**XGBoost Classification Analysis of Comorbidities for Predicting Mortality from COVID**

**J. Amanda Terzin**

**Western Governors University**

**Master of Science in Data Analytics Capstone**

**November 22, 2022**

**Introduction**

The recent Covid-19 pandemic created an unprecedented strain on hospitals and healthcare organizations. As a physician assistant working in a large emergency department throughout the pandemic, I was faced with the challenge every shift. The need for beds outnumbered the available beds (French et al., 2021). The inpatient staff was stretched thin while the outpatient staff was furloughed due to clinic closures (Office of the Assistant Secretary for Planning and Evaluation, 2022). Shortages of supplies and medications for patients and protective equipment for staff were caused by both increased use and supply chain issues (Hick, 2021). The fiscal impact was devastating as well. Hospitals faced a paradox of increased business with less revenue. Reimbursement was down due to an increase in uninsured and Medicaid patients laid off from their jobs. Elective surgeries and procedures that are a major revenue source were halted during surges (Balser, 2021).

**Research Question**

 How can healthcare organizations mitigate the impact of Covid? The ability to identify mortality risk based on preexisting conditions would allow hospitals and healthcare organizations to better manage resources. Before infection, ensure that higher-risk patients are aware of their risks, encourage vaccinations, and educate them on how to avoid infection. At the time of diagnosis, early intervention with medications can reduce the severity of the infection. Hospital admissions could be reduced by identifying the patients that can be managed safely as an outpatient (Balser, 2021). Can a predictive model be developed using the comorbidities of patients with Covid-19?

**Hypothesis**

The hypothesis of this study is that the comorbidities of patients diagnosed with Covid-19 can be used to predict the risk of mortality with statistical significance. To complete this study, the comorbidities and outcomes of patients with Covid-19 will be extracted from their medical records. A classification model will be developed with the comorbidities as the predictors and mortality as the target.

**Data Collection**

The data set used for this study is publicly available synthetic data generated using The MITRE Corporation’s SyntheaTM Synthetic Patient Population Simulation. The data set was created to facilitate modeling COVID data without privacy and security risk to patients (Walonoski et al., 2020). It contains 16 csv files of which 4 were used: patients, conditions, encounters, and observations. The set contains data from 124,150 patients, 88,166 of whom were positive for COVID. Pediatric patients were excluded from this study. Comorbidities, infection severity, and mortality rate for children are statistically different from adults (Khera et al., 2021). The final count of COVID-positive adult patients is 71,329.

One advantage of the data set is that it is synthetically generated and publicly available. In the paper presenting the Synthea patient-generating software, Walonoski et al. (2017) stated “Healthcare lags other industries in information technology, data exchange, and interoperability. The lack of freely distributable health records has long hindered innovation in health care.” Patient privacy and security restrict access to medical records needed to develop statistical models. One disadvantage is that the model produced with synthetic data is not immediately actionable. Once the model is developed, it will need to be applied to real-world data to guide healthcare organizations to manage resources (Walonoski et al., 2017).

Gathering the needed data from the Synthea set was challenging. The initial set contains 16 separate csv files. Following a review of each of these, 4 were found to have the needed information for this study: patients, conditions, encounters, and observations. The patients’ file was the most straightforward, containing one row per patient. The conditions, observations, and encounters files each contained more than a million rows. Each patient had multiple rows for each condition, observation, and encounter, and repeated additional rows for entries repeated on different dates. For example, a patient’s conditions are documented with each visit encounter, resulting in the same condition listed in multiple rows with different dates. The strategy I developed to tackle this large amount of data was to start with the patients’ file as the anchoring base with the patient’s id, birthday, and gender. The COVID test results in the observations file then identified the COVID-positive patients. The date of the test allowed me to identify adult patients at the time of diagnosis. I was able to use the patient id to reduce each of the 4 files to the 71,329 patients of interest in this study. Because this study focuses on assessing risk at the time of COVID diagnosis, conditions recorded after the date of diagnosis were disregarded. Other data extracted from the observations file includes the cause and date of death for determining patients who died from their COVID infection and the patient’s smoking history. The conditions file contained 170 unique conditions, of which 18 were chosen as the comorbidities to include in this study. Lastly, the Intensive Care Unit (ICU) and hospital days were extracted from the encounters file for any hospital admissions

**Table 1**

*Summary of original data use*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **csv** | **Rows** |  | **Columns** |  |
|  | **Original** | **Kept** | **Original** | **Kept** |
| patients | 124,150 | 71,329 | 25 | 3 |
| conditions | 1,143,900 | 114,349 | 6 | 3 |
| encounters | 1,881,954 | 21,929 | 15 | 6 |
| observations | 1,621,9969 | 132,648 | 8 | 4 |

**Data Extraction and Preparation**

Data extraction, cleaning, and preparation were done with Python in a Jupyter Notebook. Python was chosen for its versatility and numerous libraries. While R is very strong for statistical computations and is somewhat more efficient with fewer lines of code, Python has a lot of additional functionality and is more customizable. Additionally, Python has a “massive community” with numerous solutions available online (Singh, 2022).  Jupyter Notebook facilitates exploratory data analysis and code writing, allows the user to add documentation along with code, and makes it easy to share (Kumaraswamy, 2018).

**Step 1**

Graphical user interface, text

Description automatically generated

Pandas read\_csv read the csv file into a data frame using comma-separated values to form the columns. Each csv has a separate data frame.

**Step 2**

The patients’ file was straightforward with one line per patient. 3 columns were extracted and renamed and the date column was changed to the date data type.

Graphical user interface, text, application

Description automatically generated

**Step 3**

**Observations required more work with multiple lines per patient. Once the** 3 columns were extracted and renamed and the date column changed to the date data type, the 3 types of observation of interest were extracted. Since there were multiple duplicate entries for each patient, duplicates were dropped.

Graphical user interface, text, application

Description automatically generated

**Step 4**

**Next, the positive Covid patients were identified.**

Graphical user interface, text, application

Description automatically generated

**The ids of Covid patients were used to isolate them in all 4 data frames, non-Covid patients were dropped.**

Graphical user interface, text, application

Description automatically generated

**Step 5**

**The Covid test date was added to the patients’ data frame. The age at the time of diagnosis was calculated.**

Graphical user interface, text, application

Description automatically generated

**Subtracting dates produces an answer in days. Age was converted to string type, the regex module was used to strip off the text, and then it was converted to an integer. To convert to years, Age was divided by 365 and rounded. It may have been simpler to just use the year and subtract, but I felt this was more precise.**

Graphical user interface, text, application

Description automatically generated

**Step 6**

**Pediatric patients were dropped.**

Graphical user interface, text, application, email

Description automatically generated

**Step 7**

**Age was evaluated for outliers using OutlierTrimmer from the Feature-engine library.**

Graphical user interface, text, application, email

Description automatically generated

**Age over 107 is likely bad data so outliers were dropped.**

Graphical user interface, text, application

Description automatically generated

**Step 8**

**The data frames were reduced to the final list of patient ids of Covid-positive adults.**

Graphical user interface, text, application

Description automatically generated

**Step 9**

**Finally, the date of death for patients who died from Covid was extracted and added to the main patients’ data frame and converted to the target variable “Mortality”. The smoking status field will be addressed with the conditions data frame.**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Step 10**

**The conditions data frame also has multiple rows for each patient.** 3 columns were extracted and renamed and the date column was changed to the date data type.

Graphical user interface, text, application

Description automatically generated

**Step 11**

**Only conditions diagnosed before Covid were extracted. The date of Covid diagnosis was added as a column and the rows dated before Covid were marked. The same technique for subtracting the dates and stripping the text was used again here.**

Graphical user interface, text, application

Description automatically generated

**Once they were marked, it was easy to select the rows with a 1 in the PreCovid column.**

Graphical user interface, text, application

Description automatically generated

**The data frame was reduced to just the id and condition column, and duplicates were dropped.**

Graphical user interface, text, application, email

Description automatically generated

**Step 12**

**The last extraction step for conditions was to extract only the 18 conditions relevant to this study from the 170 in the data frame. The 2 obesity conditions and the 2 Alzheimers conditions were consolidated for a total of 16 conditions. Duplicate entries were dropped.**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Final count.**

Graphical user interface, text, application

Description automatically generated

**Step 13.**

**The conditions were then pivoted to columns under index Id with one row per patient.**

Graphical user interface, text, application

Description automatically generated

**Step 14**

**The smoking status from observations was extracted and added to the conditions data frame.**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Pivot value to columns under index Id and add to the conditions data frame.**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Combine redundant smoking columns.**

Graphical user interface, text, application

Description automatically generated

**Step 15**

**Remove prediabetes for patients with diabetes.**

Graphical user interface, text, application

Description automatically generated

**Step 16**

**Finally, from encounters, 6 columns were extracted and a data frame for inpatient stays for Covid was created.**

Graphical user interface, text

Description automatically generated

**4 columns were selected and renamed. The 3 types of admissions were reduced to 2.**

Graphical user interface, text

Description automatically generated

**Days for each admission were computed in a new column using the same technique from above.**

Graphical user interface, text, application

Description automatically generated

**The sum of the total days per type of admission for each patient was computed and reported in a new column. The data frame was reduced to 3 columns and the duplicates dropped.**

Graphical user interface, text, application

Description automatically generated

**Pivot hospital and ICU to columns under index Id and create a third column with the total days admitted to the hospital.**

Graphical user interface, text, application

Description automatically generated

**Step 18**

**The final step for encounters is to add the ICU and total hospital days to the main patients’ data frame.**

Graphical user interface, text, application

Description automatically generated

**Step 19**

**Prepare patients’ data frame for combining with conditions. Select the needed columns and encode gender.**

Graphical user interface, text, application

Description automatically generated

**Combine the conditions and patients’ data frame and change the data type to integer.**

Graphical user interface, text

Description automatically generated

**Arrange the columns.**

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**The final data set has 19 independent variables and one target. The are 71,091 rows representing the patients. The ICU and hospital days are to compare hospital utilization between recovered and survived patients.**

**Table 2**

***Summary of Final Variables***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Variable Type** | **Use** |
| Id | Categorical | Independent | Index |
| ICU\_days | Continuous | Independent | Demonstration |
| Total\_Hosp\_days | Continuous | Independent | Demonstration |
| Age | Continuous | Independent | In model |
| Gender | Categorical | Independent | In model |
| Alzheimer`s\_disease | Categorical | Independent | In model |
| Asthma | Categorical | Independent | In model |
| Obesity | Categorical | Independent | In model |
| Chronic\_congestive\_heart\_failure | Categorical | Independent | In model |
| Chronic\_kidney\_disease | Categorical | Independent | In model |
| Chronic\_obstructive\_bronchitis | Categorical | Independent | In model |
| Coronary\_Heart\_Disease | Categorical | Independent | In model |
| Diabetes | Categorical | Independent | In model |
| History\_of\_myocardial\_infarction | Categorical | Independent | In model |
| Hyperlipidemia | Categorical | Independent | In model |
| Hypertension | Categorical | Independent | In model |
| Hypertriglyceridemia | Categorical | Independent | In model |
| Prediabetes | Continuous | Independent | In model |
| Pulmonary\_emphysema | Categorical | Independent | In model |
| Stroke | Categorical | Independent | In model |
| Current\_smoker | Categorical | Independent | In model |
| Former\_smoker | Categorical | Independent | In model |
| Mortality | Categorical | Dependent | In model |

**Analysis**

**Exploratory Analysis**

**Once the data was extracted, cleaned, and prepared, the first step before modeling was to perform some exploratory analysis. A comparison of hospital utilization between deceased and recovered patients revealed both total hospital days and ICU days are significantly higher for the mortality group.**

Graphical user interface, text, email

Description automatically generated

**Bivariate plots were generated to visually explore possible relationships between each independent variable and the target.**

A picture containing table

Description automatically generatedGraphical user interface, application

Description automatically generated

**A correlation heatmap of the predictors was produced to evaluate** **relationships between the independent variables.**

Graphical user interface, application

Description automatically generated

**Both techniques are categorized as descriptive statistics, looking at how the data is distributed and how it may vary together (either positively or negatively). The advantage of these techniques is that it is easy to see possible relationships. The disadvantage is that it is not conclusive, and further investigation is necessary to show statistical significance (**Akiode, 2021)**.**

**Statistical Tests**

**The second step was evaluating the relationship between the independent variables and the target with a statistical test. The Chi-squared test for independence is the standard when both the predictor and target are categorical. The test computes whether the observed frequencies are more than would be expected for unrelated variables and if the difference is statistically significant (Bevans, 2022).**

Graphical user interface, text, application, email

Description automatically generated

**Model Development**

**Once the first two steps were completed, the model was created using an eXtreme Gradient Boosting (XGBoost) classifier. XGBoost is an ensemble learning algorithm that uses Classification and Regression trees (CART) to categorize an instance (a patient) based on its features (comorbidities) (Bex, 2021). One advantage of this technique is that there are no model assumptions. Unlike logistic regression, XGBoost is robust to outliers and multicollinearity and can capture non-linear relationships. A disadvantage is that tuning the hyperparameters can be difficult (**Gupta, 2020)**.**

**To simplify the hyperparameter tuning, the HyperOpt library was used. HyperOp uses Bayesian optimization that takes the results from previous steps to guide which hyperparameters to try in the next step, unlike grid and random searches that do not**. A list of parameters with the values to try is given to the objective function with the XGBoost classifier and HyperOp tries to find the best combination that minimizes loss **(**Koehrsen, 2018).

An important consideration in modeling this data is the fact it is imbalanced, that is the target for prediction is the minority by almost 20:1. Without taking any measures to account for the imbalance, the model is unlikely to do a very good job at predicting the minority class. Several strategies can improve performance. The first is stratifying the split into train and test and using sklearn’s RepeatedStratifiedFold for cross-validation. This assures the distribution of the minority class stays consistent with each split or fold. The second is choosing the appropriate evaluation metric. The best choice depends on what is most important, in this case identifying all of the true positives is the priority. The area under the precision-recall curve (AUC-PR), which focuses on the true positives, is the best metric as opposed to the area under the receiver operating characteristic curve (ROC-AUC), which treats both classes equally. The third is using XGBoost’s scale\_pos\_weight parameter to balance the weights (Mumtaz, 2020).

**Train Test Split**

The data was divided into a train and test set, 80/20 split with stratify.

Graphical user interface, text, application, email

Description automatically generated

**Hyperparameter Optimization**

A list of hyperparameters and values to iterate through were saved to space, the k-fold was set at RepeatedStratifiedFold with 10 splits and 3 repeats, and cross-validation scoring was set to average precision (AUC\_PR). The HyperOpt function fmin iterated over the parameters to find the best combination with the lowest loss.

Graphical user interface, text, application

Description automatically generated

**Train Model**

**The best parameters were entered into the classifier, trained with the training set, and predictions made with the test set.**

Graphical user interface, text, application

Description automatically generated

**Evaluate Model**

Graphical user interface, application

Description automatically generated

For classification models, there are several different evaluation metrics based on various combinations of true and false negatives (TN, FN) and true and false positives (TP, FP). Accuracy is most frequently used in classification tasks. It is the number of correct predictions (true negatives and positives) divided by the total number of predictions. In imbalanced datasets, however, it is not a good measure. The true negatives are a much higher number compared to the small number of true positives. Using accuracy, a model that classified the entire test set (14,219) as negative, the number of correct predictions would be 13,528, which gives us an accuracy of 13,528/14,219 = 95%. The ROC-AUC score is the most common metric used to compare models. It is computed using the true positive rate and the false positive rate. The baseline score to compare a model against a baseline of 0.5, the value that a no-skills classifier would make (or a coin flip). It is not very sensitive to the minority class in an imbalanced data set (Brownlee, 2021). The model performed better than the baseline at 0.83.

Graphical user interface, application

Description automatically generated

The AUC-PR is the best choice to compare models when finding all of the true positives are the objective. The curve is formed by plotting recall (true positives / (true positives + false negatives)) against precision (true positives / (true positives + false positives)) and is more sensitive to the minority class. As opposed to ROC-AUC, the baseline is the percentage of positive samples in the dataset, the chance of randomly choosing the right answer. (Draelos, 2019). The baseline is 0.49 and the model performed better at 0.22.

Graphical user interface, application

Description automatically generated

With the ultimate goal of understanding the risk factors of mortality, we can examine the predictions and how much the factors contributed to the prediction. XGBoost reports feature importance with 3 different metrics as Lundberg (2018) explained: “1) Weight- the number of times a feature is used to split the data across all trees. 2) Cover- the number of times a feature is used to split the data across all trees weighted by the number of training data points that go through those splits. 3) Gain- the average training loss reduction gained when using a feature for splitting.” Unfortunately, the metrics can give 3 different rankings of importance making the explainability of the model difficult. The SHAP package’s TreeExplainer does a better job using game theory to produce consistent and accurate results. The algorithm uses a complicated calculation to compute “SHAP values” to interpret contribution. The package also offers many easily interpreted plots that are especially useful for explaining results to stakeholders (Lundberg, 2018).

Graphical user interface, text, application

Description automatically generated

The model summary shows the top 11 features and their mean SHAP value.

Graphical user interface, application

Description automatically generated

The violin plot gives even more information about what values of the features contributed to the predictions. For example, high values of hypertension (1 in this case, indicating the patient has the condition) contributed to a positive prediction, while low values (0, indicating the patient does not have the condition) contributed to a negative prediction. If the opposite were true the plot on the right side would be blue and the left red. Higher age and all of the conditions are risk factors for mortality.

Furthermore, you can look at a specific instance (patient) and see what contributed to the prediction.

Graphical user interface

Description automatically generated

In this case, it is a false negative prediction. Age 53, male gender, obesity, and prediabetes were not enough to predict mortality. However, the patient with the same conditions but age 65 was correctly predicted in the positive class.

Graphical user interface, application

Description automatically generated

**Data Summary and Implications**

**Table 3**

*Summary of Model Evaluation Metrics*

|  |  |
| --- | --- |
| **Metric** |  |
| Accuracy | 0.75 |
| ROC-AUC | 0.83 |
| AUC-PR | 0.22 |
| True Positives | 525 |
| False Negatives | 166 |
| True Negatives | 10,141 |
| False Positives | 3,387 |

**Summary**

The answer to the research question is yes a predictive model can be made using the comorbidities to predict mortality. Further, the interpretation allows exploration of how each comorbidity contributes to the prediction. The model did a fairly good job of predicting mortality based on the risk factors, but the fact it missed 166 out of the 619 true positives and falsely labeled 3,387 as positive shows that there are other factors affecting mortality.

**Limitations**

One of the limitations of the study is that the data is synthetic. Better insights might be found in a database of actual patients. Additionally, the comorbidities are defined somewhat generally. Diabetes, for instance, covers a wide range of severity. Quantifying the severity with tools such as the Diabetes Severity Score (DISSCO) could improve the predictive value (Zghebi, 2020).

**Recommendations**

Recommend applying the model to an actual patient database to evaluate performance.

**Direction for Future Study**

1. Include clinical measurements for each patient at the time of diagnosis. A patient who has been sick for days before coming in to be seen may already be too late to benefit from treatment as well as a patient diagnosed at the first sign of sickness (Sun et al. 2020).
2. Including COVID vaccination status to see if it changes mortality. At the time of this study, COVID vaccinations were not documented in the dataset.

**References**

Akiode, A. (2021). Visualization, not enough to suggest a relationship; Confirm it with a Statistical test. Retrieved November 10, 2022, from [Visualization, not enough to suggest a relationship; Confirm it with a Statistical test | by Ayobami Akiode | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/visualization-not-enough-to-assess-relationship-between-2-variables-combine-with-statistical-test-2a32a2c99153).

Balser, J., J. Ryu, M. Hood, G. Kaplan, J. Perlin, and B. Siegel. (2021). Care Systems COVID-19 Impact Assessment: Lessons Learned and Compelling Needs. *NAM Perspectives.* Discussion Paper, National Academy of Medicine, Washington, DC. [Care Systems COVID-19 Impact Assessment: Lessons Learned and Compelling Needs - National Academy of Medicine (nam.edu)](https://nam.edu/care-systems-covid-19-impact-assessment-lessons-learned-and-compelling-needs/?gclid=Cj0KCQiAyMKbBhD1ARIsANs7rEGXsk8MFS5gaVbbD4Gt1aQY0w-y2tzLRcEaG-0HMORwJBJ0a71F7IEaAjtXEALw_wcB)

Bevans, R. (2022). Choosing the Right Statistical Test | Types and Examples. Retrieved November 08, 2022, from [Choosing the Right Statistical Test | Types & Examples (scribbr.com)](https://www.scribbr.com/statistics/statistical-tests/)

Bex T. (2021). Beginner’s Guide to XGBoost for Classification Problems. Retrieved November 08, 2022, from [Beginner’s Guide to XGBoost for Classification Problems | Towards Data Science](https://towardsdatascience.com/beginners-guide-to-xgboost-for-classification-problems-50f75aac5390)

Brownlee, J. (2021). Tour of Evaluation Metrics for Imbalanced Classification. Retrieved November 10, 2022, from [Tour of Evaluation Metrics for Imbalanced Classification - MachineLearningMastery.com](https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/)

Draelos, R. (2019). Measuring Performance: AUPRC and Average Precision. Retrieved November 08, 2022, from [Measuring Performance: AUPRC and Average Precision – Glass Box (glassboxmedicine.com)](https://glassboxmedicine.com/2019/03/02/measuring-performance-auprc/)

French G, Hulse M, Nguyen D, et al. (2021). Impact of Hospital Strain on Excess Deaths During the COVID-19 Pandemic — United States, July 2020–July 2021. *MMWR Morb Mortal Wkly Rep*, *70,* 1613–1616. [Impact of Hospital Strain on Excess Deaths During the COVID-19 Pandemic — United States, July 2020–July 2021 | MMWR (cdc.gov)](https://www.cdc.gov/mmwr/volumes/70/wr/mm7046a5.htm#suggestedcitation)

Gupta, S. (2020). Pros and cons of various Machine Learning algorithms. Retrieved November 10, 2022, from [Pros and cons of various Machine Learning algorithms | by Shailaja Gupta | Towards Data Science](https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6)

Hick, J. L., D. Hanfling, M. Wynia, and E. Toner. (2021). Crisis Standards of Care and COVID-19: What Did We Learn? How Do We Ensure Equity? What Should We Do? *NAM Perspectives.* Discussion, National Academy of Medicine, Washington, DC. [Crisis Standards of Care and COVID-19: What Did We Learn? How Do We Ensure Equity? What Should We Do? - National Academy of Medicine (nam.edu)](https://nam.edu/crisis-standards-of-care-and-covid-19-what-did-we-learn-how-do-we-ensure-equity-what-should-we-do/)

Khera, N., Santesmasses, D., Kerepesi, C., & Gladyshev, V. N. (2021). COVID-19 mortality rate in children is U-shaped. *Aging, 13(16)*, 19954–19962. [COVID-19 mortality rate in children is U-shaped - PubMed (nih.gov)](https://pubmed.ncbi.nlm.nih.gov/34411000/)

Koehrsen, W. (2018). A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning. Retrieved November 15, 2022, from Retrieved November 15, 2022, from [A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning | by Will Koehrsen | Towards Data Science](https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f)

Kumaraswamy, A. (2018). 10 reasons why data scientists love Jupyter notebooks. Retrieved November 15, 2022, from [10 reasons why data scientists love Jupyter notebooks | Packt Hub (packtpub.com)](https://hub.packtpub.com/10-reasons-data-scientists-love-jupyter-notebooks/)

Lundberg, S. (2018). Interpretable Machine Learning with XGBoost. Retrieved November 19, 2022, from [Interpretable Machine Learning with XGBoost | by Scott Lundberg | Towards Data Science](https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27)

Mumtaz, A. (2020). How to Effectively Predict Imbalanced Classes in Python. Retrieved November 08, 2022, from [How to Predict Imbalanced Classes in Python | Towards Data Science](https://towardsdatascience.com/how-to-effectively-predict-imbalanced-classes-in-python-e8cd3b5720c4)

Office of the Assistant Secretary for Planning and Evaluation. (2022). Impact of the COVID-19 pandemic on the hospital and outpatient clinician workforce: challenges and policy responses. (Issue Brief No. HP-2022-13). U.S. Department of Health and Human Services. [aspe-covid-workforce-report.pdf (hhs.gov)](https://aspe.hhs.gov/sites/default/files/documents/9cc72124abd9ea25d58a22c7692dccb6/aspe-covid-workforce-report.pdf)

Singh, M. (2022). Python Vs R in 2022! Retrieved November 11, 2022, from [Python Vs R in 2022! Welcome back! R and Python are very… | by Manpreet Singh | Medium](https://preettheman.medium.com/python-vs-r-in-2022-a26ad7ffe6a7)

Sun, Q., Qiu, H., Huang, M., & Yang, Y. (2020). Lower mortality of COVID-19 by early recognition and intervention: experience from Jiangsu Province. *Annals of intensive care, 10(1)*, 33. https://doi.org/10.1186/s13613-020-00650-2

Walonoski, J., Klaus, S., Granger, E., Hall, D., Gregorowicz, A., Neyarapally, G., Watson, A., & Eastman, J. (2020). Synthea™ Novel coronavirus (COVID-19) model and synthetic data set. *Intelligence-based medicine, 1*, 100007. [Synthea™ Novel coronavirus (COVID-19) model and synthetic data set - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S2666521220300077?via%3Dihub)

Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., & McLachlan, S. (2018). Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association: JAMIA, 25(3)*, 230–238. [Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record - PubMed (nih.gov)](https://pubmed.ncbi.nlm.nih.gov/29025144/)

Zghebi SS, and others. (2020). Development and validation of the Diabetes Severity Score (DISSCO) in 139626 individuals with type 2 diabetes: a retrospective cohort study. BMJ Open Diabetes Res Care.;8:e000962