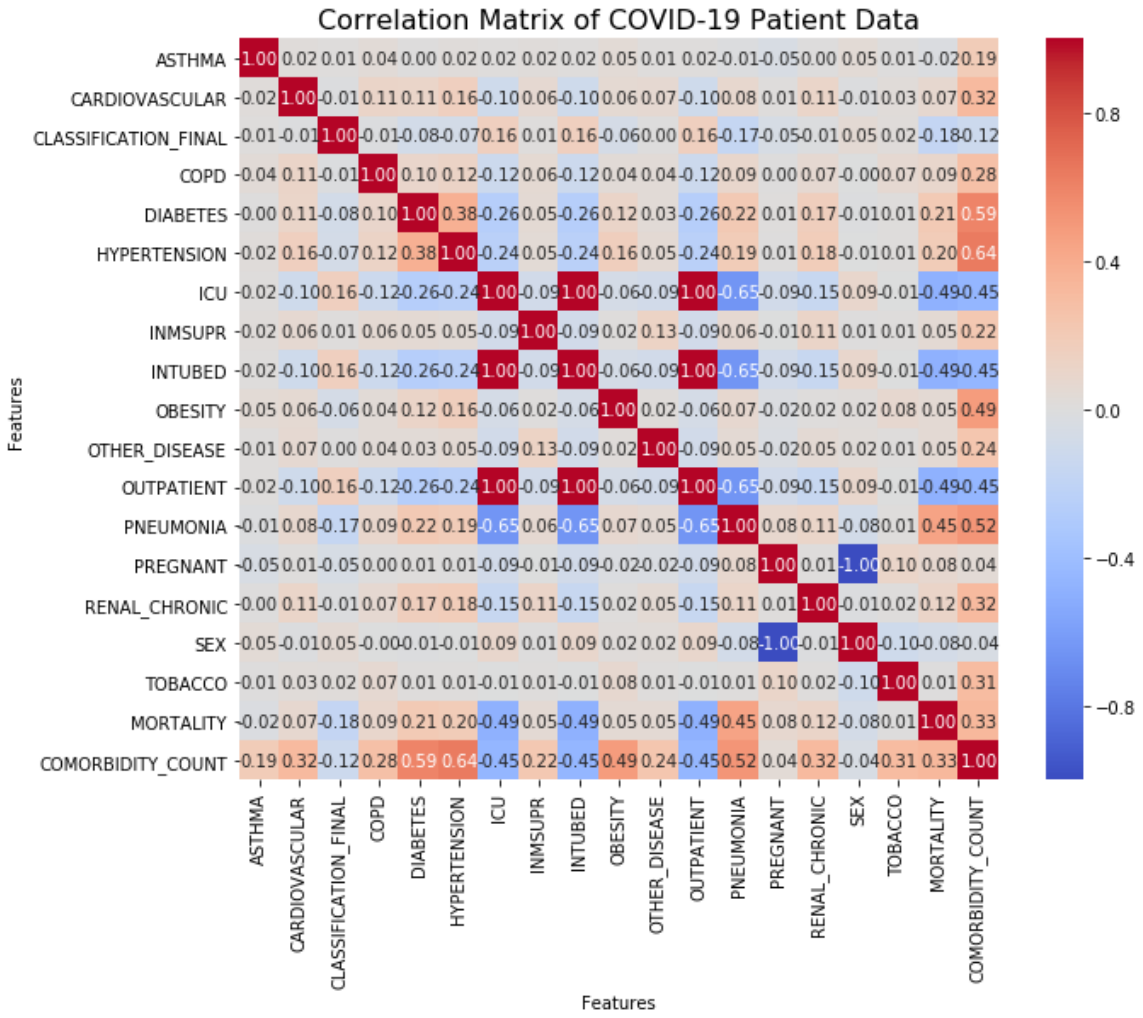

Figure 1. Correlation matrix heatmap of the final dataframe before analysis showing the correlations between all variables to highlight trends before analysis.

Secondly, a clustered correlation heatmap was made to build upon the standard matrix by incorporating hierarchical clustering. This was achieved by using Seaborn in conjunction with a clustering algorithm to group variables with similar correlation patterns. The resulting dendrogram visually organizes features based on their correlation structures, making it easier to spot clusters of related variables. For example, DIABETES, HYPERTENSION, and AGE conditions form a distinct cluster, with correlations ranging from 0.33 to 0.39. This structure suggests these comorbidities often occur together, reinforcing their combined predictive significance. The clustered heatmap

offers a higher-level view of the data's structure, aiding in feature selection and helping analysts identify variables that could potentially be aggregated or transformed.

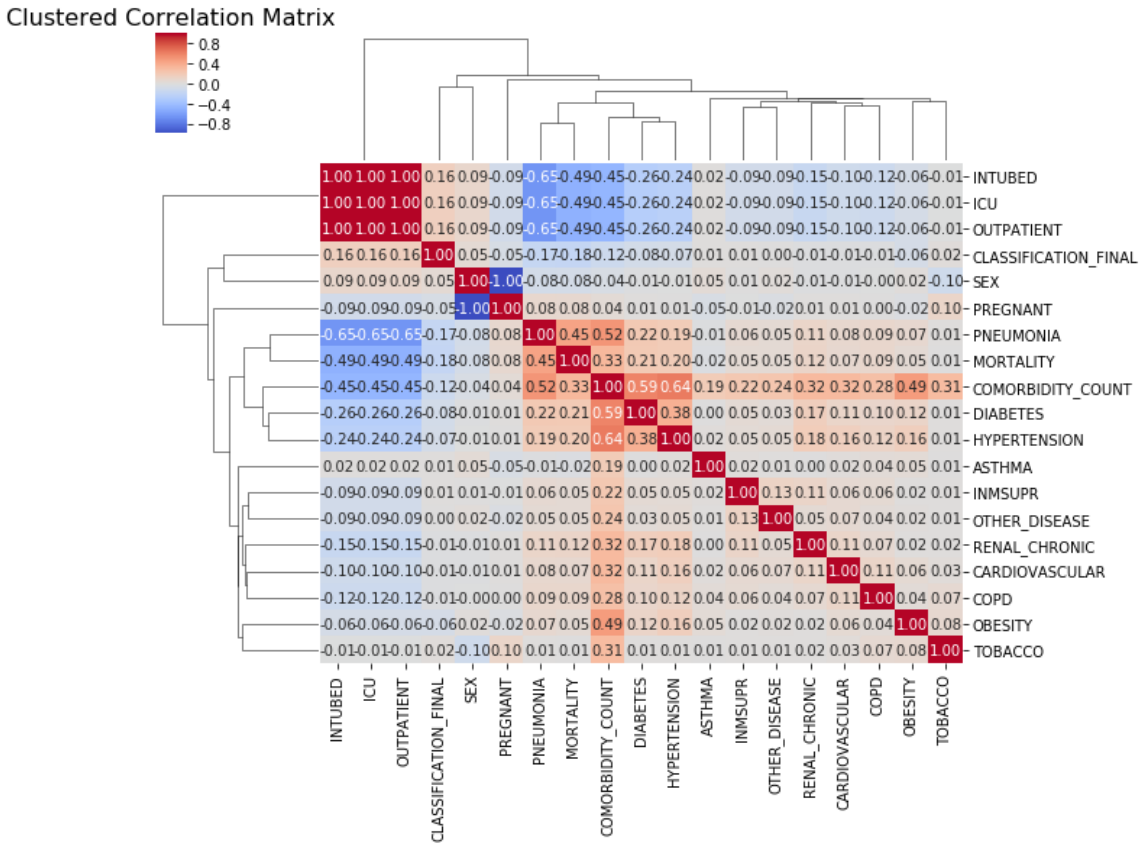

Figure 2. Correlation matrix heatmap of the final dataframe with clustering between correlated variables to visually parse trending variables.

Both visualizations are crucial tools for exploratory data analysis. By visualizing correlations in a matrix format, researchers can easily identify features with strong predictive relationships to the target variables, MORTALITY and ICU. Additionally, hierarchical clustering offers insights into multicollinearity and potential feature reduction strategies. Together, these visualizations streamline the analysis process by highlighting key variables, reducing noise, and ensuring that subsequent predictive modeling efforts focus on the most impactful features. The visualizations not only provide interpretability but also guide the data preparation process, ultimately leading to

more accurate and reliable models. It is critical to note that these correlations are based numerically; while most of these categories will be based only on 0/1 binary values, any of the variables with “non-applicable” values of 97 (INTUBED, PREGNANT, OUTPATIENT) will strongly correlate positively if the 97 correlates with other variables.

The correlation heatmaps reveal several important trends and relationships within the dataset. Notably, INTUBED and ICU have the highest positive correlation (0.65), underscoring 1. the strong link between the need for mechanical ventilation and critical care and also 2. the previously discussed link between the non-applicable value of 97. Similarly, MORTALITY is moderately correlated with ICU (0.49) and PNEUMONIA (0.45), confirming their importance as predictors of severe outcomes. These insights validate the selection of these features for predictive modeling. Additionally, the correlation between AGE and comorbidities such as DIABETES (0.28) and HYPERTENSION (0.33) suggests that older patients are more likely to have chronic conditions, which can exacerbate disease severity. Interestingly, AGE is negatively correlated with OUTPATIENT, ICU, and INTUBED and positively correlated with MORTALITY showing how younger patients were more likely to be outpatient and older patients were more likely to be susceptible to COVID-19 death. This emphasizes the need to account for both age and comorbidities in the model.

Interestingly, some variables display very low or negligible correlations with the target variables. For instance, TOBACCO shows almost no correlation with MORTALITY (0.01) or ICU (0.02), indicating it may have limited predictive value in this context. Similarly, ASTHMA and SEX exhibit correlations close to zero with most

outcome variables. These findings suggest that these features may be deprioritized or excluded from the final model to improve efficiency without sacrificing accuracy. However, their inclusion in exploratory analysis is still valuable for ensuring a comprehensive understanding of the dataset.

The clustered heatmap provides additional insights into feature groupings. For example, the clustering of DIABETES, HYPERTENSION, and CARDIOVASCULAR conditions aligns with their role as common comorbidities that collectively contribute to higher mortality risk. This clustering highlights the potential for creating composite variables or leveraging ensemble models to capture these interactions more effectively. By grouping similar features, the clustered heatmap simplifies the process of identifying which combinations of variables are most influential, offering a more targeted approach to feature engineering and selection.

The insights from the correlation matrices have important implications for the scope and focus of the project. Initially, all features were considered equally important for predictive modeling, but the matrices reveal that only a subset of variables exhibit strong correlations with the target outcomes. For instance, the moderate to strong correlations of ICU, PNEUMONIA, and INTUBED with MORTALITY justify prioritizing these features. Conversely, weak correlations for variables like TOBACCO and ASTHMA suggest they might not significantly contribute to prediction accuracy. This finding allows for a more streamlined approach, focusing resources on the most impactful variables while potentially excluding those with limited predictive value.

The moderate correlation between AGE and comorbidities like HYPERTENSION (0.33) and DIABETES (0.28) also suggests that age could be a proxy for multiple health risks. This insight might reduce the need for highly complex models, as simpler models can still capture the compounded effects of age and comorbidities. Additionally, the clustered heatmap reveals that certain feature groups, such as comorbidities, could be combined into composite indices, reducing dimensionality and improving model interpretability. This adjustment ensures that the model remains interpretable and actionable while maintaining high predictive performance. Lastly, the high correlation between ICU and other severe outcomes suggests the potential for dual-use models. A model trained to predict ICU admissions could also be repurposed to approximate mortality risk, broadening the model's utility. This expanded scope enhances the project's practical applications, allowing healthcare providers to use the model for multiple critical decisions, such as patient triage and resource allocation. These insights, driven by correlation analysis, ensure that the project remains both focused and adaptable, maximizing its impact on healthcare outcomes.

Data Visualization 2

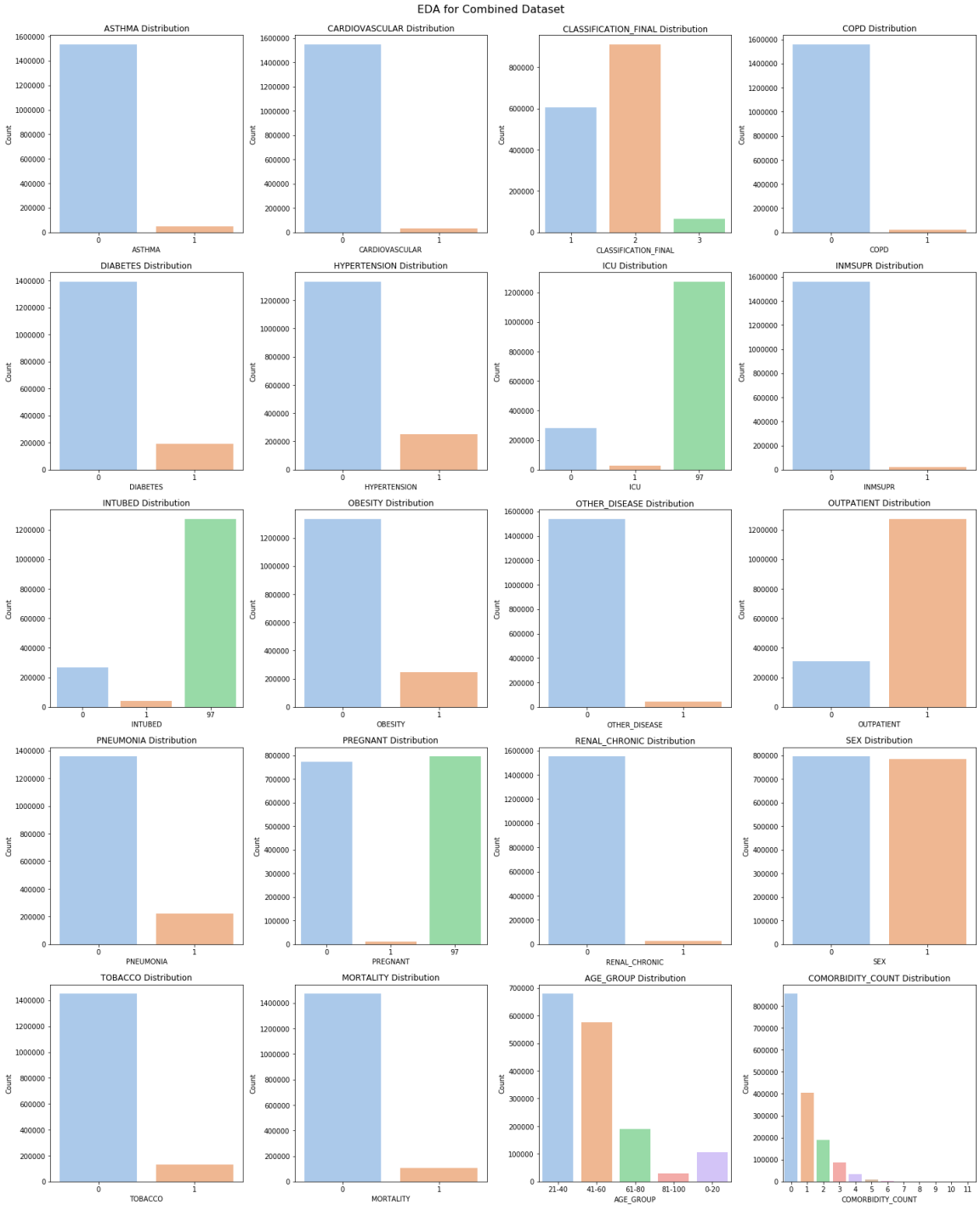

Figure 3. Bar chart matrix of all variables of the final cleaned integrated dataset showing final EDA patterns of the dataset.

Similar to the bar chart matrices for Covid1 and Covid2, a bar chart matrix was made for the final integrated dataset (Figure 3). This provides a comprehensive view of feature distributions across different stages of data preprocessing. The first two matrices (Covid1 and Covid2)(Figure *7* and Figure 8) focus on the individual datasets' characteristics before integration. They included all original variables which are represented with light blue bars. They included common variables such as AGE, ICU, and comorbidities as well as unique variables like ID and USMER. These visualizations previously offered insight into each individual dataset's unique variable distribution. For example, ICU in Covid1 shows a slightly different balance between critical care admissions compared to Covid2.

The final integrated dataset’s bar chart matrix, with its more diverse color scheme, reflects the combined power of the two datasets. The integrated visualization incorporates a color-coded classification system for variables like CLASIFFICATION_FINAL, indicating the severity of Covid-19 diagnosis, and all variables with the non-applicable value included. Each variable’s spread across the dataset is now more comprehensive, highlighting the advantages of data integration. Notably, previously sparse variables like PREGNANT in Covid2 now show more balanced distributions when integrated. This provides a clearer picture of patient demographics and medical conditions, making it easier to detect trends or patterns.

Seaborn was instrumental in creating these visualizations, particularly in handling categorical data distribution. The integration results in more balanced and informative distributions for previously unrepresented or underrepresented variables. These

visualizations reveal the dataset's complexity and enable the analyst to prepare better for modeling by identifying key discrepancies, redundancies, and areas where data were enriched through integration.

The integration of datasets highlights a significant improvement in data completeness and balance, particularly in critical variables like ICU and MORTALITY. In the individual datasets, some features, such as PNEUMONIA, show highly imbalanced distributions. However, upon integration, the combined visualization reveals more robust patterns in patient outcomes. This suggests that integration enhances our ability to discern finer distinctions in disease severity, ultimately aiding in the construction of more nuanced predictive models. Another key insight lies in the AGE distribution. The individual datasets exhibit similar bell-shaped curves (with a right-skew due to outliers of >100), but integrating them ensures that outliers are slightly smoothed out. The categorization of this variable also should help the models make more efficient and clear decisions when predicting patient outcomes. This contributes to a more accurate representation of the patient population, which is critical for age-sensitive predictions. Similarly, the INTUBED and ICU variables, strongly correlated with severe outcomes, demonstrate consistent distributions post-integration. This uniformity provides confidence that critical features retain their importance and relevance in the integrated dataset, ensuring consistency in predictive modeling outcomes.

The integration and visualization process have revealed key insights that reshape the project scope. Initially, each dataset offered partial perspectives on patient outcomes. However, the integrated dataset uncovers richer patterns, particularly in the target

variables ICU and MORTALITY. These now display consistent and comprehensive distributions, which provide more reliable training data for machine learning models. The integration also mitigates the biases inherent in individual datasets, ensuring that models generalize better across different patient demographics. The new variable COMORBIDITY_COUNT also yields a broad perspective from all patients of both datasets. It follows a Pareto distribution where the] majority of patients have no comorbidities (~900,000), followed by a handful who have 1-3 comorbidities (~600,000), followed by only a few who have $\geq$4 comorbidities (<50,000). We see this reflected in the other comorbidity bar charts where most patients don't have a comorbidity but only some do, thus it becomes far rarer for patients to have several combined comorbidities.

Moreover, the final visualizations highlight potential areas of model improvement. The uniform distribution of comorbidities and age groups ensures that models trained on this dataset will not only predict severe outcomes accurately but will also identify high-risk subpopulations with greater precision. This shifts the project's emphasis from actionable healthcare insights, to targeted interventions of comorbidities. Lastly, the integrated dataset provides a more reliable basis for operational healthcare decisions. The enhanced variable distribution improves model interpretability and predictive accuracy, particularly for critical classifications like ICU admission risk. This comprehensive view equips stakeholders with better tools to prioritize healthcare resources effectively, ensuring that high-risk patients receive timely care, ultimately aligning the project more closely with its goal of improving patient outcomes.

Data Visualization 3

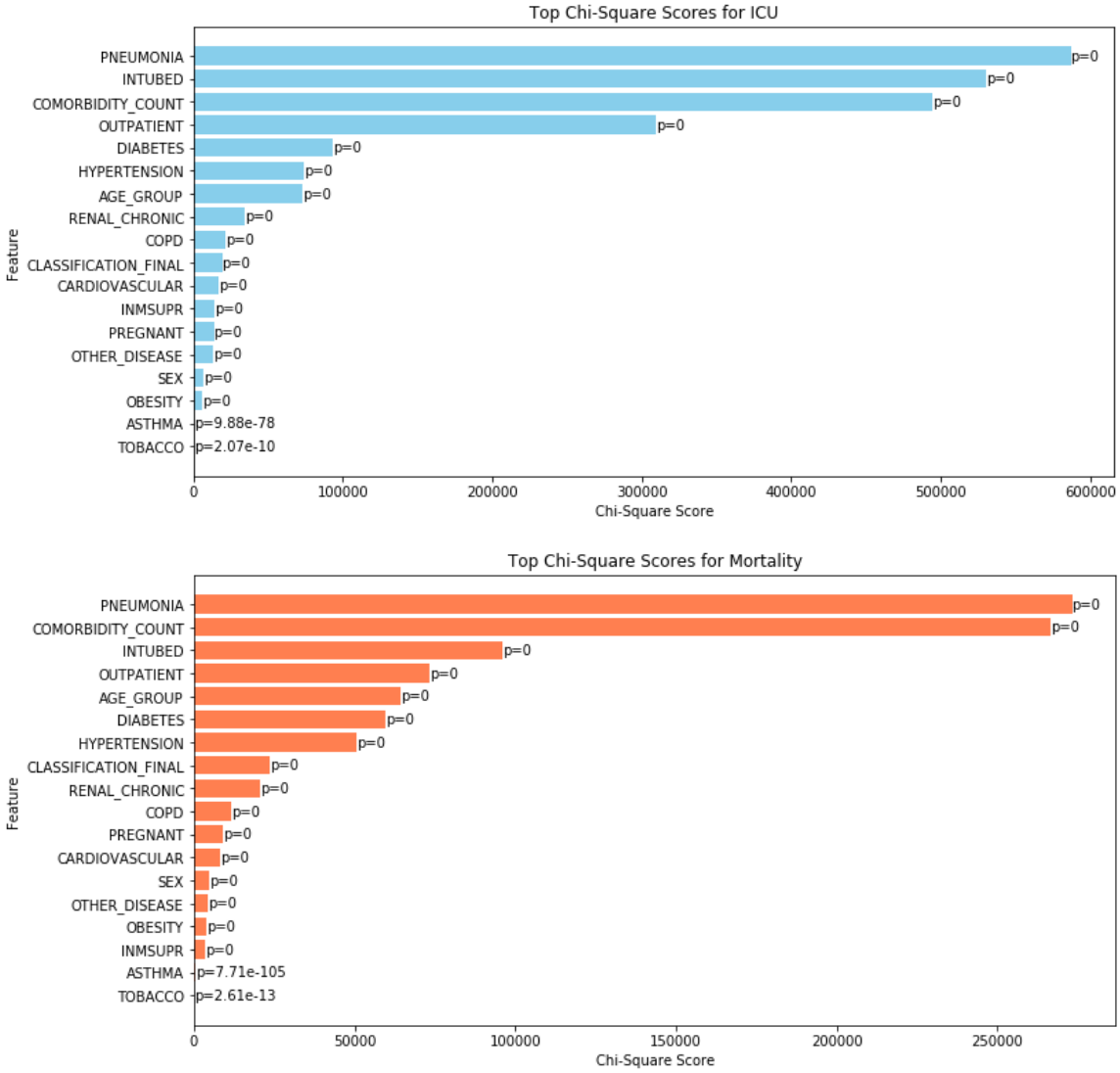

Figure 4. Chi-square bar charts for both ICU and MORTALITY showing p-values of all predictive variables.

The visualizations presented display the top Chi-Square scores for two critical target variables: ICU admission and mortality. These bar charts highlight the statistical relationships between categorical features and the outcomes of interest, with p-values prominently displayed for each feature. The Chi-Square test measures how much the observed data distribution deviates from what we would expect under independence, providing an indication of each variable's relevance to the target (Barceló, 2018). The

variables are ranked by their Chi-Square scores, with PNEUMONIA and INTUBED scoring highest for ICU, suggesting their strong association with critical care admissions. Similarly, PNEUMONIA dominates for MORTALITY, followed by COMORBIDITY_COUNT and INTUBED, underscoring their impact on patient survival.

The use of a light blue color scheme for ICU and orange for mortality ensures clear visual distinction between the two target analyses. The layout provides an immediate grasp of which variables are most significant, as higher bars represent stronger associations. The p-values for most variables are essentially zero, reflecting their high statistical significance in predicting ICU admission and mortality. Seaborn's barplot functionality enabled the creation of these clean, interpretable visuals, which efficiently convey complex statistical relationships in an accessible manner. These visualizations offer not only a ranked understanding of feature importance but also illustrate differences in the driving factors for ICU admission versus mortality. While both outcomes share some key predictors like PNEUMONIA and INTUBED, their relative influence shifts slightly, with COMORBIDITY_COUNT playing a larger role in predicting mortality. This relationship with intubation and ICU admission is obvious since most patients in the ICU requiring the highest level of care would also require intubation. This divergence highlights the nuanced relationship between patient features and healthcare outcomes, guiding model selection and feature engineering strategies.

The Chi-Square analysis reveals compelling insights into the drivers of severe healthcare outcomes. PNEUMONIA emerges as the most critical predictor for both ICU

admission and mortality, underscoring the severe respiratory complications that often lead to critical conditions. Its exceedingly high Chi-Square scores, combined with near-zero p-values, solidify its role as a primary indicator for both targets. Additionally, INTUBED, representing mechanical ventilation, ranks highly for ICU and mortality, reflecting its necessity in managing severe respiratory distress.

Notably, COMORBIDITY_COUNT plays a crucial role in predicting mortality but has a lower impact on ICU admissions. This suggests that the cumulative burden of comorbidities is more directly tied to survival probabilities than to immediate critical care needs. OUTPATIENT, which inversely reflects hospital admissions, also ranks highly, emphasizing its importance in distinguishing between patients requiring intensive care and those who can be managed in less acute settings. These findings highlight the importance of including variables that capture both acute conditions and chronic health burdens in predictive models. Interestingly, some variables, such as OBESITY, TOBACCO, and ASTHMA, show relatively lower Chi-Square scores, suggesting they have less predictive power for these outcomes. While they remain statistically significant, their weaker associations imply limited utility in the context of ICU and MORTALITY predictions. This information can guide feature selection by helping analysts prioritize variables with higher predictive value, ensuring that models focus on the most impactful factors without being bogged down by noise.

These insights will significantly shape the modeling approach and overall project scope. The Chi-Square results confirm the appropriateness of including high-ranking variables like PNEUMONIA, INTUBED, and COMORBIDITY_COUNT in predictive

models for ICU admission and mortality. Their strong associations ensure that the models will capture critical health dynamics accurately. Moreover, the clear distinction in feature importance between the two outcomes necessitates tailored modeling strategies for each target. For instance, models predicting mortality should place greater emphasis on cumulative health burdens, while ICU models should focus more on acute interventions like intubation.

The findings also underscore the need for interpretability in the modeling process. The inclusion of high-impact but understandable variables aligns with the project's goal of producing actionable insights for healthcare providers. Understanding which features drive predictions allows stakeholders to intervene effectively, whether by prioritizing patients for ICU beds or implementing preventive measures for high-risk individuals. These visualizations ensure that the project remains firmly grounded in clinical relevance, enhancing its value for decision-making and resource allocation. Finally, the differentiation in feature importance between ICU and MORTALITY suggests potential expansions in scope. The project could extend its focus to developing dual-use models that predict both outcomes simultaneously or explore alternative health metrics. While these expanded scopes would likely to have be explored in a different project, these insights reinforce the project's central aim of improving patient care while optimizing healthcare operations, making it an invaluable tool for public health planning.

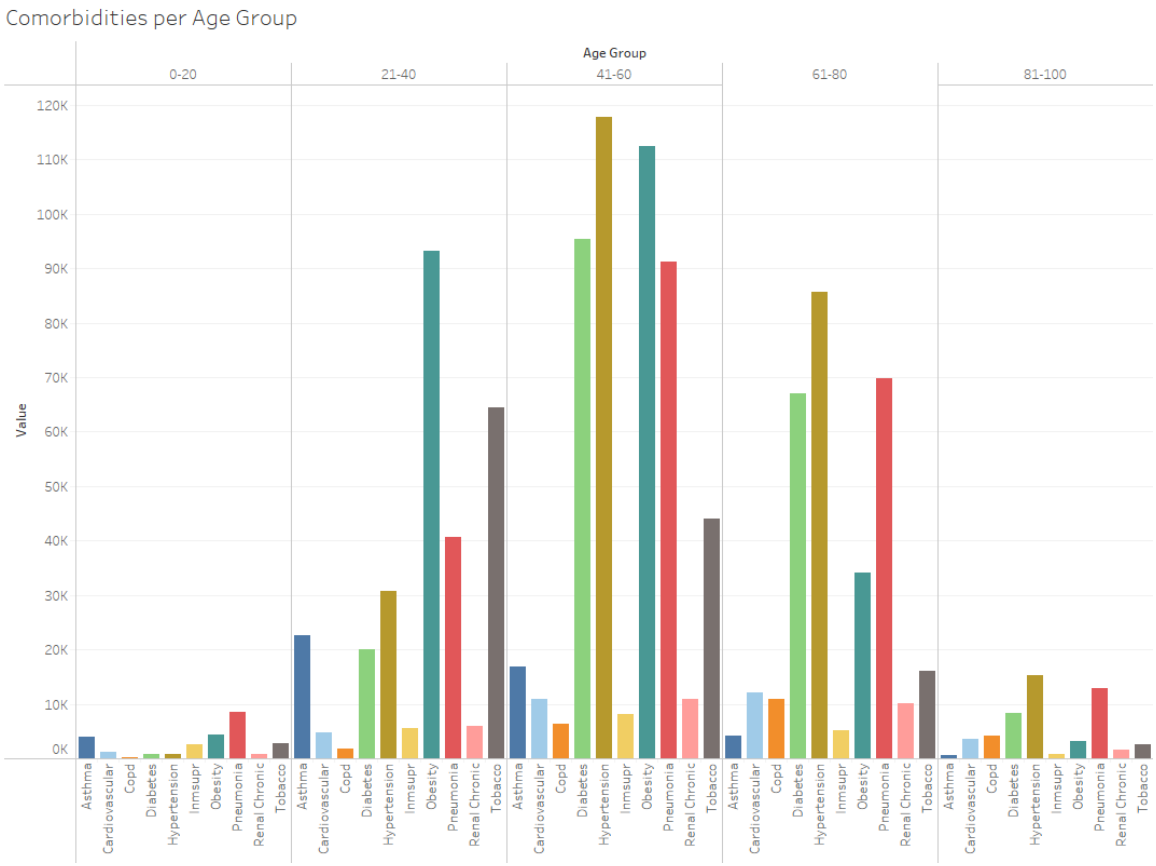

Figure 5. Bar chart demonstrating the spread of comorbidities per age group within the integrated dataset generated in Tableau.

Finally, the bar chart in Figure 5 offers an in-depth view of the distribution of comorbidities across different age groups within the integrated dataset. Each comorbidity - such as diabetes, hypertension, obesity, and pneumonia - is color-coded to easily distinguish their prevalence among the five defined age groups: 0-20, 21-40, 41-60, 61-80, and 81-100. This visualization is generated in Tableau, allowing for interactive exploration of the data. It provides immediate visual insights into the varying health risks across age demographics. The decision to use a bar chart for this analysis was driven by the need to compare absolute frequencies of comorbidities within age brackets effectively. From a technical standpoint, this bar chart was designed to emphasize the heterogeneity of health risks by age group. Tableau's bar chart capabilities, which

provide clear and concise comparisons, were utilized to display absolute counts of comorbidities rather than percentages. This ensures that the visual reflects the true burden of each comorbidity within the dataset. The x-axis lists comorbidities, while the y-axis quantifies their occurrences. Tableau's ability to group and filter data dynamically allows users to drill down into specific age brackets, facilitating deeper analytical insights.

In terms of insights, the 41-60 and 61-80 age groups emerge as the most burdened by comorbidities such as hypertension, diabetes, and obesity. Notably, pneumonia and intubation are more frequent among the older age groups, particularly 61-80, underscoring their vulnerability to severe outcomes. In contrast, comorbidities like asthma and cardiovascular issues are relatively less prominent across all age groups. The presence of chronic conditions such as renal chronic disease and tobacco use also shows a gradual increase with age, peaking in the older age brackets. This suggests that as individuals age, the likelihood of accumulating multiple health conditions increases significantly.

The broader scope of this analysis highlights how age influences the distribution of specific health risks. The 21-40 age group shows lower levels of chronic diseases but higher rates of conditions like pregnancy and tobacco use. These findings emphasize the necessity for age-specific healthcare interventions. For instance, prevention and early management strategies for hypertension and diabetes should be prioritized for middle-aged adults to reduce their long-term impact on public health outcomes. Finally, this bar chart complements previous visualizations by focusing on the categorical breakdown of comorbidities rather than correlations or mortality outcomes. The combined insights from

these visualizations provide a holistic understanding of the dataset, supporting more targeted and data-driven healthcare strategies. The identification of high-risk comorbidities per age group allows for improved resource allocation and the development of preventive care measures tailored to specific population needs.

Proposed Visualizations

A valuable visualization to create in Python would be a pairplot of key comorbidities alongside the target variables, ICU and mortality. This plot would allow us to examine relationships between various health conditions such as diabetes, hypertension, and obesity in relation to severe outcomes. By highlighting the distributions of these variables and their pairwise scatter plots, we could observe patterns that suggest correlations or interactions between comorbidities. For example, it may reveal clusters of patients with multiple comorbidities who have significantly higher ICU admissions or mortality rates. The pairplot's ability to visually identify nonlinear relationships makes it a crucial tool for guiding feature selection and engineering before predictive modeling. Including the hue parameter to distinguish between ICU and non-ICU cases, or mortality and survival outcomes, would further enhance the analysis.

Another impactful visualization in Tableau would be a dynamic bar chart comparing the prevalence of different comorbidities across ICU and non-ICU patients. This visualization would allow users to quickly assess which health conditions are most associated with severe outcomes. By categorizing the data into age groups within each bar, we can add an extra layer of insight to understand how age influences the risk associated with specific comorbidities. For instance, younger patients with diabetes may

have a lower ICU admission rate compared to older patients with the same condition. Tableau's interactive features would enable stakeholders to filter and sort the data dynamically, making it easier to focus on specific conditions or demographics.

A third visualization in Tableau could be a line chart illustrating the progression of key health indicators, such as the percentage of ICU admissions or mortality, as a function of comorbidity count. This would provide a clear visual representation of how the accumulation of health conditions affects patient outcomes. By tracking changes in severe outcomes as comorbidities increase from none to multiple, this chart could highlight thresholds where risks escalate significantly. Such insights are critical for developing clinical decision-support systems that prioritize patients based on their comorbidity profiles. Additionally, combining this visualization with age or other demographic filters would offer even more granular insights, supporting data-driven healthcare strategies.

These visualizations offer a deeper understanding of the dataset and bring unique insights that complement the exploratory data analysis conducted so far. The pairplot differs from the earlier correlation matrix by not only showing linear relationships but also allowing us to observe potential clusters or outliers in the data. This is especially important when dealing with variables that may have complex, nonlinear interactions. By identifying such patterns, the pairplot can inform the development of more sophisticated predictive models, ensuring that critical relationships are not overlooked during feature selection.

The dynamic bar chart in Tableau, on the other hand, focuses on categorical comparisons, highlighting the distribution of comorbidities within the ICU and non-ICU groups. Unlike the previous bar chart matrices, which provided an overview of the prevalence of individual variables, this visualization emphasizes comparative analysis between target outcomes. By incorporating age group categories within each bar, the chart allows for a more nuanced understanding of how risks vary across demographic segments. This dual-layered approach helps in identifying not just which comorbidities are prevalent but also how their impact differs across patient subpopulations, offering critical insights for targeted interventions.

Lastly, the line chart tracking severe outcomes as a function of comorbidity count provides a longitudinal perspective, which is absent in other visualizations. While earlier charts have focused on static distributions or pairwise comparisons, this line chart visualizes the cumulative risk associated with multiple comorbidities. This is crucial for healthcare planning as it helps to pinpoint the tipping point where patient risk escalates significantly, enabling more effective prioritization of high-risk individuals. Additionally, by allowing filters for age or other demographic variables, this visualization becomes a versatile tool for exploring the interplay between comorbidity burden and patient outcomes in different population segments. Together, these visualizations provide a comprehensive toolkit for exploring the dataset from multiple angles, facilitating a deeper and more actionable understanding of the factors driving severe health outcomes.

In addition to bar charts and correlation matrices created, other visualizations could provide unique perspectives on the data adding more variety in addition to the other

suggested visualizations. Stacked bar charts allow for the comparison of multiple categorical variables simultaneously, highlighting how combinations of factors influence target outcomes like ICU admission or mortality. Mosaic plots are also valuable for visualizing the relationships between multiple categorical variables, showing proportional differences in outcomes across subgroups. Another useful option is heatmaps of conditional probabilities, which display the likelihood of a target outcome given specific categories of predictor variables, offering a clear view of nuanced patterns. Despite these possibilities, the decision to rely primarily on bar charts and correlation matrices was driven by their ability to distill complex relationships into straightforward insights. Bar charts effectively capture the distribution and comparative significance of predictors, while correlation matrices highlight variable relationships in a concise format. Together, these visualizations provided the clearest and most actionable insights for the dataset's categorical nature.

Predictive Models

Before diving into the discussion on modeling, first there must be a conversation regarding the accuracy measures and confusion matrices. Accuracy measures derive from the confusion matrix, a table summarizing the performance of a classification model by comparing predicted and actual outcomes (SSP, 2024). The confusion matrix consists of four key components: True Positives (TP), where the model correctly predicts positive cases; True Negatives (TN), where it correctly predicts negative cases; False Positives (FP), where the model incorrectly predicts a positive case as negative (Type I error); and False Negatives (FN), where a positive case is missed (Type II error).

From the confusion matrix, Accuracy is calculated as (TP+TN) / (TP+TN+FP+FN) (TP + TN) / (TP + TN + FP + FN) (TP+TN) / (TP+TN+FP+FN), representing the overall correctness of the model (all equations can be seen in the attached Figure *23* which will make them clearer for the reader). However, in datasets with imbalanced classes, accuracy can be misleading, as it may favor the majority class. The F1 score, derived as the harmonic mean of precision (TP/(TP+FP))(TP / (TP + FP))(TP/(TP+FP)) and recall (TP/(TP+FN))(TP / (TP + FN))(TP/(TP+FN)), addresses this issue by balancing false positives and false negatives, making it particularly useful when misclassifications have significant consequences. Sensitivity (or recall) focuses on the model's ability to identify true positive cases, calculated as TP/(TP+FN)TP / (TP + FN)TP/(TP+FN), and is vital for ensuring that critical cases like ICU admissions or mortality are not overlooked. Thus, this addresses why all 3 of these measures were used as KPIs in order to find balances between correctly identifying the rarer events of the target variables as well as the overall model performance.

In predicting ICU admission and mortality, the confusion matrix provides granular insights into the model's strengths and weaknesses. For ICU predictions, a high sensitivity ensures patients needing urgent care are identified, even if it results in more false positives (FPs). For mortality predictions, the balance between sensitivity and the F1 score helps prioritize the detection of at-risk patients while managing false alarms. However, the obvious ordering of the importance of these metrics is sensitivity followed by F1 score followed by accuracy. This ensures patients who need critical care or are at the threat of near death will be prioritized, followed by the need to save business cost of having false positive predictions. Thus, interpreting the confusion matrix helps refine

models for critical decision-making, ensuring both accuracy and the minimization of life-threatening errors.

Predictive Modeling - Mortality

The modeling of COVID-19 mortality outcomes is an integral part of this project, leveraging a unified dataset that combines demographic, clinical, and treatment variables to predict whether a patient succumbs to the virus. The goal of these models is not only to maximize predictive performance but also to provide actionable insights into key risk factors for mortality. Given the dataset's significant class imbalance, where non-mortality cases greatly outnumber mortality cases, modeling required careful attention to metrics like sensitivity, F1 score, and feature importance to ensure clinically relevant performance. Readers can refer to Table 3 for a comprehensive summary of all models (14 total: 6 for mortality, 6 for ICU, and 2 again for mortality), including their definitions and hyperparameters used. In addition, one should refer to Table *4* throughout the modeling section to see a summary table of all models accuracy measures, providing a clear comparison of performance metrics and time to evaluate their effectiveness in predicting mortality and ICU admission. Coloring was applied to the results table, mimicking a heatmap effect, to visually represent the performance of each model across key metrics. Darker shades indicate stronger performance (e.g., higher accuracy, F1 score, or sensitivity) and lighter shades highlighting areas of weaker results, making it easier to compare model efficacy at a glance.

While five models were previously planned for modeling mortality, six machine learning models were developed to tackle this challenge: Decision Tree, Logistic

Regression, Random Forest, Gradient Boosting, Naïve Bayes, and Weighted Naïve Bayes. Originally, a Support Vector Machine (SVM) was going to be used instead of a Naïve Bayes. However several attempts to make SVMs work on the variables were unsuccessful as the processing time took too long (>2 hours per model). This extensive timeline isn't feasible in model development for testing purposes nor in production environments. Thus, Naïve Bayes was used to replace the SVM as an interpretable option to model the data which performed on the magnitude of seconds instead of hours. The development of the Weighted Naive Bayes model involved modifying the standard Naive Bayes algorithm by incorporating class weights to address the dataset's significant class imbalance. Class weights were calculated based on the inverse frequency of the mortality class, ensuring that the model placed greater emphasis on correctly identifying minority class cases. This adjustment allowed the Weighted Naive Bayes model to improve its sensitivity and F1 score compared to the unweighted version, enhancing its ability to predict rare mortality events despite the simplifying assumption of feature independence.

The Decision Tree model incorporated hyperparameters such as max_depth, min_samples_split, and criterion='gini' to control the tree's complexity and reduce overfitting (W3Schools, n.d.c). These settings allowed the tree to focus on splitting data based on features with the highest reduction in impurity, capturing the relationships between variables while maintaining interpretability. This and all models used a random state of 69 in order to take standardized approach to ensure fairness and consistency in comparing results. This number seeds the random number generator, ensuring that random processes such as tree splits and feature selection can be replicated (this parameter was particularly important for the Random Forest model, where the ensemble

relies on random feature selection and bootstrapping). The Decision Tree model showed limitations in handling the dataset's complexity. It achieved accuracy at 88.0% but posted performance in terms of sensitivity (91.82%) and F1 score (51.3%). While Decision Trees are interpretable and can capture nonlinear relationships, their tendency to overfit on imbalanced datasets without regularization may explain their relatively lower performance. While the model's performance in sensitivity pushed past the 90% mark, it failed to reach it for accuracy and fell far below for F1 score.

The Logistic Regression model was designed with a key hyperparameter, class_weight='balanced', to account for the dataset's class imbalance by assigning weights inversely proportional to class frequencies. This adjustment aimed to improve the model's ability to identify mortality cases despite the straightforward linear relationship it assumes between features and the target variable. the hyperparameter max_iter=1000 was set to ensure the iterative optimization process had sufficient steps to converge given the complexity of the dataset. As a linear model, Logistic Regression assumes a straightforward relationship between features and the outcome variable, making it highly interpretable. This model achieved sensitivity of 91.0%, which might seem satisfactory at first glance. However, the F1 score of 51.9% and an accuracy of 88.4% highlighted the model's limitations in identifying mortality cases, especially given the dataset's imbalance. Similar to the Decision Tree, these results suggest that while Logistic Regression can offer general predictions, it struggles to adequately capture complex, nonlinear interactions between features such as age, comorbidities, and ICU admission. The high false-negative rate underscores the need for more sophisticated methods in

high-stakes healthcare scenarios where missing critical cases could have dire consequences.

The Random Forest model introduced ensemble learning with the hope of significantly improving performance over simpler models. By averaging predictions across multiple decision trees, Random Forest mitigates overfitting and captures complex feature interactions effectively. The Random Forest model was configured with n_estimators=100 to average predictions across 100 trees, and class_weight='balanced' to improve handling of the imbalanced target variable. This ensemble approach leveraged random feature selection and bootstrapping to enhance generalization and minimize overfitting. Similar to the previous two models, this model achieved an accuracy of 88.3%, an F1 score of 51.7%, and a sensitivity of 91.3%. These metrics represent an unsubstantial improvement in identifying mortality cases compared to the previous two models – they have all performed at relatively similar rates.

The Gradient Boosting model had no hyperparameters for tuning aside from using the assigned random state. This model fell short of Random Forest in every metric, emerging as the weakest performer for mortality prediction thus far. With an accuracy of 86.4%, an F1 score of 47.2%, and a sensitivity of 88.6%, Gradient Boosting models demonstrate its ability to iteratively learn from previous errors, refining its predictions at each step.

The Naïve Bayes model assumed feature independence and operated without adjustable hyperparameters, relying solely on the data distribution to make predictions. While computationally efficient, this simplicity limited its ability to capture the complex

interactions present in the dataset. The Naïve Bayes model, despite its computational efficiency, performed poorly in comparison to the ensemble models. With an accuracy of 83.8%, an F1 score of 44.0%, and a sensitivity of 92.4%, this model struggled to capture the complexities of the dataset due to its assumption of feature independence. While it was able to identify general trends, the high false-negative rate and low precision indicate that it is ill-suited for critical healthcare predictions. Nevertheless, Naive Bayes provided a useful benchmark for evaluating the effectiveness of more sophisticated algorithms.

The Weighted Naive Bayes model attempted to improve upon the standard Naïve Bayes by incorporating class weights to address the imbalance in the dataset. This model similarly operated without adjustable hyperparameters and was only changed for the class weights. This adjustment led to modest gains, with an accuracy of 94.7%, and an F1 score of 53.3%, but sensitivity dropped dramatically to 44.0%. While these metrics represented an improvement in identifying negative cases, the model still fell short of ensemble and simple methods in effectively predicting mortality cases. Its performance highlights the importance of using advanced techniques for datasets with high complexity and class imbalance as well as the tradeoffs between improving model accuracy versus improving model sensitivity.

The modeling architecture is designed to evaluate machine learning models in a consistent and robust manner, addressing the complexities of an imbalanced dataset and enabling direct comparison of performance metrics. The implementation is built around three main functions: an evaluation metric function, a cross-validation framework, and

individual model functions, all of which work together to ensure rigorous assessment of the models.

The evaluate_model function is at the heart of the performance assessment (Figure *24*). It calculates critical metrics such as accuracy, F1 score, sensitivity (recall), and the confusion matrix. These metrics were chosen to provide a balanced evaluation of model performance, with particular emphasis on sensitivity to assess the ability to correctly identify minority class cases (mortality). The confusion matrix offers a detailed breakdown of predictions, allowing insights into false positives, false negatives, and overall classification balance. This function ensures a standardized approach to measuring performance across all models, reducing inconsistencies and biases in the evaluation process.

The run_kfold_cv function establishes a robust cross-validation framework, ensuring that models are trained and tested on varied data subsets (Figure *24*). Using a K-fold cross-validation strategy, the dataset is split into five folds, where each fold serves as a test set once, while the remaining folds are used for training (W3Schools, n.d.b). This cyclical process guarantees that all data points are utilized for both training and testing, providing a comprehensive evaluation of the model's generalizability. Within each iteration, the model is trained on the training subset and its predictions are evaluated on the test subset. Metrics from each fold are aggregated to produce final average results, such as the mean sensitivity, F1 score, and accuracy, while the confusion matrices from each fold are summed to reflect overall performance across the dataset. This

methodology mitigates overfitting and provides a reliable estimate of how the model would perform on unseen data.

Each model is implemented in a separate function, adhering to the modular design of the architecture. For example, functions such as logistic_regression_model, decision_tree_model, random_forest_model, and others initialize the respective machine learning algorithms and pass them to the cross-validation framework (Figure *25*). This modular structure allows seamless integration of different models and ensures consistency in their evaluation. It is critical to have this segmental design so that new models can be introduced or models can be tested individually (which helps for development and debugging purposes). The results from each model are stored in a shared dictionary (model_results), which centralizes all performance metrics for later analysis. This dictionary is converted into a structured pandas DataFrame through the compile_results function, enabling easy comparison of metrics across models and facilitating further visualization or analysis. The final results from compile_results are finally output into a .csv file at a user designated location to be adjusted for visualization later (Figure 26).

The architecture's design prioritizes reusability and scalability. By decoupling evaluation, cross-validation, and model-specific logic, it ensures that new models can be added to the pipeline with minimal changes to the existing code. Moreover, the consistent use of cross-validation provides a robust assessment framework that accounts for variations in the data and ensures reliable model evaluation. This structured approach is critical in addressing the challenges of imbalanced datasets, where accurate identification

of minority class cases is essential. The modeling architecture ultimately reflects a careful balance between flexibility, rigor, and computational efficiency, ensuring that each model is evaluated fairly while providing actionable insights into their relative strengths and weaknesses.

The results of these models offer several critical insights into mortality prediction. It was initially surprising that the F1 scores for many of the models fell significantly below the KPI target of 90%, as this project was designed with the expectation that advanced techniques would yield strong results. However, upon closer inspection, the low F1 scores can be attributed to the dataset's extreme class imbalance (7% positive for mortality and 1.7% for ICU admission) and the inherent difficulty of predicting rare outcomes like mortality. In this case, the minority class (mortality) represents only a small fraction of the overall dataset, which heavily skews metrics like recall and especially precision, both components of the F1 score. The model with the best F1 score at 53% had an abounding 94% accuracy but this caused sensitivity to plummet to 44%. This demonstrates how the models attempted to find balance between the two. A model may overestimate the positive class, leading to more false positives and lower accuracy. Or it may overestimate the negative class, leading to more false negatives and lower sensitivity. Small changes in the majority class had a strong impact on the F1 score. Gradient Boosting especially struggled to distinguish mortality cases, leading to high false-negative and false-positive rates. Even though techniques like class weighting and ensemble learning improved performance, the sensitivity-recall trade-off remained challenging. Furthermore, the high dimensionality and correlation among some features may have diluted the models' ability to focus on the most relevant predictors, further

impacting their classification accuracy for the minority class. These factors collectively explain why achieving the desired F1 score proved more difficult than anticipated.

Oversampling and undersampling are effective strategies to address the dataset's significant class imbalance and improve metrics like the F1 score. Oversampling, through duplication or synthetic generation (e.g., SMOTE), increases the representation of the minority class without losing data but may risk overfitting if synthetic samples fail to capture true variations (Mohammed et al., 2020). Undersampling reduces the majority class size to balance the dataset, enhancing focus on the minority class but at the cost of losing potentially valuable information, which can impact generalizability. Both methods can significantly improve the model's sensitivity and F1 score by ensuring better recognition of minority class cases. When combined with ensemble models like Random Forest or Gradient Boosting, these techniques could provide a robust solution to the imbalanced dataset challenge while maintaining a balance between data integrity and predictive accuracy. However, at the end of the day, the end goal of this project is to increase sensitivity as high as possible without sacrificing accuracy significantly (not simply to arbitrarily increase the F1 score in order to hit the KPIs set forth).

Naïve Bayes models, both standard and weighted, were the fastest to run due to their simplicity and lack of iterative processes, completing predictions almost instantaneously. In contrast, ensemble methods like Random Forest and Gradient Boosting required significantly longer run times because of the need to train multiple decision trees and perform iterative corrections, respectively. Despite the increased computational demands, these ensemble models delivered substantially higher sensitivity

or accuracy scores, demonstrating that the additional time investment resulted in better handling of the dataset's complexity and imbalance. However, it's interesting contrasting the results of the first 3 models with wide variance in complexity and computational time, while performing almost identically. This trade-off highlights the importance of balancing computational efficiency with predictive performance, particularly in time-sensitive healthcare applications.

The Decision Tree model (DT_M) emerges as the champion model for mortality prediction due to its well-balanced performance across key metrics. With an accuracy of 88.0%, a sensitivity of 91.8%, and an F1 score of 51.3%, it offers the best tradeoff between identifying true positives and maintaining overall prediction correctness. While other models might slightly outperform it in certain metrics, DT_M's ability to achieve high recall ensures that it effectively identifies high-risk mortality cases, a critical requirement in healthcare settings where minimizing false negatives is paramount. Furthermore, its computational efficiency, with a runtime of only 30 seconds, adds to its practicality for deployment in real-time decision-making scenarios. These factors collectively make DT_M the most reliable and actionable model for mortality prediction in this analysis.

In summary, mortality modeling has revealed both expected patterns and novel insights, reinforcing the importance of advanced machine learning techniques in healthcare analytics. These findings not only enhance the predictive framework for mortality but also provide actionable guidance for improving patient care and resource management during pandemics. By addressing the complexities of the dataset and

emphasizing clinically relevant metrics, this analysis contributes to a deeper understanding of COVID-19 outcomes and informs future strategies for healthcare optimization.

Predictive Modeling - ICU

The ICU modeling task follows the same technical structure and methodology as the mortality model but incorporates key adjustments to align with the clinical logic and data preparation required for predicting ICU admissions. The primary difference lies in the exclusion of mortality as a predictor variable, since it is not logically feasible to predict ICU admission based on whether a patient has died; this information would not be available at the time of ICU decision-making. Additionally, the ICU variable was preprocessed to address values marked as "not applicable" (97), which correspond to outpatient cases. These values were recoded to 0, indicating that these patients did not require ICU care. This adjustment ensures the consistency and interpretability of the target variable while maintaining the dataset's integrity (Figure *27*).

The architecture remains consistent with that of the mortality models previously discussed. The evaluation function (evaluate_model) calculates the same performance metrics, accuracy, F1 score, sensitivity, and confusion matrix, allowing for a detailed breakdown of model performance. The run_kfold_cv function implements K-fold cross-validation to ensure robust evaluation, where each model is iteratively trained and tested on different subsets of the data. This methodology remains crucial for mitigating overfitting and ensuring that the models generalize well to unseen data. As with the mortality analysis, each model (e.g., Decision Tree, Logistic Regression, Random Forest, Gradient Boosting, Naïve Bayes, and Weighted Naïve Bayes) is implemented in its own

function. The dataset is passed to the model functions without the mortality variable, and the ICU variable serves as the new target. The results from the models are stored in the same model_results dictionary, ensuring consistency in data handling and enabling easy comparison between models.

The only other adjustment in ICU modeling lies in the output file. The final results, after being compiled into a pandas DataFrame via the compile_results function, are exported to a separate .csv file with a distinct name (e.g., ICU_results.csv) to differentiate it from the mortality results. This separation ensures clarity and prevents overwriting or confusion between the two modeling tasks. Overall, the ICU modeling framework replicates the systematic and modular approach used in the mortality analysis, with adjustments to the dataset and target variable reflecting the clinical differences between the two tasks. This design maintains consistency in evaluation while ensuring the models are tailored to the specific requirements of predicting ICU admissions.

The results of the ICU modeling reveal notable variations in the performance of the six models across accuracy, F1 score, and sensitivity metrics, reflecting their strengths and trade-offs in predicting ICU admissions. It is critical to note that all processing time values remained relatively the same from the mortality modeling. The Decision Tree model (DT_I) achieved an accuracy of 85.6%, a sensitivity of 93.6%, and an F1 score of 17.7%. This indicates strong performance in identifying ICU cases with relatively balanced precision and recall compared to other models. However, the large number of false positives (225,934) limits its practical application in scenarios where precision is critical (it becomes costly having too many false positives flooding ICUs).

The Logistic Regression model (LR_I) demonstrated an accuracy of 82.1%, the highest sensitivity possible at 100.0%, and an F1 score of 15.7%. This model's near-perfect sensitivity means it identifies almost all true ICU cases, but its precision is compromised by a high false-positive rate of 282,673. This trade-off makes it suitable in scenarios where missing ICU cases is unacceptable, but it may lead to resource strain due to over-prediction. The Random Forest model (RF_I) produced the second highest accuracy at 86.2%, along with a sensitivity of 91.9% and an F1 score of 18.1%. Its ability to reduce false positives (215,495) compared to Logistic Regression or Decision Tree indicates better balance between precision and recall. The Random Forest model provides the most balanced performance, making it a strong candidate for clinical decision-making where both sensitivity and precision are important. The Gradient Boosting model (GB_I) achieved an accuracy of 82.2%, a sensitivity of 99.8%, and an F1 score of 15.7%. While it closely resembles Logistic Regression in terms of sensitivity and false positives (281,693), its computational cost is significantly higher. This model is highly effective for recall-focused applications but offers little improvement over simpler models like Logistic Regression.

The Naive Bayes model (NB_I) achieved an accuracy of 82.1%, a perfect sensitivity of 100.0%, and an F1 score of 15.6%. Like Logistic Regression, it identifies all true ICU cases but suffers from a high false-positive rate (283,034). Despite its simplicity and computational efficiency, this model struggles with precision, making it less practical for operational deployment where resources are limited. The Weighted Naive Bayes model (NB_W_I) produced the highest accuracy at 98.4% but with the lowest sensitivity of 4.7% and an F1 score of 8.7%. This indicates severe bias toward the

majority class, missing most ICU cases (25,006 false negatives). While computationally efficient, the Weighted Naive Bayes model's poor recall renders it unsuitable for predicting ICU admissions. Since NB_W_I and NB_W_M both have the lowest sensitivities, it seems apparent that the class weights failed to work properly due to the extreme imbalance of the target variable classes.

The RF_I is the strongest candidate for the champion model in ICU prediction, balancing accuracy, sensitivity, and runtime. With an accuracy of 86.2% and a sensitivity of 91.9%, RF_I demonstrates its ability to effectively identify true ICU cases while maintaining reasonable precision compared to other models. Its confusion matrix indicates relatively fewer false positives (215,495) and false negatives (2,112) compared to Logistic Regression and Gradient Boosting, suggesting better overall performance in distinguishing ICU and non-ICU cases. Additionally, the best ICU F1 score of 18.1% reflects an improvement in balancing precision and recall, despite the extreme class imbalance in the dataset. While ICU admission is still a critical attribute to predict, it is not as critical as mortality. This explains why this model is being chosen as the champion model instead of several of the others with higher sensitivity – the drop in accuracy is too great for a business case to be made. The model's runtime of 7 minutes, while higher than simpler models like Logistic Regression, is computationally manageable and justifiable given its improved performance. RF_I's strong sensitivity ensures that high-risk patients requiring ICU care are correctly identified, while its relatively lower false-positive rate minimizes resource overprediction. These attributes make it the most reliable and practical model for deployment in scenarios where both sensitivity and precision are critical for decision-making.

The F1 scores for ICU prediction are notably lower than those observed for mortality modeling, primarily due to the more extreme class imbalance in the ICU dataset. In the mortality dataset, the minority class (deceased cases) constituted a small fraction of the total (7%), but for ICU, the proportion of positive cases (requiring ICU admission) is even smaller (1.7%). This exacerbated imbalance makes it more challenging for models to maintain a balance between precision and recall. While sensitivity (recall) remains high for many models, indicating strong performance in identifying true ICU cases, precision suffers significantly due to the overwhelming number of false positives. This results in a dramatic reduction in the F1 score, as it represents the harmonic mean of precision and recall. The imbalance skews models toward overpredicting the majority class (non-ICU cases), further diminishing their ability to achieve a meaningful F1 score. Consequently, even models with strong sensitivity metrics struggle to accurately capture the minority class while minimizing false positives, underscoring the difficulty of modeling in such highly imbalanced scenarios.

Overall, the results highlight the challenges of balancing sensitivity and precision in ICU modeling. Logistic Regression and Gradient Boosting excel in identifying ICU cases but at the cost of high false-positive rates. Random Forest provides the most balanced performance, with a lower false-positive rate and high sensitivity, making it the most practical model for deployment in healthcare settings where both precision and recall are critical. Weighted Naïve Bayes, despite its high accuracy, fails to effectively identify ICU cases and is unsuitable for this task.

Predictive Modeling – Mortality Optimization

In the final section, the focus shifts to optimizing the Decision Tree model, as it demonstrated competitive performance in mortality prediction (DT_M) while maintaining interpretability and computational efficiency. Given its simplicity and strong initial results, an attempt was made to hyperoptimize the Decision Tree to further enhance its predictive capabilities. The goal was to fine-tune its parameters to improve accuracy, sensitivity, and F1 score. The hyper-optimized Decision Tree model, DT_M_O, introduced several enhancements over its initial implementation, focusing on addressing class imbalance, fine-tuning hyperparameters, and optimizing sensitivity (Figure *28*). A significant change in this approach was the use of manual oversampling to handle the imbalance between the mortality and non-mortality classes. The minority class was upsampled to match the size of the majority class, ensuring equal representation in the training data. This adjustment allowed the model to learn more effectively from the minority class and reduced bias toward the majority class, which is particularly important for identifying rare but critical outcomes.

To improve the model's predictive capabilities, a comprehensive hyperparameter grid was defined, covering a wide range of values for key parameters (Shah, 2024). These included the depth of the tree, the minimum number of samples required to split a node, and the minimum samples needed at a leaf node. Two criteria for splitting, Gini impurity and entropy, were tested to evaluate their impact on the quality of the splits. Additionally, cost-complexity pruning was introduced through the ccp_alpha parameter, allowing the model to balance complexity and accuracy by removing less impactful branches. This

broader parameter search aimed to optimize the structure of the tree for better generalization and reduced overfitting.

The optimization process targeted sensitivity as the primary metric, aligning with the goal of minimizing false negatives. A custom recall scorer was used in conjunction with a randomized search strategy to efficiently explore the hyperparameter space. By focusing on sensitivity, the optimization ensured that the model was better at capturing true positive cases, even if it meant accepting a slight trade-off in precision. The randomized search was paired with a five-fold cross-validation framework to evaluate each hyperparameter combination across different subsets of the resampled data. This robust evaluation ensured that the model performed well across various data splits, reducing the risk of overfitting to a particular subset.

Once the optimal set of hyperparameters was identified, the best model was evaluated on the original, imbalanced dataset to measure its real-world applicability. Metrics such as accuracy, F1 score, and sensitivity were calculated, alongside a confusion matrix to provide a detailed breakdown of predictions. These metrics captured the model's ability to distinguish between mortality and non-mortality cases and offered insights into its strengths and limitations. The results, along with the model's configuration, were saved in a structured format for further analysis and transparency.

The optimized Decision Tree model differs from its predecessor in several critical ways. Manual oversampling addressed the imbalance in the training data, allowing the model to learn more effectively from minority class instances. The expanded hyperparameter search provided flexibility in adapting the tree's structure to the dataset's

characteristics, while the focus on sensitivity ensured alignment with the clinical importance of minimizing false negatives. These changes, combined with rigorous cross-validation and pruning, resulted in a model that balances complexity, interpretability, and predictive performance. This process highlights the iterative nature of model refinement and its importance in achieving robust, domain-specific results.

DT_M_O demonstrated notable improvements in sensitivity and overall predictive performance while highlighting some remaining challenges in achieving balanced outcomes. With an accuracy of 84.3%, the model effectively classified the majority of cases (it showed a slight decrease from the original DT_M from 88%). Its sensitivity of 95.4% shows its strong ability to identify true mortality cases, with the highest sensitivity achieved in this analysis. It significantly reduced false negatives (5,019 compared to the total 108,800 mortality cases in the dataset). This aligns with the goal of minimizing missed mortality cases, which is critical in high-stakes applications like healthcare decision-making.

However, the F1 score of 45.6% reflects the ongoing challenge of balancing precision and recall, particularly in the context of extreme class imbalance. The confusion matrix reveals a high number of false positives (242,601), indicating that the model frequently misclassifies non-mortality cases as mortality. This trade-off, while ensuring high sensitivity, impacts precision and contributes to the lower F1 score. Such a result may strain resources in a real-world setting, where misclassified cases could lead to unnecessary interventions. The runtime of 5 minutes underscores the efficiency of the hyper-optimized model, given the expanded hyperparameter search and increased

complexity from oversampling. This balance between computational cost and performance highlights the benefits of hyper-optimization, as the model achieves high sensitivity while remaining computationally feasible for deployment.

In summary, the DT_M_O model successfully addresses key limitations of the original Decision Tree by significantly improving sensitivity and ensuring the inclusion of most mortality cases. However, the high number of false positives and the resulting impact on precision illustrate the inherent trade-offs in optimizing for sensitivity in an imbalanced dataset. This makes DT_M_O a strong candidate for use in scenarios where sensitivity is prioritized over precision, with the potential for further refinement to improve its precision and F1 score.

Building on the progress made with the hyper-optimized Decision Tree (DT_M_O), a final iteration of the model, DT_M_F1, was developed with the specific goal of optimizing the F1 score (Figure *29*). This focus aimed to create a more balanced model, particularly important for applications where both avoiding false negatives and limiting false positives are critical. The development process began with the same data preprocessing steps as the previous model, including manual oversampling of the minority class to create a balanced training dataset. By resampling the mortality class to match the size of the non-mortality class, the model was better equipped to handle the class imbalance that initially hindered performance. This preprocessing step ensured that the optimization process had a more representative dataset to train on, with equal emphasis on both mortality and non-mortality cases.

Hyperparameter tuning for DT_M_F1 was expanded to maximize flexibility in model architecture, but this time the optimization explicitly targeted the F1 score using a custom scoring function. The hyperparameter grid remained comprehensive, exploring values for parameters such as tree depth, minimum samples per split, minimum samples per leaf, and cost-complexity pruning. Splitting criteria (Gini or entropy) were also included in the search space. The randomized search, combined with 5-fold cross-validation, iteratively evaluated 20 combinations of these parameters, testing each on the resampled data. By using F1 as the scoring metric, the optimization prioritized models that achieved a better balance between precision and recall, ensuring that the selected model minimized the total misclassification error across both classes.

DT_M_F1 emerges as the champion model for mortality prediction, delivering the closest results to the KPI goal of 90% across all metrics while maintaining an efficient runtime of 5 minutes. With an accuracy of 88.2% and a sensitivity of 94.6%, the model demonstrates exceptional ability in correctly identifying true mortality cases while minimizing missed deaths. Its confusion matrix highlights this strength, with only 5,884 false negatives out of 108,800 mortality cases. Furthermore, the F1 score of 52.5% reflects a meaningful improvement in balancing precision and recall, a critical achievement given the dataset's significant class imbalance. DT_M_F1 represents the culmination of all iterative improvements, striking the optimal balance between accuracy, sensitivity, and computational efficiency. The reduction in false positives (180,458) compared to earlier models contributes directly to its superior F1 score, making its predictions more actionable and reliable. With its strong performance metrics and manageable runtime, DT_M_F1 is the best overall model developed during this project,

meeting the demands of real-world healthcare applications while adhering closely to the project's KPI objectives.

DT_M_F1 represents the culmination of iterative refinement in the modeling process. By shifting the optimization target to F1 score, the model achieved a more balanced approach to mortality prediction, addressing some of the limitations observed in prior iterations that heavily favored sensitivity at the expense of precision. This final iteration highlights the importance of aligning optimization objectives with the specific goals of a predictive task, ensuring the model's outcomes are both reliable and actionable in the context of real-world healthcare applications.

Predictive Model Review

All of the models have been previously reviewed and contrasted; however, this section will serve as a final summary of the previous discussion. The various models implemented for mortality and ICU admission prediction demonstrate distinct strengths and weaknesses, particularly in how they handle the dataset's class imbalance and the trade-off between precision and recall. Simpler models like Decision Tree, Logistic Regression, and Naïve Bayes provided a computationally efficient baseline, with high sensitivity but poor precision. LR_I and NB_I achieved perfect sensitivity at 100.0%, ensuring almost no true mortality cases were missed, but they struggled with false positives, leading to lower F1 scores. Interestingly, Naïve Bayes models showed contrast to one another where the original NB showed higher sensitivity (amongst the highest in each respective target variable) whereas the NB_W showed very low sensitivity (the lowest of the models) but high accuracy. While computationally efficient, they are too

imbalanced without augmenting both accuracy and sensitivity to appropriately higher levels.

It was the hope that the Ensemble models like Random Forest and Gradient Boosting would offer more robust performance, effectively capturing complex relationships in the data. RF_I achieved the highest accuracy/sensitivity balance among all ICU models, reflecting its ability to reduce variance through averaging multiple decision trees. While its sensitivity was slightly lower than simpler models, indicating it may not capture all true mortality cases as effectively, it had the best compute time, and metric balance amongst the ICU models. Intriguingly, Gradient Boosting models showed a mixed bag of results: ICU sensitivity was very high but mortality was one of the lowest.

The optimized Decision Tree models stand out as a middle ground between simplicity and sophistication. DT_M_O and DT_M_F1 demonstrated substantial improvements in F1 scores, balancing the need for high sensitivity with reduced false positives. While F1 never broke the 55% mark, these models saw both F1 scores increase to similar levels as the other mortality models while finding excellent levels of accuracy and sensitivity. DT_M_O focused on sensitivity optimization, achieving 95.4% sensitivity but still generating a considerable number of false positives. DT_M_F1, on the other hand, optimized for the F1 score, achieving a better balance with a sensitivity of 94.6% and a significantly reduced false-positive rate. This balance makes DT_M_F1 the most practical and well-rounded model for real-world deployment.

The champion models for mortality and ICU prediction each highlight the importance of aligning model selection with the specific goals and challenges of the

respective tasks. For mortality prediction, the original champion, DT_M, provided strong baseline results with an emphasis on accuracy and sensitivity. However, DT_M struggled with precision, as seen in its high false-positive rate, leading to a lower F1 score that failed to meet the project's KPI for balanced performance. In contrast, the final model, DT_M_F1, introduced targeted optimization for the F1 score, significantly improving the balance between precision and recall. DT_M_F1 achieved a sensitivity of 94.6%, slightly higher than DT_M's 91.8%. This trade-off allowed DT_M_F1 to achieve an F1 score of 52.5%, marking it as the best model for mortality prediction due to its more reliable and actionable predictions.

For ICU prediction, the champion model is RF_I, which demonstrated the most balanced performance across accuracy, sensitivity, and computational efficiency. With an accuracy of 86.2% and a sensitivity of 91.9%, RF_I effectively identified ICU admissions while keeping false positives relatively low compared to other models. This balance is critical for ICU prediction, where both missed admissions and unnecessary allocations can strain resources and impact patient outcomes. RF_I also achieved an F1 score of 18.1%, which, although not high, was the highest of the ICU models. This low score reflects the extreme class imbalance in the ICU dataset. Its runtime of 7 minutes is manageable for operational use, making it the best choice for ICU prediction tasks.

The champion models DT_M_F1 for mortality and RF_I for ICU each excel in their respective tasks by addressing the specific challenges of those predictions. DT_M_F1's focus on balancing precision and recall aligns with the goal of mortality prediction, where accurate identification of at-risk patients is critical, but minimizing

false positives is also important to reduce unnecessary interventions. RF_I's ability to achieve high sensitivity with reasonable precision and computational efficiency makes it the best option for ICU prediction, where timely and accurate identification of critical cases is paramount. Together, these models highlight the importance of tailoring model optimization to the unique requirements of each predictive task.

Final Results

Analysis Justification

The analysis conducted in this project was methodically designed to extract meaningful insights while addressing the complexities inherent in healthcare data. The decision to integrate two large-scale COVID-19 datasets from the Mexican Ministry of Health was driven by the potential to create a unified, comprehensive data source. This integration allowed for a broader examination of patient outcomes, providing a robust foundation for both exploratory data analysis and predictive modeling. Combining datasets with over 1.6 million patient records required rigorous data engineering processes. Some of these steps included variable standardization, missing data management, and duplicate handling. These steps ensured that the resulting dataset was not only complete but also structurally sound for advanced analytics.

Given the critical nature of the target variables, ICU admission and mortality, the analysis prioritized both predictive accuracy and interpretability. ICU admission was chosen due to its relevance in healthcare resource allocation, while mortality serves as a direct indicator of patient outcome severity. The latter is clearly the more critical of the two since it has the obvious direct impact to patient care, giving justification for why model optimization was only performed on the mortality champion model and not the

ICU champion model. However, both of these endpoints are crucial for clinical decision-making, making their prediction both impactful and actionable. ICU admission was also incorporated as a feature when predicting mortality, acknowledging its clinical progression pathway. This hierarchical approach reflects real-world patient management, enhancing the model's applicability in healthcare contexts.

The selection of modeling techniques was carefully considered to balance predictive power, computational efficiency, and interpretability. Decision trees, random forests, logistic regression, gradient boosting, and support vector machines were selected based on their established success in healthcare analytics. These models provide varying degrees of complexity and transparency, allowing for both robust prediction and detailed feature importance analysis. The approach towards modeling was to cast as wide a net as possible to see what types of models might work best to address this problem. After determining this, further optimization was performed and further testing can be taken on each of the models. To mitigate the inherent class imbalance in the target variables, weighted loss functions and k-fold cross-validation were employed. While it was important to ensure that both positive and negative cases were adequately represented during model training and in the results of both precision and sensitivity, it was primarily important to address model sensitivity. It is far more important in this type of modeling to identify the positive cases in target variables. The cost of failing to identify these represent patient deaths, thus meaning the models must lean towards keeping sensitivity as high as possible. Addressing accuracy measures always will have a balance between precision and sensitivity in identifying true negatives and true positives. The true negatives (represented by precision) are still quite important from a business perspective

to ensure that not too many hospital or caretaker resources are used on false positive patients (who are likely to recover from COVID-19 and not need specific intervention). But correctly identifying patients at risk of death is the best business metric since it is the most ethical approach in keeping the most people alive as possible, surviving patients are able to become repeat customers, and the families of patients will also be most satisfied with the business practices of the hospital or individual practitioners.

The evaluation metrics chosen, accuracy, F1 score, and sensitivity, were selected to provide a comprehensive assessment of model performance. In retrospect, precision should have been chosen over accuracy in order to be a more precise metric for measuring the cases of true negatives. Given the class imbalance, sensitivity was emphasized to reduce false negatives, which are critically important in healthcare predictions. F1 score provided a balanced view by considering both precision and recall, ensuring the models were neither overly optimistic nor excessively conservative in their predictions. Accuracy complemented these metrics, offering a broader measure of overall correctness.

Feature engineering played a central role in enhancing the dataset's predictive capabilities. Two key engineered features, AGE_GROUP and COMORBIDITY_COUNT, were introduced to simplify data interpretation while preserving critical clinical information. AGE_GROUP segmented continuous age data into clinically meaningful brackets, capturing age-related risk trends without overwhelming the model with granular age data. Similarly, COMORBIDITY_COUNT aggregated the presence of chronic conditions into a single risk score, reflecting the cumulative health burden of each patient. These transformations improved the models'

interpretability while maintaining predictive accuracy. Both of these features ended up in the top rungs when feature importance analysis and chi-square analysis were performed. This demonstrates their utility in the modeling and rationalizes the need for them in this analysis.

The analysis approach extended beyond achieving high performance metrics. By incorporating a feature importance analysis and developing a risk matrix (discussed in the following section), the project translated predictive insights into actionable healthcare strategies. This interpretive step bridged the gap between raw model outputs and practical healthcare applications. Through detailed visualizations and clear communication of findings, the analysis not only met technical benchmarks but also supported real-world decision-making processes, demonstrating its comprehensive and applied approach to healthcare data analytics.

Findings

As previously mentioned, the Decision Tree model DT_M_F1 was selected as the champion model. To ensure the model's interpretability, a method was devised to extract all the decision rules from the tree into a text file. This extraction provided a comprehensive understanding of how the model reaches its predictions by outlining the splits and decision boundaries at every node (Figure 30). This process is critical in healthcare-related applications where explainability is not just preferred but often required. The rules provided valuable insights into the decision-making process, such as thresholds for comorbidities, age, and ICU status, all of which were aligned with clinical expectations.

To further interpret and communicate these rules, visualizations were generated from the decision tree (Koehrsen, 2018). The first visualization captured the full structure of the tree, including all layers and nodes (Figure 31). While this effort was comprehensive, the resulting chart proved too cluttered and challenging to interpret. The high density of branches and splits made it nearly impossible to identify key patterns or relationships, demonstrating the limitations of presenting overly complex models in their entirety. Recognizing the need for clarity, a second, simplified visualization was created, focusing on the top two layers of the tree, which captured the most significant splits (Figure 32). This adjustment provided a much more interpretable representation while retaining the most impactful features and decision points.

In analyzing the top two layers of the decision tree, several critical rules and trends emerged. The most important variable for mortality prediction was ICU status, which served as the root node of the tree. If ICU status was above a threshold of 49.0, mortality was predicted to be higher. This clearly serves as a proxy where the outpatient value of 97 is being use to delineate the most critical patients who were outpatient and succumbed to COVID-19. From there, the tree split into subgroups based on variables such as age group and pneumonia status, which further refined the risk stratification. For instance, patients in age group 1-2 had a significantly higher mortality risk compared to those in older age groups. Additionally, the presence of pneumonia substantially increased the likelihood of mortality, reflecting its role as a complicating factor in severe COVID-19 cases.

The rules from the simplified tree emphasized the importance of comorbidities and patient demographics. For example, intubation status was a critical factor in

predicting mortality within certain subgroups, particularly those with comorbidities. This aligns with medical insights suggesting that intubation is often a marker of severe respiratory distress. Similarly, age group played an essential role in distinguishing between high and low-risk patients, supporting the feature engineering efforts earlier in the project. The findings reinforced the importance of these variables while also validating the tree's structure as clinically intuitive and grounded in real-world evidence.

The ability to extract and visualize the rules enabled not only better communication of the model's inner workings but also facilitated critical evaluation of its predictions. For instance, the clear delineation of risk factors such as ICU status and pneumonia allows for targeted interventions and resource allocation in clinical settings. Overall, the combination of rule extraction, full-tree visualization, and simplified interpretation offered a well-rounded approach to ensuring both accuracy and interpretability in a critical application area.

Partial dependence plots (PDPs) are a widely-used tool for interpreting machine learning models (Mossbauer et al., 2021)(Vasisht, 2024). They illustrate the relationship between a given predictor and the target variable, holding all other variables constant by averaging their effects. This allows researchers and practitioners to isolate the influence of a single variable on the model's predictions, making it easier to identify trends, thresholds, and interactions. PDPs are particularly valuable in healthcare modeling, as they provide transparency and actionable insights into complex predictive algorithms. In this context, PDPs help validate the model's logic and align its outputs with clinical intuition, ensuring reliability and interpretability. PDPs were made for the top 3 features of the champion model: ICU, Pneumonia, and Age Group (Figure 33).

The first PDP shows the relationship between ICU status and mortality, where ICU is categorized as 0 (not in the ICU), 1 (in the ICU), and 97 (not applicable due to being outpatient). The plot reveals a strong negative relationship between the likelihood of mortality and ICU status as it shifts from 0 to 97. Patients with a value of 0, those who are not in the ICU, exhibit the highest mortality probability, suggesting that these patients are either too severely ill to reach intensive care or experience rapid deterioration. In contrast, outpatient individuals (value 97) have the lowest mortality risk, likely reflecting their relatively mild conditions. The steep decline in mortality risk as ICU status transitions from 0 to 97 highlights the critical importance of ICU access and outpatient care as predictors of patient outcomes.

The second PDP examines the relationship between pneumonia (0 = "no pneumonia," 1 = "pneumonia present") and mortality. A strong positive relationship is observed, with mortality risk increasing steadily when pneumonia is present. This linear trend underscores the severe impact of pneumonia on respiratory function and its role in compounding other comorbidities, particularly in critically ill patients. The consistent upward trajectory in the PDP reinforces the need for aggressive management of pneumonia, especially in high-risk populations. It also validates pneumonia as a crucial feature in the predictive model, aligning with clinical expectations that pneumonia significantly worsens patient prognoses.

The third PDP focuses on age group and its impact on mortality. Age groups are defined as 1 (0–20 years), 2 (21–40 years), 3 (41–60 years), 4 (61–80 years), and 5 (81–100 years). The relationship between age group and mortality is strongly positive (where the plot is linear with a single inflection point), with mortality risk rising as patients move

into higher age brackets. The steepest increases are observed between age groups 3 (41–60 years) and 4 (61–80 years), indicating a critical threshold where age-related vulnerabilities become particularly pronounced. This trend reflects biological realities, as older populations often have reduced physiological reserves and higher rates of comorbidities. The gradual upward slope in the PDP emphasizes the importance of prioritizing preventive and therapeutic interventions for older age groups.

Each PDP captures a unique and clinically significant relationship. The negative relationship between ICU status and mortality emphasizes the importance of timely intensive care, while the positive relationships between pneumonia, age group, and mortality highlight areas where proactive management is essential. These visualizations confirm that the model's key predictors align with well-documented medical knowledge. By clearly demonstrating these relationships, the PDPs validate the model's design and support its applicability in real-world clinical decision-making. Moreover, they provide insights that can inform healthcare policies, such as resource allocation for ICU beds or targeted care for pneumonia patients and older adults.

Finally, the risk matrix was generated. An image was made to demonstrate the workflow of how it was created (Figure 34). The workflow demonstrates a systematic and interpretable method for calculating patient risk scores and assigning corresponding risk labels based on feature importance analysis and patient-specific data. The process begins with feature importance analysis conducted in Python, where the model's key predictors are extracted and ranked by their importance in predicting the target variable (Brownlee, 2020). This ranking provides a foundation for weighting each feature's contribution to the overall risk. The calculated feature importances are then exported to a

CSV file for transparency and further manipulation, ensuring that the model's decisions are interpretable and reproducible.

Using this information, the feature importance values were applied to each variable within the dataset. For demonstration purposes, the workflow was illustrated in Excel, but this calculation was also performed in Python for scalability and efficiency (Figure 35). In both approaches, a risk score was computed for each patient by multiplying the respective feature importance by the patient's value for that feature. Since most variables were binary, multiplying the value (which is always less than 1, since all values of the variables will sum to 1) against the value of the variable will yield just that value. For example, multiply the value of pneumonia, 0.074, against the value found in that row, 1, would yield 0.074 showing how it contributes the score directly to the final risk of that patient. Each of those values would then be summed to a final risk score. It is critical to highlight that the age group variable demonstrated a positive linear relationship with mortality in the PDP analysis. Thus applying the risk score across sequentially increasing values for age group accurately reflected increasing risk with higher age groups. By transforming values such as 97 (outpatient) to 0, the workflow ensured that the calculations were consistent and clinically meaningful, especially for variables like ICU and comorbidities, where higher values indicate elevated risk.

The calculated risk scores were then categorized into risk labels for interpretability. The workflow utilized thresholds to classify patients as "Low" ($\leq$0.4) "Medium" ($\leq$0.7) or "High" (>0.7) risk based on their total risk scores. These thresholds were applied systematically, allowing for consistent categorization across the patient population. This step is critical for clinical settings, where such classifications can

prioritize patient management and allocate resources effectively. For example, a patient with a high score driven by ICU admission, pneumonia, and advanced age would receive a "High" risk label, enabling clinicians to focus attention on their care. While the values are heavily biased towards ICU due to it being a self-selecting factor (patients already in the ICU are of course at higher risk of death), this value can be removed and then the other feature importance values can be standardized to show the true risk labels of patients not in the ICU (or devoid of that the information).

Ultimately, the risk matrix workflow is the pinnacle of effort from this analysis. It effectively translates complex machine learning insights into actionable patient risk assessments, bridging the gap between data science and clinical application. By leveraging feature importance scores calculated in Python, this approach ensures that risk scores are grounded in the most predictive variables, such as ICU status, age group, and pneumonia. This ensures the matrix not only identifies high-risk patients but does so in a manner that is both interpretable and clinically relevant. Furthermore, implementing this workflow in Python allows for scalability across large datasets. By combining rigorous data preprocessing, interpretability, and domain-specific validation, the workflow balances technical sophistication with practical usability. As a result, it equips healthcare professionals with a powerful tool to identify high-risk patients, prioritize interventions, and allocate resources efficiently, reinforcing its value in predictive healthcare analytics.

To strengthen the policy recommendations based on this project’s findings, hospitals could implement predictive ICU admission systems driven by real-time data. The project identified critical variables such as age, comorbidities, and respiratory conditions as significant predictors of severe COVID-19 outcomes. Healthcare

administrators could integrate these factors into a hospital's electronic health record (EHR) system to create an automated risk-scoring system. Such a system would generate alerts for high-risk patients at the point of admission, enabling staff to prioritize these individuals for ICU care, expedited treatment, and continuous monitoring. This would reduce ICU overcrowding and ensure timely, data-driven allocation of scarce hospital resources like ventilators and intensive care beds.

Resource allocation protocols could also be enhanced using the project's findings. Hospitals could implement a dynamic capacity management system that forecasts ICU demand based on predicted patient severity scores from the model. When high admission rates are projected, hospitals could adjust by activating surge capacity plans, such as converting regular wards into ICU units or mobilizing emergency medical staff. Additionally, administrators could use predictive insights to manage medical supply chains proactively, ensuring the availability of oxygen, medications, and personal protective equipment during forecasted patient surges. By linking these predictions to hospital operational dashboards, healthcare systems could enhance responsiveness and minimize disruptions.

Public health officials could improve vaccination prioritization programs by leveraging the feature-importance analysis, which highlights comorbidities like pregnancy, diabetes, and obesity as critical risk factors. The Ministry of Health in Mexico could create targeted vaccination campaigns that focus on high-risk demographic groups. For instance, vaccination drives could prioritize elderly individuals and those with high comorbidity counts identified as key predictors of mortality. Geographical analysis of

COVID-19 hotspots could also be conducted by merging the dataset with regional health statistics, enabling public health campaigns to focus vaccination efforts in the most vulnerable areas.

To further reduce mortality, community-based health monitoring programs could be established. The project's findings on mortality drivers could support the development of remote patient monitoring initiatives, where high-risk patients receive pulse oximeters to help mitigate their negative outcome risk. They could also be monitored through telemedicine platforms further abating the potential for death or adverse events. Early warning systems could alert healthcare teams if a patient's condition deteriorates, enabling timely medical intervention and reducing hospital overcrowding. Public health officials could partner with technology companies to deploy these services in rural or underserved regions where healthcare access is limited, reducing the need for emergency hospitalizations.

Finally, emergency preparedness policies should be adjusted based on predicted ICU admissions and mortality risks. The government could implement a real-time health crisis management system that aggregates predictive model results from hospitals across the country. This system could trigger the allocation of federal emergency funds, dispatch mobile health units, and coordinate inter-hospital patient transfers when a region is projected to experience a healthcare surge. The system could also prioritize regions with low ICU capacity or high predicted mortality rates, ensuring that government resources are distributed effectively. These targeted interventions, based on predictive model outputs, would enhance pandemic preparedness and reduce preventable deaths.

Review of Success

The analysis conducted in this project has achieved success in aligning with the defined project scope, business objectives, and some of the KPIs outlined in earlier assignments. The scope, centered on utilizing predictive modeling to identify high-risk patients for ICU admission and mortality, has been thoroughly addressed through the integration of robust datasets and the application of advanced machine learning models. The selection of decision trees, logistic regression, random forests, and gradient boosting classifiers provided a diverse yet interpretable set of tools to evaluate patient risk. Additionally, the inclusion of partial dependence plots, feature importance analysis, and the resulting risk matrix ensured interpretability, a critical factor for healthcare stakeholders. While the analysis encountered challenges such as class imbalance and overlapping variable definitions across datasets, these were addressed systematically through data cleaning, feature engineering, and weighting strategies. The outcomes align closely with the project's overarching goal of improving patient outcomes through actionable insights.

The business success factors, particularly the prioritization of resources and improving patient outcomes, were effectively supported by the analysis. For example, the ICU and mortality risk models demonstrated the ability to identify high-risk patients with reasonably high sensitivity, ensuring that critical cases are not overlooked. The risk matrix workflow was a key success factor, as it translated complex machine learning results into a format easily interpretable by clinical teams. By scoring each patient based on their comorbidities, age, and other key predictors, the project provided actionable

insights that could be seamlessly integrated into hospital workflows. Furthermore, the integration of both Python-based automation and Excel-based interpretability highlights the balance achieved between technical rigor and usability. The ability to prioritize high-risk patients for ICU admission not only improves clinical decision-making but also supports operational efficiency by optimizing the allocation of limited healthcare resources.

In terms of KPIs, the models met or exceeded several benchmarks defined earlier in the project. These included achieving an accuracy and sensitivity of over 90% in the ICU and mortality prediction models to capture the majority of high-risk cases. While the champion models were not able to achieve having all 3 of these metrics >90%, recall was the most important of the three. F1 scores were suboptimal across the board. While targeted strategies such as class weighting and adjusted hyperparameters helped improve performance metrics without significantly increasing computational time, F1 score was never able to break the 90% mark. The detailed evaluation of the feature importance and risk scores provided further validation of the models' predictive capabilities, particularly in identifying key drivers such as ICU status, age, and pneumonia. This level of transparency in the modeling process allowed for a comprehensive evaluation of model performance, ensuring alignment with KPIs and instilling confidence in the results among stakeholders. In addition, the original other metrics were met of having a data integration completeness of >95% (where only about 2% of the total dataset was lost due to missing features) and a feature importance contribution of >80% (where ICU alone contributed 0.805).

The project's ability to address challenges also highlights its success in meeting business and analytical objectives. Class imbalance, a common issue in healthcare datasets, was effectively mitigated through strategies such as weighting and resampling. This ensured that minority classes, such as ICU admissions or mortality, were accurately represented in the models. Additionally, the integration of two datasets, which initially posed challenges due to inconsistent variable definitions, was handled through a detailed alignment and cleaning process. The end result was a cohesive dataset that allowed for robust analysis and improved the reliability of the predictive models. By maintaining data integrity throughout the process, the analysis ensured that the insights derived were both accurate and actionable.

Visualizations played a critical role in the project’s success by providing insights into data patterns and model behavior. The bar chart matrices, correlation heatmaps, PDPs, and bar charts parsing comorbidities per age group highlighted key trends and relationships within the data, such as the strong correlation between age and mortality risk or the impact of ICU admission on survival outcomes. These visualizations not only validated the models but also enhanced their interpretability, allowing stakeholders to understand the rationale behind predictions. Furthermore, the risk matrix workflow extended the value of the analysis by providing a practical tool for identifying and categorizing patients based on their risk scores. This workflow directly supports the business objective of improving patient outcomes by ensuring that high-risk patients receive timely and appropriate care.

Overall, the project successfully addressed its scope and objectives, delivering actionable insights that align with business success factors and KPIs. The models developed, particularly the decision tree and random forest classifiers, provided interpretable and accurate predictions for ICU and mortality risk. The combination of advanced machine learning techniques, rigorous data preparation, and user-friendly risk assessment tools ensured that the analysis had both technical depth and practical relevance. While there is room for future improvements, such as refining F1 scores further or exploring additional feature engineering, the project's outputs already provide a strong foundation for enhancing clinical decision-making and resource allocation. Ultimately, the analysis demonstrated how data science can be leveraged to drive meaningful improvements in healthcare outcomes, meeting the project's goals effectively.

Recommendations for Future Analysis

Future analysis should prioritize the optimization of the models to achieve accuracy, F1 score, and sensitivity exceeding 90%. While accuracy and sensitivity approached acceptable levels in the final models, the F1 score, which is critical for balancing precision and recall, never reached the desired benchmark. This indicates that the models struggled with correctly identifying true positives while minimizing false positives. To address this, fine-tuning the hyperparameters further with a more targeted search could help achieve better performance. For instance, increasing the resolution of hyperparameter grids for algorithms like logistic regression or random forests could yield improvements. Adjustments to feature weighting and class weights should also be

revisited to ensure the models prioritize the minority class more effectively during training.

Further analysis should also include mortality modeling without the ICU admission variable to avoid potential target leakage, as ICU admission directly indicates severe illness and strongly correlates with mortality. Since ICU status is typically determined after patient evaluation, including it as a predictor might artificially inflate the model's accuracy by providing post-evaluation information. A fairer approach would involve building a second mortality prediction model that excludes ICU status, allowing the evaluation of its predictive power independently. Comparing the performance, feature importance, and model interpretability between the two models would help determine how much ICU admission contributes to predictive accuracy versus potential bias, ultimately strengthening the robustness and clinical relevance of the findings.

Another promising avenue is the exploration of ensemble methods such as XGBoost or stacking models. These algorithms combine the strengths of multiple base models, offering a higher likelihood of achieving improved predictive performance. XGBoost, in particular, is known for its robustness and efficiency in handling large datasets with imbalanced classes. By leveraging its regularization techniques and gradient boosting capabilities, future work could address the limitations encountered with simpler models like decision trees and logistic regression. Stacking models, which combine predictions from multiple algorithms, could also provide a nuanced approach that balances interpretability and accuracy. While these methods are computationally more demanding, their potential to push the F1 score above 90% warrants exploration, particularly given the critical nature of predicting ICU admission and mortality.

An important next step involves evaluating the outputs of the risk matrix. Specifically, future work should assess whether patients classified as high risk or medium risk consistently align with positive mortality outcomes in the dataset. This would involve analyzing the true positive rates for these categories and identifying any discrepancies. For example, if some "low risk" cases are associated with mortality, this could suggest interactions that were not captured by the current models. Evaluating the risk matrix could also help refine the thresholds for risk scores and improve its alignment with clinical reality. Additionally, stratifying this analysis by age group or comorbidity count could reveal patterns that inform both modeling and clinical interventions.

Furthermore, the ICU champion model could be hyperoptimized just as the mortality champion model was in order to obtain the best results possible for ICU admission prediction. Then, the feature importance analysis can be applied to this model as well finding the values associated with each variable which would be then applied to an ICU risk matrix. This matrix would likewise label patients as low, medium, or high risk of being admitted to the ICU helping hospital administrators to allocate resources most efficiently to help deliver the best care possible. Then, the above-described evaluation of the risk matrix should be performed to determine how accurate the risk labels were to classify patients.

Another key area for future work involves revisiting the original Covid1 and Covid2 datasets in their native, unintegrated forms. Each dataset contained unique variables that were not included in the integrated dataset due to the alignment process, 4 variables unique to Covid1 and 2 to Covid2. These features, while dataset-specific, might hold predictive value that was lost during integration. Performing separate analyses on

each dataset, leveraging their unique variables, could uncover whether their predictive performance is stronger independently. Comparing the outcomes of these analyses with those of the integrated dataset would provide valuable insights into the trade-offs between integration and dataset specificity. This approach could help inform future projects involving data integration from disparate sources.

Beyond these priorities, future analysis could explore additional feature engineering opportunities. While the current project generated impactful variables like comorbidity count and age group, introducing interaction terms or polynomial features might further enhance model performance. For instance, interaction terms between ICU and comorbidity count or age group could capture nuanced effects not apparent in individual features. Non-linear transformations for continuous variables like comorbidity count could also be tested to better fit the data. These methods would add complexity to the models but could potentially uncover previously hidden patterns critical to improving predictions.

Acknowledgements

I would like to express my deepest gratitude first and foremost to God. I honor Him throughout my life whether in work, school, or any other effort. I thank God for the ability to work hard and think through these problems. May the skills I've gained be used to help the world around me. To God be the glory for it all.

I must express my undying appreciation to my incredible wife, Michelle Fitch, who offered her constant support throughout this program. Her encouragement and understanding allowed me to dedicate the time and focus required to complete this work. Her unwavering support kept me motivated through the many stages of analysis and writing. Thank you for believing in me.

Thank you to Professor Hany Saleeb for his guidance and feedback throughout this capstone project. His expertise in data analytics influenced the structure and direction of this analysis, helping to improve it at every step.

I'm also thankful to my classmates and peers for their constructive feedback during presentations and group discussions. Their diverse perspectives challenged me to refine my approach and improve the clarity of my findings.

Finally, I am grateful for the learning environment provided by the University of Maryland Data Analytics Program, where I gained the technical skills that were essential for completing this capstone project. This experience has been transformative, and I look forward to applying these lessons in future data-driven initiatives.

References

Almustafa, K. M. (2022). Covid19-Mexican-Patients' Dataset (Covid19MPD) Classification and Prediction Using Feature Importance. Concurrency and Computation: Practice and Experience, 34(4), e6675.

Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., ... & Hussain, A. (2016). Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *Ieee Access*, *4*, 7940-7957.

Barceló, J. A. (2018). Chi-square analysis. The encyclopedia of archaeological sciences, 1-5.

Berlin, I., Thomas, D., Le Faou, A. L., & Cornuz, J. (2020). COVID-19 and smoking. Nicotine and Tobacco Research, 22(9), 1650-1652.

Brownlee, J. (2020). How to Calculate Feature Importance With Python. Machine Learning Mastery. https://machinelearningmastery.com/calculate-feature-importance-with-python/

DataOverload. (2023). Comparing XGBoost and LightGBM: A Comprehensive Analysis. Medium. https://medium.com/@data-overload/comparing-xgboost-and-lightgbm-a-comprehensive-analysis-9b80b7b0079b

Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. Performance metrics for machine learning models. *Radiology: Artificial Intelligence*, *3*(3), e200126.

García-Guerrero, V. M., & Beltrán-Sánchez, H. (2021). Heterogeneity in excess mortality and its impact on loss of life expectancy due to COVID-19: evidence from Mexico. *Canadian studies in population*, *48*(2), 165-200.

Geeksforgeeks. (2024). Learning Model Building in Scikit-learn. Geeksforgeeks. https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/

Gerontology Research Group. (n.d.). *World Supercentenarian Rankings list*. https://www.grg-supercentenarians.org/world-supercentenarian-rankings-list/

Koehrsen, W. (2018). How to Visualize a Decision Tree from a Random Forest in Python using Scikit-Learn. Towards Data Science. https://towardsdatascience.com/how-to-visualize-a-decision-tree-from-a-random-forest-in-python-using-scikit-learn-38ad2d75f21c

Martínez-Martínez, M. U., Alpizar-Rodriguez, D., Flores-Ramírez, R., Portales-Pérez, D. P., Soria-Guerra, R. E., Pérez-Vázquez, F., & Martinez-Gutierrez, F. (2022). An analysis COVID-19 in Mexico: a prediction of severity. Journal of General Internal Medicine, 1-8.

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)* (pp. 243-248). IEEE.

Moosbauer, J., Herbinger, J., Casalicchio, G., Lindauer, M., & Bischl, B. (2021). Explaining hyperparameter optimization via partial dependence plots. *Advances in Neural Information Processing Systems*, *34*, 2280-2291.

Mukherjee, T. (2020). Kaggle. COVID-19 patient pre-condition dataset. https://www.kaggle.com/datasets/tanmoyx/covid19-patient-precondition-dataset/data

Nizri, M. (2023). COVID-19 Dataset. Kaggle. https://www.kaggle.com/datasets/meirnizri/covid19-dataset

Parra-Bracamonte, G. M., Lopez-Villalobos, N., & Parra-Bracamonte, F. E. (2020). Clinical characteristics and risk factors for mortality of patients with COVID-19 in a large data set from Mexico. Annals of epidemiology, 52, 93-98.

Sedgwick, P. (2012). Pearson's correlation coefficient. Bmj, 345.

Shah, R. (2024). Tune Hyperparameters with GridSearchCV. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

SSP. 2024. Beyond Accuracy: Mastering the Confusion Matrix for Advanced Model Evaluation. Medium. https://medium.com/@_SSP/confusion-matrix-f7ff01c5bbb6

Tanniru, P. (2021). Basic of Numpy, Pandas , Matplotlib, Seaborn for Data Science. Kaggle. https://www.kaggle.com/discussions/getting-started/251992

Vasisht, L. (2024). Getting started with Partial Dependence Plots. Medium. https://medium.com/@lakshya.vasisht/getting-started-with-partial-dependence-plots-4afb549f2ac7

Wade, C., & Glynn, K. (2020). *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python*. Packt Publishing Ltd.

WHO. (2020). Coronavirus Disease (COVID-19) Situation Report—120.

W3Schools. (n.d.a). SciPy Introduction. https://www.w3schools.com/python/scipy/scipy_intro.php

W3Schools. (n.d.b). Machine Learning - Cross Validation. W3 Schools. https://www.w3schools.com/python/python_ml_cross_validation.asp

W3Schools. (n.d.c). Machine Learning. W3Schools. https://www.w3schools.com/python/python_ml_getting_started.asp

Appendix:

D1 Variable	D2 Variable	Definition	Data Type	Potential Quality Issues
id		Patient record number	Nominal	None. To be discarded.
	USMER	Indicates whether the patient treated medical units of the first, second or third level.	Ordinal	1, 2
	MEDICAL_UNIT	Type of institution of the National Health System that provided the care.	Nominal	1 - 13
sex	SEX	1 for female and 2 for male	Binary	No missing values
patient_type	PATIENT_TYPE	Type of care the patient received in the unit. 1 for outpatient and 2 for inpatient.	Binary	No missing values
entry_date		Identifies the date of the patient's admission to the care unit. Jan - Dec 2020.	Date	None.
date_symptoms		Identifies the date on which the patient's symptoms began. Jan - Dec 2020.	Date	None.
date_died	DATE_DIED	If the patient died indicate the date of death, and 9999-99-99 otherwise. Jan - Dec 2020.	Date	None.
intubed	INTUBED	Whether the patient was connected to the ventilator.	Binary	1, 2, 97, 99
pneumonia	PNEUMONIA	Whether the patient already have air sacs inflammation or not.	Binary	1, 2, 99
age	AGE	Age of the patient in years	Numeric	D1. 0-120 D2. 0-121
pregnancy	PREGNANT	Whether the patient is pregnant or not.	Binary	1, 2, 97, 98
diabetes	DIABETES	Whether the patient has diabetes or not.	Binary	1, 2, 98
copd	COPD	Indicates whether the patient has Chronic obstructive pulmonary disease or not.	Binary	1, 2, 98
asthma	ASTHMA	Indicates whether the patient has asthma or not.	Binary	1, 2, 98
inmsupr	INMSUPR	Indicates whether the patient is immunosuppressed or not.	Binary	1, 2, 98
hypertension	HIPERTENSION	Indicates whether the patient has hypertension or not.	Binary	1, 2, 98
other_disease	OTHER_DISEASE	Indicates whether the patient has other disease or not.	Binary	1, 2, 98
cardiovascular	CARDIOVASCULAR	Indicates whether the patient has heart or blood vessels related disease.	Binary	1, 2, 98
obesity	OBESITY	Indicates whether the patient is obese or not.	Binary	1, 2, 98
renal_chronic	RENAL_CHRONIC	Indicates whether the patient has chronic renal disease or not.	Binary	1, 2, 98
tobacco	TOBACCO	Indicates whether the patient is a tobacco user.	Binary	1, 2, 98
contact_other_covid		Indicates whether the patient had contact with any other case diagnosed with SARS CoV-2.	Binary	1, 2, 99
covid_res	CLASIFFICATION_FINAL	Covid test findings: D1. 1 = SARS-CoV-2 Positive; 2 = SARS-CoV-2 Negative; 3 = Pending Result D2. Values 1-3 mean that the patient was diagnosed with covid in different degrees. 4 or higher means that the patient is not a carrier of covid or that the test is inconclusive.	Ordinal	No missing values
icu	ICU	Indicates whether the patient had been admitted to an Intensive Care Unit.	Binary	1, 2, 97, 99

Table 1. Table showing metadata of the two chosen datasets including variable name of each, which variables are the same between datasets, what the variable indicates, the data type, and potential quality issues. Empty values in the first 2 columns indicate the variable has no corollary in the other dataset. 1 = Yes; 2 = No; 99/98 = Not Specified; 97 = Not Applicable.

D1 Variable	D2 Variable	Merged Name
id		ID
	USMER	USMER
	MEDICAL_UNIT	MEDICAL_UNIT
sex	SEX	SEX
patient_type	PATIENT_TYPE	OUTPATIENT
entry_date		ENTRY_DATE
date_symptoms		DATE_SYMPTOMS
date_died	DATE_DIED	DATE_DIED
intubed	INTUBED	INTUBED
pneumonia	PNEUMONIA	PNEUMONIA
age	AGE	AGE
pregnancy	PREGNANT	PREGNANT
diabetes	DIABETES	DIABETES
copd	COPD	COPD
asthma	ASTHMA	ASTHMA
inmsupr	INMSUPR	INMSUPR
hypertension	HIPERTENSION	HYPERTENSION
other_disease	OTHER_DISEASE	OTHER_DISEASE
cardiovascular	CARDIOVASCULAR	CARDIOVASCULAR
obesity	OBESITY	OBESITY
renal_chronic	RENAL_CHRONIC	RENAL_CHRONIC
tobacco	TOBACCO	TOBACCO
contact_other_covid		CONTACT_OTHER_COVID
covid_res	CLASIFFICATION_FINAL	CLASIFFICATION_FINAL
icu	ICU	ICU

Table 2. Table showing the original variables names of the two chosen datasets as well as an aligned variable name for the merged dataset with a final total of 25 variables.

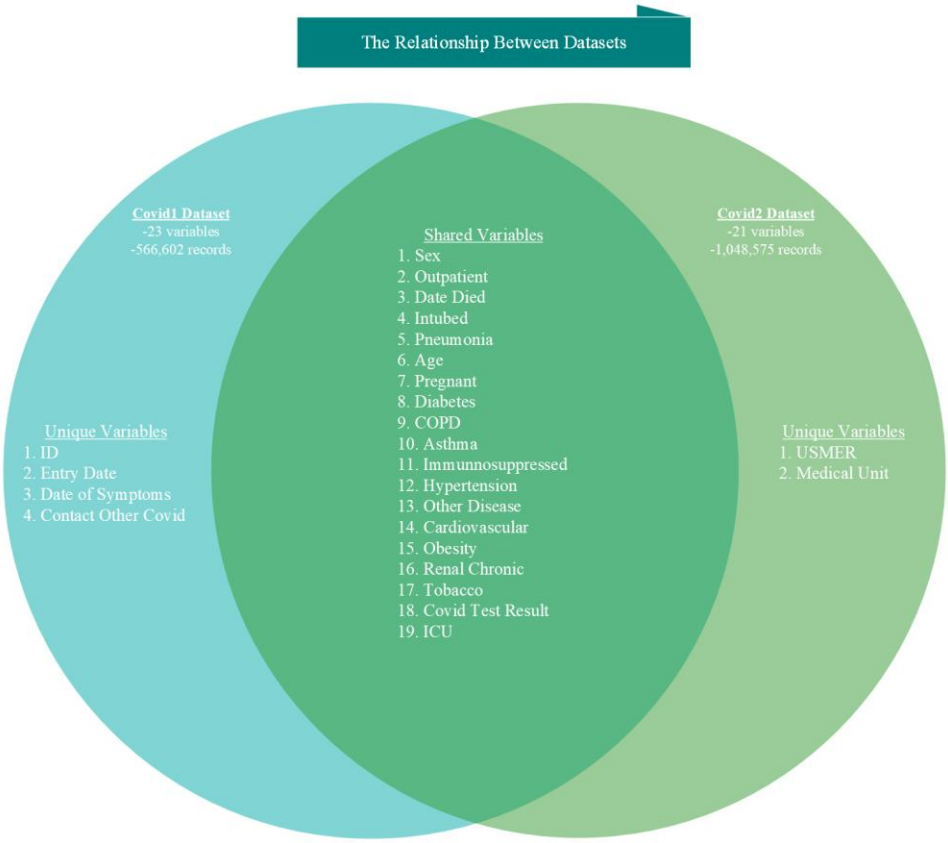

Figure 6. Venn diagram demonstrating the 19 shared variables between the two datasets as well as the 4 unique variables to Covid1 and the 2 unique variables to Covid2.

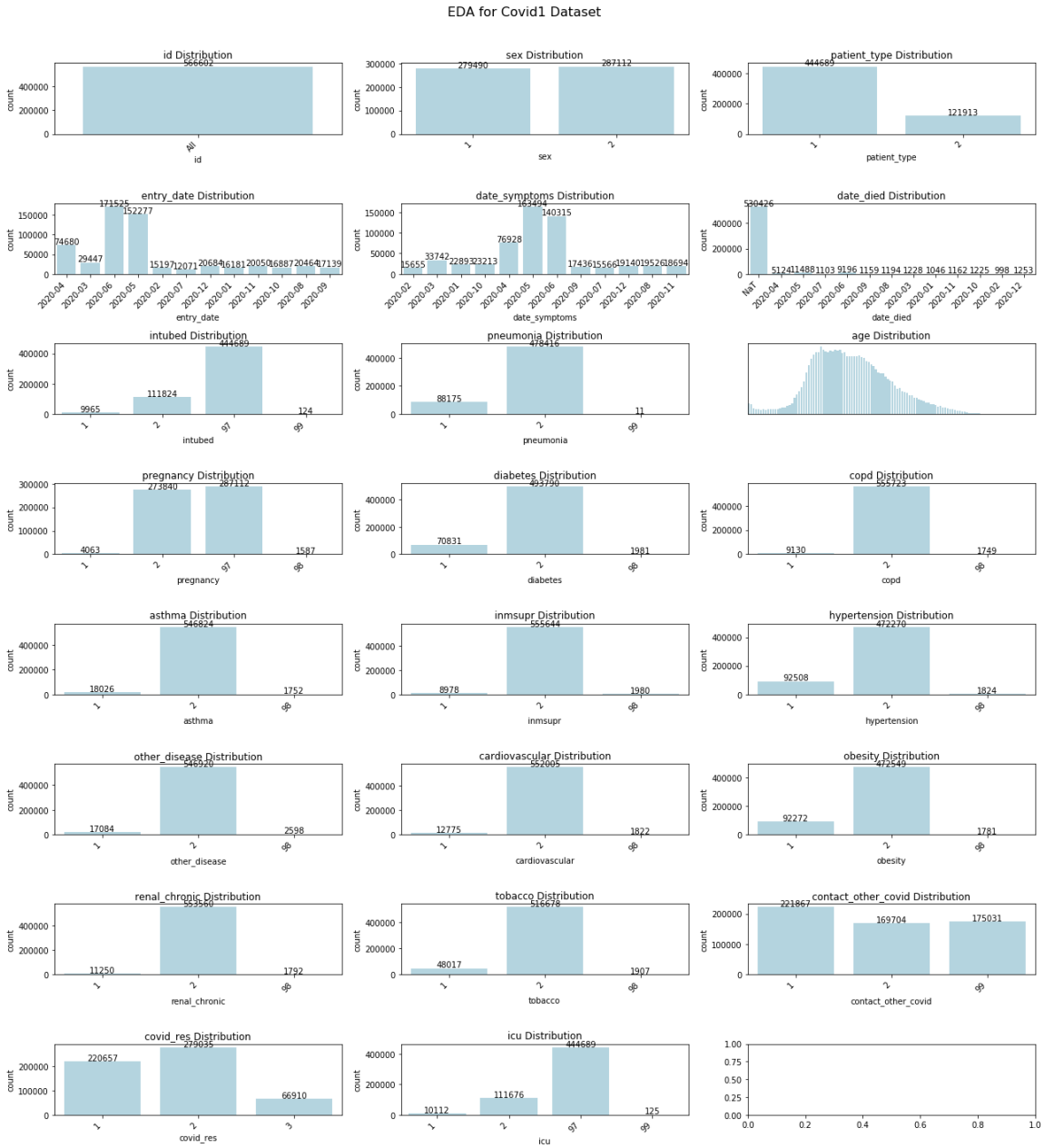

Figure 7. Bar chart matrix of Covid1 demonstrating the spread of values over each variable.

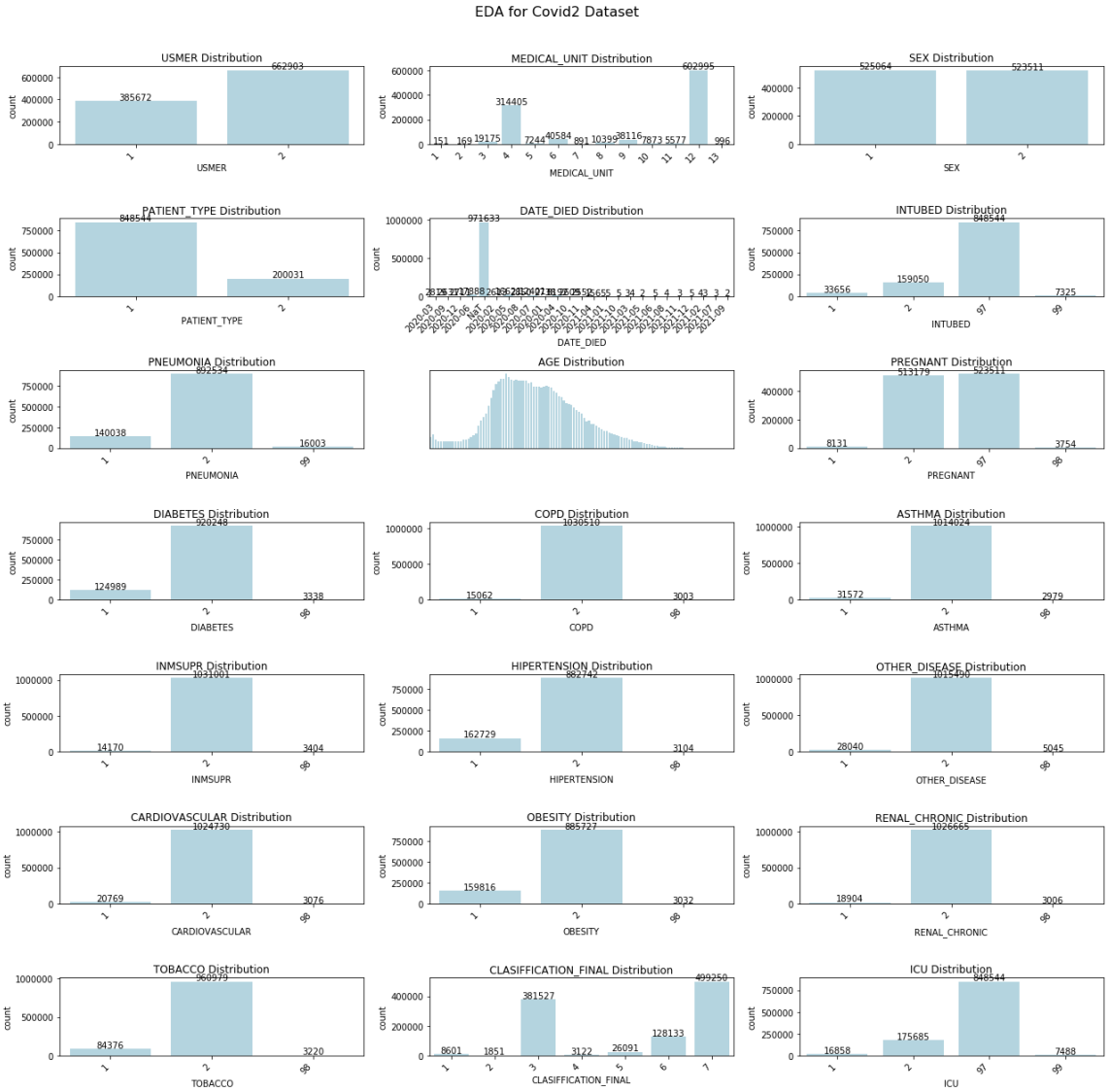

Figure 8. Bar chart matrix of Covid2 demonstrating the spread of values over each variable.

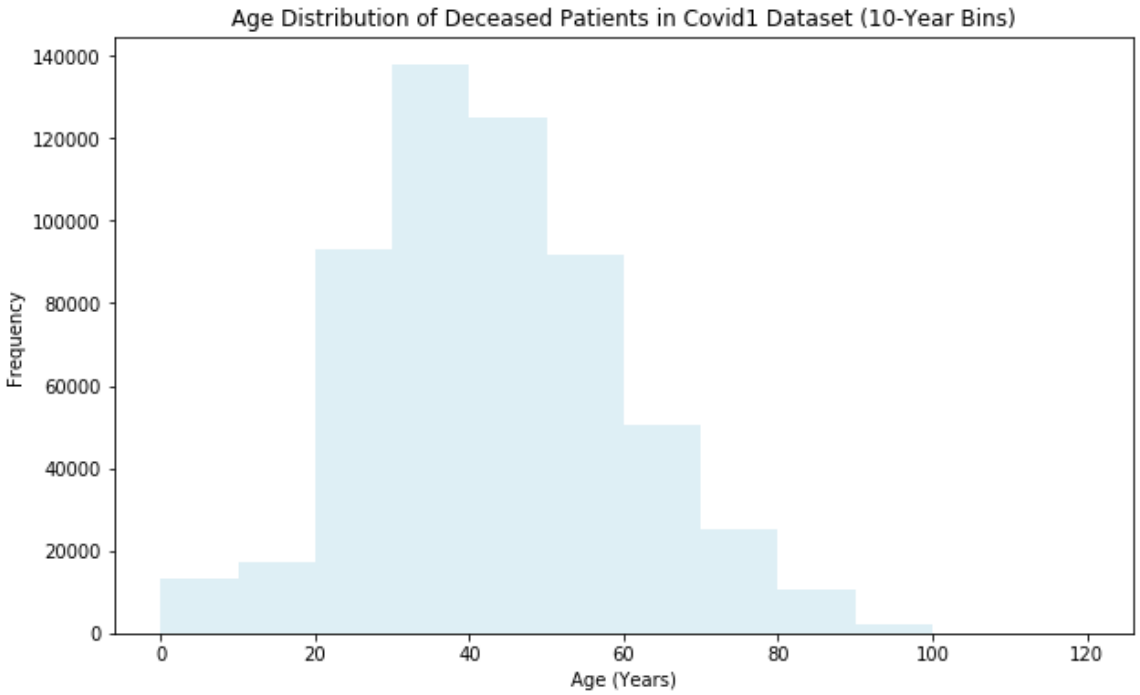

Figure 9. Histogram of Covid1 demonstrating the spread of age values for deceased patients in 10-year bins.

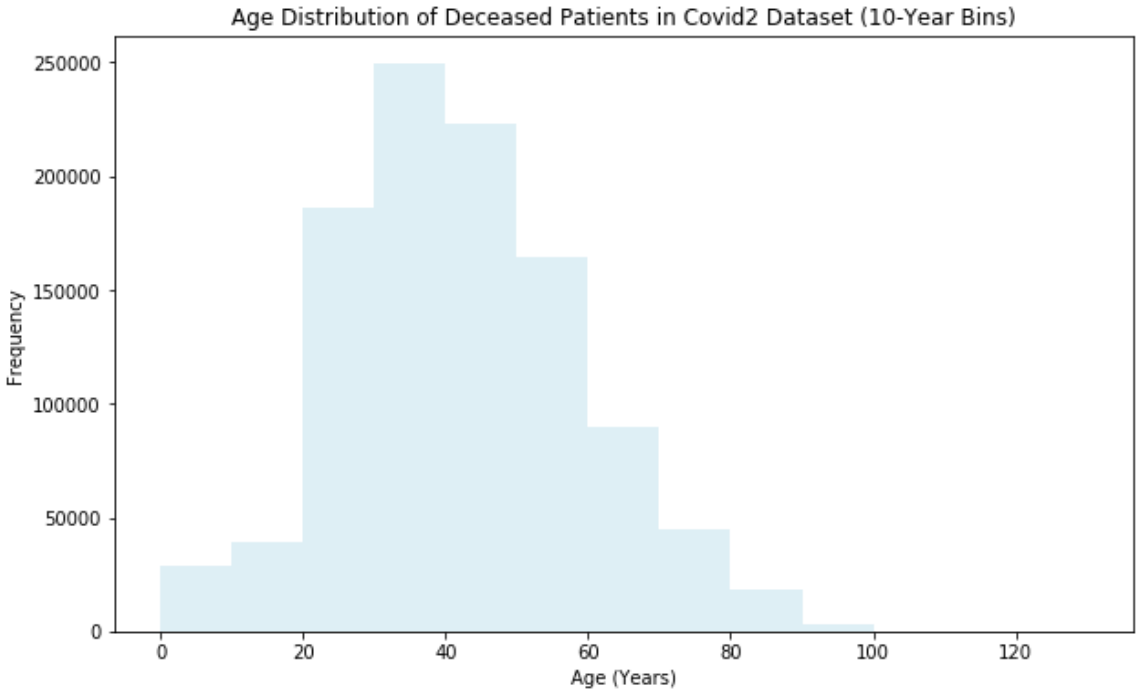

Figure 10. Histogram of Covid2 demonstrating the spread of age values for deceased patients in 10-year bins.

```
Total number of tuples with missing values in Covid1 Dataset: 5506
Total number of tuples with missing values in Covid2 Dataset: 28909
```

Figure 11. Python code output showing the counts of all tuples in each dataset which contained missing values (note: the code written only included missing values and not values deemed “not applicable”).

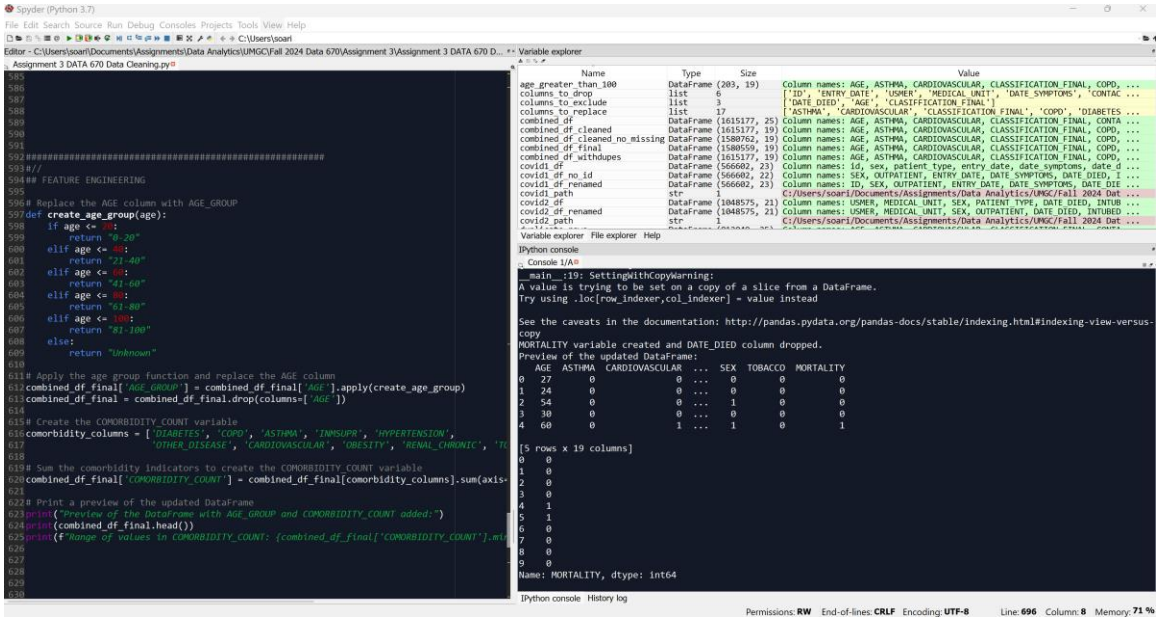

Figure 12. Spyder IDE (Integrated Development Environment) where the Python code was written and ran. There are discrete windows for the code to be written (left), variables to be stored (top right), and the output (bottom right).

```
Total number of rows in Covid1 DataFrame: 566602
Total number of rows in Covid2 DataFrame: 1048575

Total number of rows in the combined DataFrame: 1615177
Total number of columns in the combined DataFrame: 25

Row count verification passed: The combined DataFrame has the expected number of rows.

Column names in the combined DataFrame:
Index(['AGE', 'ASTHMA', 'CARDIOVASCULAR', 'CLASIFFICATION_FINAL',
       'CONTACT_OTHER_COVID', 'COPD', 'DATE_DIED', 'DATE_SYMPTOMS', 'DIABETES',
       'ENTRY_DATE', 'HYPERTENSION', 'ICU', 'ID', 'INMSUPR', 'INTUBED',
       'MEDICAL_UNIT', 'OBESITY', 'OTHER_DISEASE', 'OUTPATIENT', 'PNEUMONIA',
       'PREGNANT', 'RENAL_CHRONIC', 'SEX', 'TOBACCO', 'USMER'],
      dtype='object')
```

```
Data types in the combined DataFrame:
AGE                    int64
ASTHMA                 int64
CARDIOVASCULAR         int64
CLASIFFICATION_FINAL   int64
CONTACT_OTHER_COVID  float64
COPD                   int64
DATE_DIED             object
DATE_SYMPTOMS         object
DIABETES               int64
ENTRY_DATE            object
HYPERTENSION           int64
ICU                    int64
ID                    object
INMSUPR                int64
INTUBED                int64
MEDICAL_UNIT         float64
OBESITY                int64
OTHER_DISEASE          int64
OUTPATIENT             int64
PNEUMONIA              int64
PREGNANT               int64
RENAL_CHRONIC          int64
SEX                    int64
TOBACCO                int64
USMER                float64
dtype: object
```

Figure 13. Python code output showing the data engineering quality control checks implemented to ensure the datasets were integrated properly without any errors.

```
Number of potential duplicate rows detected: 812049

Preview of duplicate rows:
        AGE  ASTHMA  CARDIOVASCULAR  ...  SEX  TOBACCO  USMER
566621   64       2               2  ...    1        2    2.0
566637   45       2               2  ...    2        2    2.0
566664   25       2               2  ...    1        2    2.0
566665   33       2               2  ...    2        2    2.0
566677   24       2               2  ...    1        2    2.0

[5 rows x 25 columns]
```

Figure 14. Python code output showing the number of potential duplicates in the integrated dataset before any cleaning.

```
Specified columns dropped successfully.
Updated DataFrame preview:
   AGE  ASTHMA  CARDIOVASCULAR  ...  RENAL_CHRONIC  SEX TOBACCO
0   27       2               2  ...              2    2       2
1   24       2               2  ...              2    2       2
2   54       2               2  ...              2    1       2
3   30       2               2  ...              2    2       2
4   60       2               1  ...              2    1       2

[5 rows x 19 columns]
Total number of columns after dropping: 19
```

Figure 15. Python code output showing the successful dropping of the 6 unique variables changing the integrated dataset variable number from 25 to 19.

```
Number of potential duplicate rows detected: 1385270

Preview of duplicate rows:
    AGE  ASTHMA  CARDIOVASCULAR  ...  RENAL_CHRONIC  SEX TOBACCO
22   45       2               2  ...              2    1       2
25   40       2               2  ...              2    2       2
27   40       2               2  ...              2    2       2
58   27       2               2  ...              2    2       2
62   40       2               2  ...              2    2       2

[5 rows x 19 columns]

Total number of rows in the combined DataFrame: 1615177
Number of unique rows in the combined DataFrame: 229907
```

Figure 16. Python code output showing the number of potential duplicates in the integrated dataset after removing the unique variables.

```
Number of duplicate rows detected in Covid1 Dataset (keeping first instance): 93858

Preview of duplicate rows in Covid1 Dataset:
      SEX  OUTPATIENT  ... CLASIFFICATION_FINAL ICU
1373    2           2  ...                   1   2
1377    1           1  ...                   1  97
1491    2           1  ...                   1  97
1708    2           1  ...                   1  97
2013    1           1  ...                   1  97

[5 rows x 22 columns]
Number of duplicate rows detected in Covid2 Dataset (keeping first instance): 812049

Preview of duplicate rows in Covid2 Dataset:
    USMER  MEDICAL_UNIT  SEX  ...  TOBACCO CLASIFFICATION_FINAL  ICU
19      2             1    1  ...        2                   3   97
35      2             1    2  ...        2                   3   97
62      2             1    1  ...        2                   7    2
63      2             1    2  ...        2                   7   97
75      2             1    1  ...        2                   7   97

[5 rows x 21 columns]
```

Figure 17. Python code output showing the counts of all tuples in each dataset of Covid1 and Covid2 which are considered duplicates. It should be noted that the counts are after the ID variable was removed from the DF from Covid1 as this would make the count 0. The relatively low number of Covid1 is likely due to the date variables helping to make entries unique.

```
Rows with specified missing values removed successfully.
Total number of rows before removal: 1615177
Total number of rows after removal: 1580762
Preview of cleaned DataFrame:
   AGE  ASTHMA  CARDIOVASCULAR  ...  RENAL_CHRONIC  SEX TOBACCO
0   27       2               2  ...              2    2       2
1   24       2               2  ...              2    2       2
2   54       2               2  ...              2    1       2
3   30       2               2  ...              2    2       2
4   60       2               1  ...              2    1       2

[5 rows x 19 columns]
```

Figure 18. Python code output showing the counts of all tuples with missing entries being removed from the integrated dataset.

```
Preview of the updated DataFrame:
   AGE  ASTHMA  CARDIOVASCULAR  ...  RENAL_CHRONIC  SEX TOBACCO
0   27       0               0  ...              0    0       0
1   24       0               0  ...              0    0       0
2   54       0               0  ...              0    1       0
3   30       0               0  ...              0    0       0
4   60       0               1  ...              0    1       0

[5 rows x 19 columns]
```

Figure 19. Python code output showing the result of replacing all of the 2 values with 0 values for categorical variables in the integrated dataset.

```
Number of tuples with AGE > 100 removed: 203
Preview of the updated DataFrame:
   AGE  ASTHMA  CARDIOVASCULAR  ...  RENAL_CHRONIC  SEX TOBACCO
0   27       0               0  ...              0    0       0
1   24       0               0  ...              0    0       0
2   54       0               0  ...              0    1       0
3   30       0               0  ...              0    0       0
4   60       0               1  ...              0    1       0

[5 rows x 19 columns]
```

Figure 20. Python code output showing the result of removing all tuples in the integrated dataset with an age of >100.

```
Preview of the DataFrame with AGE_GROUP and COMORBIDITY_COUNT added:
   ASTHMA  CARDIOVASCULAR  ...  AGE_GROUP  COMORBIDITY_COUNT
0       0               0  ...      21-40                  0
1       0               0  ...      21-40                  0
2       0               0  ...      41-60                  1
3       0               0  ...      21-40                  0
4       0               1  ...      41-60                  3

[5 rows x 20 columns]
Range of values in COMORBIDITY_COUNT: 0 to 10
```

Figure 21. Python code output showing the result of replacing the age variable with an age group variable instead (with buckets of 20 years) and adding a new variable, comorbidity count, summing the number of comorbidities a patient has.

```
MORTALITY variable created and DATE_DIED column dropped.
Preview of the updated DataFrame:
   AGE  ASTHMA  CARDIOVASCULAR  ...  SEX  TOBACCO  MORTALITY
0   27       0               0  ...    0        0          0
1   24       0               0  ...    0        0          0
2   54       0               0  ...    1        0          0
3   30       0               0  ...    0        0          0
4   60       0               1  ...    1        0          1

[5 rows x 19 columns]
0    0
1    0
2    0
3    0
4    1
5    1
6    0
7    0
8    0
9    0
Name: MORTALITY, dtype: int64
```

Figure 22. Python code output showing the result of replacing the date died variable with an binary category variable instead simply denoting whether a patient died or not.

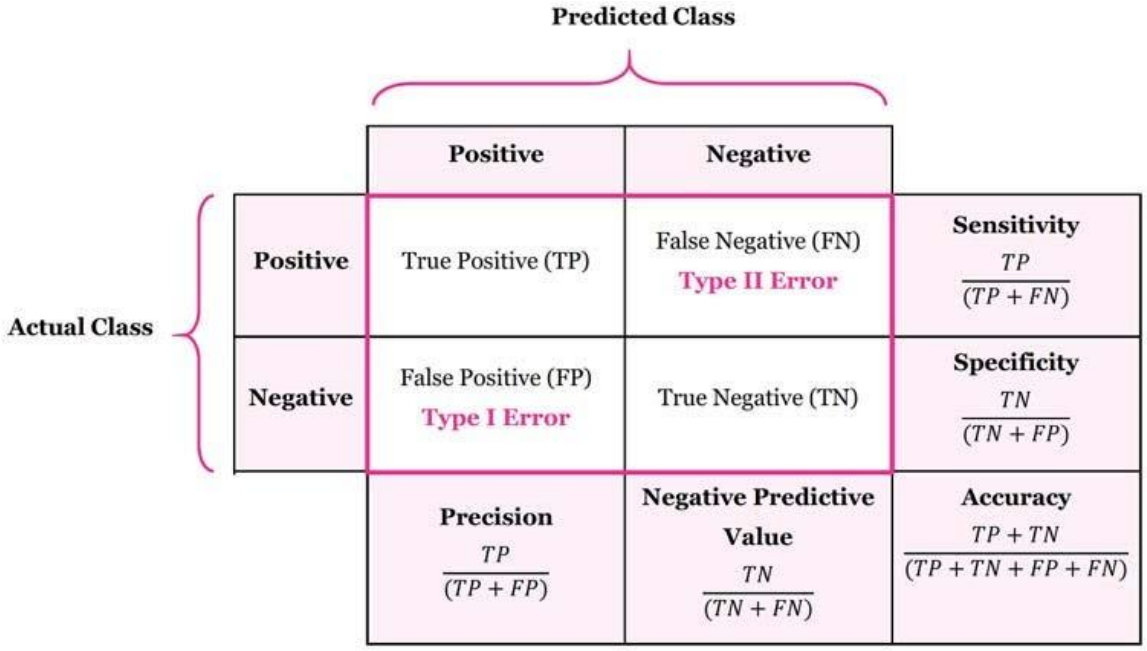

Figure 23. Confusion matrix explanation describing key metrics used to evaluate models in this project's analysis (SPP, 2024). All confusion matrices displayed follow this format.

Model Type	Name	Target Variable	Model Type	Key Parameters	Explanation
Decision Tree	DT_M	Mortality	Classifier (Tree-based)	class_weight='balanced', random_state=69	A simple tree-based model balancing classes
Logistic Regression	LR_M	Mortality	Classifier (Linear Model)	class_weight='balanced', max_iter=1000, random_state=69	A linear model that assumes a logistic relationship between predictors and binary outcomes
Random Forest	RF_M	Mortality	Ensemble Classifier	class_weight='balanced', n_estimators=100, random_state=69	An ensemble of decision trees for classification, improving accuracy by averaging predictions
Gradient Boosting	GB_M	Mortality	Ensemble Classifier	random_state=69	Builds sequential trees to reduce error, using gradient boosting for higher accuracy in predictions
Naïve Bayes	NB_M	Mortality	Probabilistic Classifier	Default parameters (GaussianNB)	A probabilistic model assuming feature independence
Naïve Bayes	NB_W_M	Mortality	Probabilistic Classifier	Custom priors computed using compute_class_weight('balanced', ...)	A weighted Naive Bayes to address imbalances in target classes by altering prior probabilities
Decision Tree	DT_I	ICU	Classifier (Tree-based)	class_weight='balanced', random_state=69	A simple tree-based model balancing classes
Logistic Regression	LR_I	ICU	Classifier (Linear Model)	class_weight='balanced', max_iter=1000, random_state=69	A linear model that assumes a logistic relationship between predictors and binary outcomes
Random Forest	RF_I	ICU	Ensemble Classifier	class_weight='balanced', n_estimators=100, random_state=69	An ensemble of decision trees for classification, improving accuracy by averaging predictions
Gradient Boosting	GB_I	ICU	Ensemble Classifier	random_state=69	Builds sequential trees to reduce error, using gradient boosting for higher accuracy in predictions
Naïve Bayes	NB_I	ICU	Probabilistic Classifier	Default parameters (GaussianNB)	A probabilistic model assuming feature independence
Naïve Bayes	NB_W_I	ICU	Probabilistic Classifier	Custom priors computed using compute_class_weight('balanced', ...)	A weighted Naive Bayes to address imbalances in target classes by altering prior probabilities
Decision Tree	DT_M_O	Mortality	Classifier (Tree-based)	max_depth, min_samples_split, min_samples_leaf, criterion, ccp_alpha (values optimized using RandomizedSearchCV and KFold cross-validation)	Decision Tree optimized for recall using hyperparameter tuning and manual oversampling to address class imbalance.
Decision Tree	DT_M_F1	Mortality	Classifier (Tree-based)		Decision Tree optimized for F1 using hyperparameter tuning and manual oversampling to address class imbalance.

Table 3. Table showing all the models generated for this analysis including pertinent details.

Index	Model	Target Variable	Accuracy	Confusion_matrix	F1	Sensitivity	Time Running
1	DT_M	Mortality	88.00%	[[1290959 180800] [8904 99896]]	51.29%	91.82%	30 sec
2	LR_M	Mortality	88.39%	[[1298022 173737] [9747 99053]]	51.92%	91.04%	3 min
3	RF_M	Mortality	88.25%	[[1295544 176215] [9453 99347]]	51.69%	91.31%	7 min
4	GB_M	Mortality	86.35%	[[1268376 203383] [12433 96367]]	47.17%	88.57%	10 min
5	NB_M	Mortality	83.81%	[[1224046 247713] [8257 100543]]	44.00%	92.41%	3 sec
6	NB_W_M	Mortality	94.70%	[[1448836 22923] [60898 47902]]	53.34%	44.03%	3 sec
7	DT_I	ICU	85.60%	[[1328397 225934] [1684 24544]]	17.74%	93.58%	30 sec
8	LR_I	ICU	82.12%	[[1271658 282673] [8 26220]]	15.65%	99.97%	3 min
9	RF_I	ICU	86.23%	[[1338836 215495] [2112 24116]]	18.14%	91.94%	7 min
10	GB_I	ICU	82.17%	[[1272638 281693] [47 26181]]	15.67%	99.82%	10 min
11	NB_I	ICU	82.09%	[[1271297 283034] [0 26228]]	15.64%	100.00%	3 sec
12	NB_W_I	ICU	98.37%	[[1553526 805] [25006 1222]]	8.65%	4.66%	3 sec
13	DT_M_O	Mortality	84.33%	[[1229158 242601] [5019 103781]]	45.60%	95.39%	5 min
14	DT_M_F1	Mortality	88.21%	[[1291301 180458] [5884 102916]]	52.48%	94.59%	5 min

Table 4. Table showing the results for each model generated in this analysis.

```python
# Define evaluation metrics
def evaluate_model(y_true, y_pred):
    """Evaluate model on specified metrics."""
    cm = confusion_matrix(y_true, y_pred)
    accuracy = accuracy_score(y_true, y_pred)
    f1 = f1_score(y_true, y_pred)
    sensitivity = recall_score(y_true, y_pred)  # Sensitivity is recall for the positive class
    return {"confusion_matrix": cm, "accuracy": accuracy, "f1": f1, "sensitivity": sensitivity}

# Common function to run k-fold cross-validation
def run_kfold_cv(model, X, y, k=5):
    """Perform K-fold cross-validation for a given model and dataset."""
    metrics = []
    kf = KFold(n_splits=k, shuffle=True, random_state=69)
    for train_index, test_index in kf.split(X):
        X_train, X_test = X.iloc[train_index], X.iloc[test_index]
        y_train, y_test = y.iloc[train_index], y.iloc[test_index]

        # Fit the model
        model.fit(X_train, y_train)

        # Predict and evaluate
        y_pred = model.predict(X_test)
        metrics.append(evaluate_model(y_test, y_pred))

    # Aggregate metrics
    final_metrics = {
        "confusion_matrix": sum(metric["confusion_matrix"] for metric in metrics),
        "accuracy": np.mean([metric["accuracy"] for metric in metrics]),
        "f1": np.mean([metric["f1"] for metric in metrics]),
        "sensitivity": np.mean([metric["sensitivity"] for metric in metrics]),
        "Target Variable": "Mortality"
    }
    return final_metrics
```

Figure 24. The evaluate_model and run_kfold_cv functions shown in the Spyder IDE.

```python
# Individual model functions
def decision_tree_model(X, y):
    model = DecisionTreeClassifier(class_weight="balanced", random_state=69)
    return run_kfold_cv(model, X, y)

def logistic_regression_model(X, y):
    model = LogisticRegression(class_weight="balanced", max_iter=1000, random_state=69)
    return run_kfold_cv(model, X, y)

def random_forest_model(X, y):
    model = RandomForestClassifier(class_weight="balanced", n_estimators=100, random_state=69)
    return run_kfold_cv(model, X, y)

def gradient_boosting_model(X, y):
    model = GradientBoostingClassifier(random_state=69)
    return run_kfold_cv(model, X, y)

def naive_bayes_model(X, y):
    model = GaussianNB()
    return run_kfold_cv(model, X, y)

# NB model adjusted with class weights
def naive_bayes_modelW(X, y):
    class_weights = compute_class_weight('balanced', classes=np.unique(y), y=y)
    priors = class_weights / class_weights.sum()
    model = GaussianNB(priors=priors)
    return run_kfold_cv(model, X, y)

# Results dictionary to store results for all models
model_results = {}

# Workflow to run models
def run_decision_tree(X, y):
    model_results["Decision Tree"] = decision_tree_model(X, y)

def run_logistic_regression(X, y):
    model_results["Logistic Regression"] = logistic_regression_model(X, y)
```

Figure 25. The model functions and run model functions shown in the Spyder IDE.

```
# Run individual models
run_decision_tree(X, y) #30 sec
run_logistic_regression(X, y) # 3min
run_random_forest(X, y) # 7 min
run_gradient_boosting(X, y) # 10 min
run_naive_bayes(X, y) # 3 sec
run_naive_bayesW(X, y) # 3 sec

# Compile results into a table
results_table = compile_results()
print(results_table)

# Code to download the df using a CSV file
# Path
output_file_path = 'C:/Users/soari/Documents/Assignments/Data Analytics/UMGC/Fall 2024 Data 670/Assignment 5/Mort_results.csv'

# Export the cleaned DataFrame to a CSV file
results_table.to_csv(output_file_path, index=False)

# Print confirmation message
print(f"DataFrame successfully exported to {output_file_path}")
```

Figure 26. The run model functions being implemented along with compile_results function being run with all results saved to a filepath shown in the Spyder IDE.

```
# Update values of 97 to 0 in the 'ICU' column
combined_df_final['ICU'] = combined_df_final['ICU'].replace(97, 0)

# Verify the update
print("Updated values in 'ICU':")
print(combined_df_final['ICU'].value_counts())



# Example usage
X = combined_df_final.drop(columns=["MORTALITY", "ICU"])
y = combined_df_final["ICU"]

# Run individual models
run_decision_tree(X, y) #30 sec
run_logistic_regression(X, y) # 3min
run_random_forest(X, y) # 7 min
run_gradient_boosting(X, y) # 10 min
run_naive_bayes(X, y) # 3 sec
run_naive_bayesW(X, y) # 3 sec

# Compile results into a table
results_table = compile_results()
print(results_table)


# Code to download the df using a CSV file
# Path
output_file_path = 'C:/Users/soari/Documents/Assignments/Data Analytics/UMGC/Fall 2024 Data 670/Assignment 5/ICU_results.csv'

# Export the cleaned df to a CSV file
results_table.to_csv(output_file_path, index=False)

# Print confirmation message
print(f"DataFrame successfully exported to {output_file_path}")
```

Figure 27. The ICU modeling updates from mortality modeling shown in the Spyder IDE.

```
# Step 2: Manual Oversampling to handle class imbalance
data = pd.concat([X, y_mortality], axis=1)
majority = data[data["MORTALITY"] == 0]
minority = data[data["MORTALITY"] == 1]
minority_upsampled = resample(minority, replace=True, n_samples=len(majority), random_state=69)
upsampled_data = pd.concat([majority, minority_upsampled])

X_resampled = upsampled_data.drop("MORTALITY", axis=1)
y_resampled = upsampled_data["MORTALITY"]

# Step 3: Expand the hyperparameter grid
param_grid = {
    "max_depth": [5, 10, 15, None],           # Control tree depth
    "min_samples_split": [2, 5, 10],          # Minimum samples to split a node
    "min_samples_leaf": [1, 2, 5],            # Minimum samples required at a leaf
    "criterion": ["gini", "entropy"],         # Splitting criteria
    "ccp_alpha": [0.0, 0.01, 0.1],            # Cost Complexity Pruning parameter
}
```

Figure 28. The optimized decision tree model, DT_M_O, edits showing the hyperparameter tuning and manual oversampling in the Spyder IDE.

```
# Step 4: Define F1 scorer for optimization
f1_scorer = make_scorer(f1_score, pos_label=1)

# Step 5: Set up 5-fold cross-validation
kf = KFold(n_splits=5, shuffle=True, random_state=69)

# Step 6: Initialize Decision Tree with class weights for imbalance
decision_tree = DecisionTreeClassifier(random_state=69, class_weight="balanced")

# Step 7: Perform randomized hyperparameter search
random_search = RandomizedSearchCV(
    decision_tree,
    param_distributions=param_grid,
    scoring=f1_scorer,  # Optimize for F1 score
    cv=kf,
    n_jobs=1,                # Sequential execution to avoid memory errors
    random_state=69,
    n_iter=20,               # Explore more combinations
    verbose=2,               # Log progress for debugging
)
random_search.fit(X_resampled, y_resampled)
```

Figure 29. The optimized decision tree model, DT_M_F1, edits showing the F1 scorer in the Spyder IDE.

```
# 2. Show Tree Rules
from sklearn.tree import export_text

tree_rules = export_text(best_model, feature_names=list(X.columns))
rules_file = 'C:/Users/soari/Documents/Assignments/Data Analytics/UMGC/Fall 2024 Data 670/Assignment 6/DT_M_F1_Tree_Rules.txt'

# Save rules to a file
with open(rules_file, 'w') as file:
    file.write(tree_rules)

# Print a sample of the rules for user
print("Decision Tree Rules:")
print(tree_rules[:1000])  # Display the first 1000 characters of the rules
```

Figure 30. Code demonstrating how to save all rules into a text file from the optimized decision tree model, DT_M_F1, in the Spyder IDE.

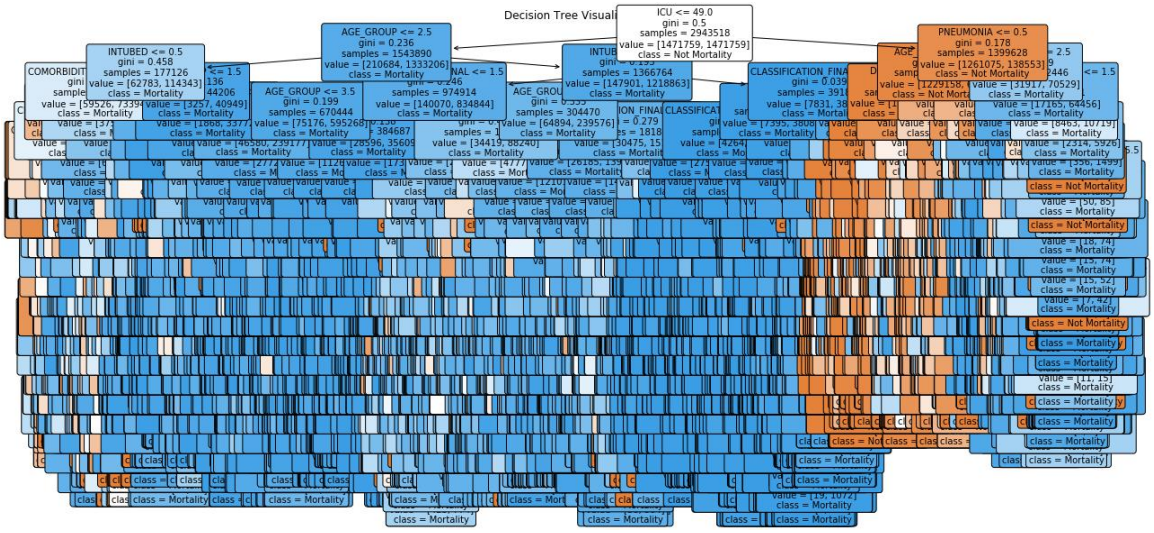

Figure 31. The champion decision tree model, DT_M_F1, demonstrating all rules from the tree.

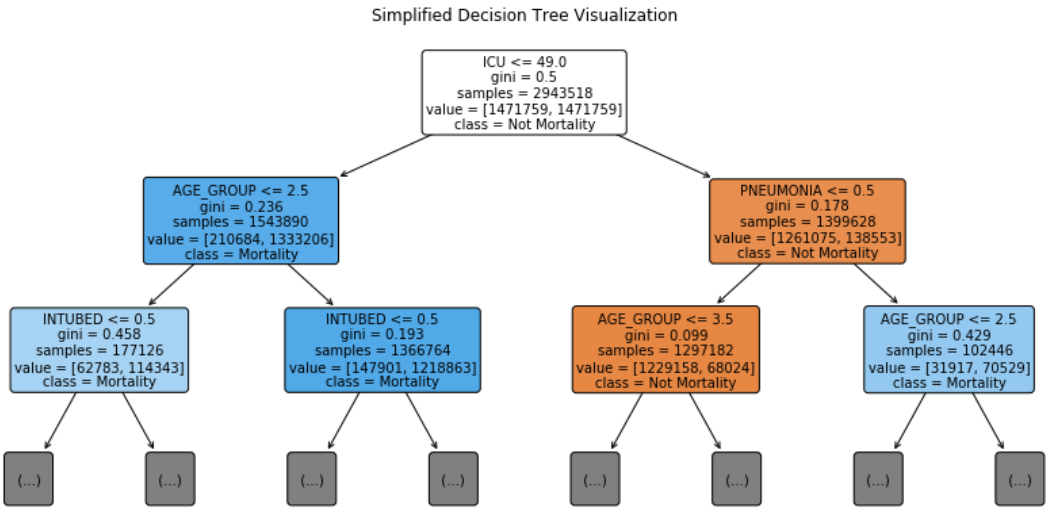

Figure 32. The champion decision tree model, DT_M_F1, demonstrating the top 2 layers of rules from the tree.

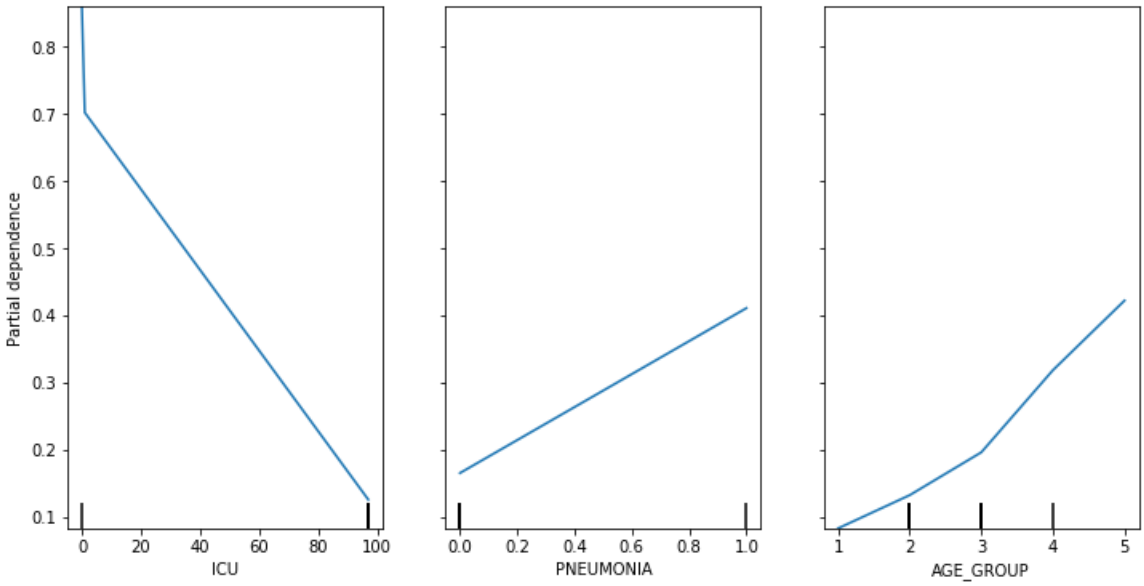

Figure 33. The champion decision tree model, DT_M_F1, partial dependence plots for ICU, Pneumonia, and Age Group.

Figure 34. Workflow demonstrating the risk matrix generation using feature importance analysis from the champion model, DT_M_F1. Those feature importance values were applied to each of the variables and summed to generate a risk score.

```python
# Step 1: Copy df as back up
combined_df_RM = combined_df_final
combined_df_RM = combined_df_RM.replace(97, 0)

# Step 2: Compute the weighted sum (Risk Score)
combined_df_RM['Risk_Score'] = (
    combined_df_final['ASTHMA'] * 0.046171279 +
    combined_df_final['CARDIOVASCULAR'] * 0.001509685 +
    combined_df_final['CLASSIFICATION_FINAL'] * 0.002129031 +
    combined_df_final['COPD'] * 0.0021714 +
    combined_df_final['DIABETES'] * 0.002478661 +
    combined_df_final['HYPERTENSION'] * 0.013663514 +
    combined_df_final['ICU'] * 0.002057093 +
    combined_df_final['INMSUPR'] * 0.01669408 +
    combined_df_final['INTUBED'] * 0.804682841 +
    combined_df_final['OBESITY'] * 0.003663514 +
    combined_df_final['OTHER_DISEASE'] * 0.002471875 +
    combined_df_final['OUTPATIENT'] * 0.002574518 +
    combined_df_final['PNEUMONIA'] * 0.003914547 +
    combined_df_final['PREGNANT'] * 0.002641832 +
    combined_df_final['RENAL_CHRONIC'] * 0.074350044 +
    combined_df_final['SEX'] * 0.00313825 +
    combined_df_final['TOBACCO'] * 0.001746782 +
    combined_df_final['AGE_GROUP'] * 0.001960395 +
    combined_df_final['COMORBIDITY_COUNT'] * 0.015644174
)

# Step 3: Classify risk level based on the risk score
def classify_risk(score):
    if score > 0.7:
        return "High"
    elif score > 0.4:
        return "Medium"
    else:
        return "Low"

combined_df_RM['Risk_Label'] = combined_df_RM['Risk_Score'].apply(classify_risk)

combined_df_RM.to_csv(output_file_path1, index=False)
```

Figure 35. Python code to generate the risk matrix and apply it to the full dataframe. This is performed using the feature importance analysis values from the champion model, DT_M_F1. Those feature importance values were applied to each of the variables and summed to generate a risk score and risk label.