Patient Mortality Prediction

Zac Webel, Joelle Fitzgerald, Yuan (Ansel) Lian,

Introduction & Task

With the expansion of Electronic Health Records (EHRs), generating larger amounts of data to help physicians make clinical decisions, the feasibility and efficiency of precision medicine is dependent upon the management of vast quantities of EHR data. Enabling clinicians to comprehend large volumes of clinical data in a meaningful way is paramount when striving to streamline patient care. In recent years, predictive models have modernized the filtering and application of healthcare data to guide clinicians when caring for their patients. Machine Learning (ML) tools and models that produce relevant analysis and can predict patient outcomes are critical in supporting clinical decision making and the advancement of medicine in conjunction with the expansion of EHR systems. Knowing this, our team strived to create a machine learning model aimed to determine whether a patient will pass away within 180 days of having their last (available) exam utilizing EHR data. For this project, our team developed and evaluated an end-to-end machine learning pipeline predicting mortality of patients within six months after their last visit based on several clinical features.

Methods

Given the medical record data of a synthetic dataset created by the Centers of Medicare and Medicaid Services converted to the OMOP Common Data Model by the CMS Working Group of the Observational Health Data Sciences and Informatics (OHDSI) community, we selected relevant clinical features that could possibly predict a patient's mortality. Demographic data, patient diagnosis and/or comorbidities, medication records, and Charlson Comorbidity Index (CCI) – calculated for each patient – were used to weigh patient risk of mortality within 180 days (6 months) of their last recorded visit. In our feature selection process, we used the given CSV files: 'death', 'visit occurrence', 'observation period', 'observation', 'drug exposure', 'condition occurrence', and 'person'. Performing an exploratory data analysis of each given CSV file and cleaning for missing values, we worked to develop a set of features to train our model broken down into various steps.

In Step 1 of our feature selection, our objective was to populate demographic data for patients with a recorded birth date. From the 'persons' dataset, we identified patients' with a year of birth, gender, and ethnicity and created a filtered 'demographic' dataset. Visualizing the 'death' dataset, we merged our extracted persons' demographics on unique person ids to identify patients with a death date and time. During exploratory data analysis of our newly merged dataset, 76 patients did not have a recorded date of birth, 96822 patients did not have a recorded date of death (we assume these patients are still alive), and 1096 patients were reported to be

dead with a recorded death date and time. Now having demographic data populated for our training set, we moved to Step 2 - combining patient demographic data with information from their last recorded visit.

Date and time information from a patient's last recorded visit allows us to identify patients who meet or fail to meet our intended outcome - mortality after 180 days of their last recorded clinical exam. To find the last recorded visit for a given patient a unique list of patient ids from the demographics data set created in Step 1 was generated. For each unique patient id, visit information was extracted and appended by the most recent visit occurrence into a new list using the 'visit start datetime' column from the 'visit occurrence' file. This newly generated list of visit dates was added as a new column to the 'demographics' data set where all missing values for last visits were dropped. Now having the date of a given patient's last exam, we can calculate their age from birth until when they were last seen in clinic. A generated list of ages at last visit was appended as a new demographic feature to our demographic data set. From this, we visualized a histogram distribution of age noting that most of our patient population falls between the ages of 65 and 90 years old. In Step 2, we collected enough features to identify our main outcome. We assigned patients with an outcome value of zero if they did not have a recorded death date/time or with a death date difference (from their last clinical exam) greater than 180 days. Patients who did not fall within this criteria met our main mortality objective of dying within 180 days past their last visit and were assigned an outcome value of 1. From this, we successfully identified 520 patients who died within 180 days of their last visit and 97320 who did not meet this criteria. Having an extremely large training set of data, we randomly selected 10,000 samples (ensuring that all 520 identified to meet our outcome criteria were kept) by using pandas.sample with a fraction of 1 to shuffle the data frame and select the first 10,000 samples sorted by outcome descending. Subsequently, a new data frame was created having relevant demographic fields, last visit date and time, and recorded outcome (met or unmet) for each unique patient in our training set.

In Step 3, we used the 'conditions' file to populate the Charlson Comorbidity Index (CCI) for each patient in our training set. A good indication of survival rate, the Charlson Comorbidity Index calculates a patient's likelihood of mortality based on recorded clinical observations and diagnoses. Our team identified this calculated index as a necessary feature in predicting mortality 180 days after a patient's last recorded visit. Identifying corresponding condition concept codes for each CCI criteria and their respective score, we assigned a given weight to each clinical condition. The conditions data set was filtered to only include condition concept codes relevant to calculating the CCI for each patient. Creating a new data frame with populated condition concept codes (columns) for each unique patient (rows), we went through each patient and calculated a condition weight based on CCI criteria adjusting for patients with several concept codes for the same condition. For example, a patient who meets several dementia codes will be recorded as 1. The CCI was calculated for each patient based on accumulated conditions (each given a unique weight) relevant to the Charlson Comorbidity Index calculation. Under the CCI

estimation, patients are given larger scores as age increases which is also incorporated into our CCI feature calculation (see Figure 1). Exploring how the CCI for each patient correlates with our mortality outcome, we found there was no significant difference in means between both features. Our hypothesis for this lack of difference is that our data may not include all relevant concept codes important for comorbidity and mortality assessment.

Variable	Point	Number of patients (%)
Age		
40≧Age	0	11317(7.2)
50≧Age>40	1	30094(19.3)
60≧Age>50	2	41781(26.8)
70≧Age>60	3	33897(21.7)
80≧Age>70	4	27706(17.7)
Age>80	4	11356(7.3)
Myocardial infarction	1	305(0.2)
Congestive heart failure	1	966(0.6)
Peripheral vascular disease	1	167(0.1)
Cerebrovascular disease	1	1726(1.1)
Dementia	1	193(0.1)
Chronic pulmonary disease	1	3143(2.0)
Rheumatic disease	1	220(0.1)
Peptic ulcer disease	1	6414(4.1)
Mild liver disease	1	5184(3.3)
Diabetes mellitus without end-organ damage	1	7775(5.0)
Diabetes mellitus with end-organ damage	2	732(0.5)
Hemiplegia	2	118(0.1)
Renal disease	2	1053(0.7)
Any malignancy*	2	10737(6.9)
Lymphoma	2	267(0.2)
Leukemia	2	95(0.1)
Moderate liver disease	3	846(0.5)
Metastatic solid tumor	6	6135(3.9)
Acquired immunodeficiency syndrome (AIDS)	6	2(0.0)

*Patients with more than one type of cancer in this study population.
Renal disease: chronic glomerulonephritis; nephritis and nephropathy; chronic renal failure.
Mild liver disease: chronic hepatitis; alcoholic cirrhosis; biliary cirrhosis. Moderate liver disease: liver diseases with cirrhosis-related complications.

Figure 1: Charlson Comorbidity Index (CCI) Scoring Criteria

At the end of Step 3, we successfully populated a demographic data set of applicable features including person ids, outcome (met recorded as 1 or unmet recorded as 0), age (scored by CCI), gender source value, ethnicity source value, and patient age at last visit. In Step 4, we will use our populated demographic data set to merge all conditions for a given patient. Noting specific conditions for our training set is important for our machine learning model to identify

patients whose conditions may put them at a higher risk for mortality 180 days after their last visit. Missing conditions were filtered out of our data set and concept ids were converted into strings for ease of matching with patient ids. Our approach to extracting conditions for a given patient and merging this data with our populated demographic data involved looping over each patient index and referencing the data frame location by matching the person index and string of a given concept id. Populated conditions for each patient were merged onto our demographics data set.

Having pertinent fields for our training set such as demographic data (age, gender, race), calculated CCIs, and conditions for each unique patient, we can now train our model. In Step 5, we imported all relevant packages and libraries to train our machine learning model including tools from pandas, numpy, matplotlib, sklearn, and xgboost. In our first attempt at training our model, we used all selected features in Steps 1 through 4. Having a binary outcome – 1 signifying a patient dying 180 days after their last visit and 0 signifying a patient not dying within 180 days after their last visit – we decided to perform a logistic regression. After splitting our data into training and testing sets and running our data on a logistic regression model, we found a high accuracy (95%) with low AUC-ROC curve (0.47) indicative of mediocre model performance selecting true predictors of mortality within 180 days post last recorded exam date. Essentially understanding that our model was using random guessing and predicting a majority of class 0 (outcome = 0) over class 1 (outcome = 1), we decided to continue with a more scientific approach to training our data set focusing on patient conditions.

Seeing room for improvement, we decided to run a second attempt using CCI and binary features for serious medical conditions. In our next attempt at training our model, we set a variance threshold to be 0.95 aiming to remove all features showing low variance from our dataset that are of no great use in modeling our training data. However, after performing our second attempt with a variance threshold, we found no significant improvement in our model seeing a slight increase in our AUC-ROC curve from 0.47 to 0.50.

Problem solving further, we decided to explore the sampling distribution of our data. We noted our data was skewed, having only 520 patients out of 97918 make up class 1 (outcome = 1). Imbalanced data in machine learning can result in skewed predictions favoring the majority class. Having a heavily imbalanced data set affecting the performance of our model, we strived to re-sample our data in order to counter class 0 predictions. To achieve this, we considered both undersampling and oversampling our data. In undersampling, we keep every training sample of the under-represented class and pick a fixed number of samples from the over-represented class. In contrast, oversampling generates ‘new’ synthetic samples from the under-represented class to balance the distribution. In our third attempt at training our model, we decided to try undersampling the data so that our class 0 sample size was more comparable to the class 1 sample. In undersampling our data, we selected all class 1 samples as well as 9480 random class 0 samples with binary features covering all of conditions, observations, drugs from the 10,000 patient training set, CCI, and patient demographics. From running our model with an

undersampled data set, we observed an improvement in our model. To narrow our focus even more and improve the performance of our model, we decided to only include the most important features showing a significant impact on our model's predictions (see Figure 2).

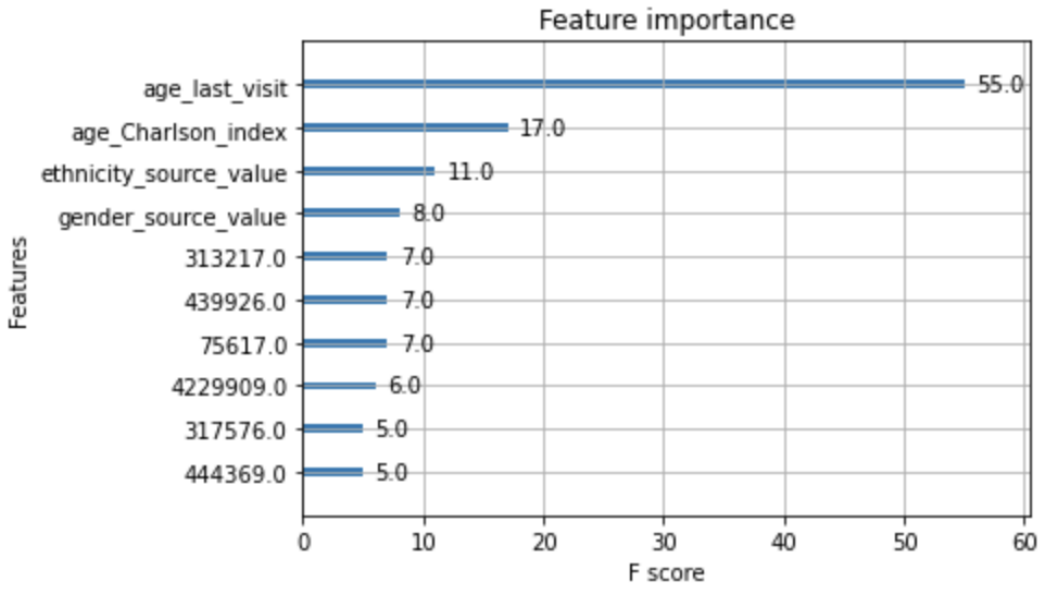

Figure 2: Feature Importance calculated using F1-score

With an extremely large population of class 0 outcomes, we initially thought to increase our class 1 sample size by oversampling our data. To accomplish this, we investigated incorporating the Synthetic Minority Oversampling Technique (SMOTE) from imblearn. SMOTE works by utilizing a k-nearest neighbor algorithm to create synthetic data points used to oversample our current class 1 data set. SMOTE first starts by choosing random data points from the minority class. Then, k-nearest neighbors from the data are set. Synthetic data is then made between the random data and the randomly selected k-nearest neighbor. Ultimately, this method did not prove to be effective as we were not working with a low dimensional feature set.

In Step 6 of our approach to creating an end-to-end machine learning workflow to predict mortality within 180 days of last visit, we finalize our test set. Creating our test set involved incorporating all methods in feature extraction from our training model to ensure that our testing group had the same selected fields as our training group.

Finally, in Step 7, we fitted our training model on the test set returning probability statistics and final AUC-ROC curve of our model's performance. Using xgboost to fit our model with n_estimators at 50 and a max depth of 5, we obtained an accuracy of 95% with an AUC-ROC curve of 0.48.

Results

Multiple attempts at fitting our model based on various ranges of feature selection and adjusted parameters proved to be unsuccessful in creating a highly performative machine learning model. As stated above, our final attempt to improve the predictive accuracy of our model resulted in a high accuracy of 95% but low AUC-ROC curve at 0.48. A low AUC-ROC curve is a good indicator of our model randomly guessing true positives in our data set, essentially favoring the majority class where, in our case, the outcome is 0 or people who did not die within 180 days of their last visit. A high accuracy also favors this result, telling us our model is highly acute towards selecting a majority class when working with a class-imbalanced data set where there is a significant disparity between the number of positive and negative labels.

Conclusion and Future Work

Generation of large amounts of healthcare data with the expansion of EHR systems has led to the possibility of predictive machine learning models capable of aiding clinicians in foreseeing a patient's status or response, thus enabling more proactive and precise medical practice. Our main objective for this project was to implement an end-to-end machine learning model to determine whether a patient will pass away within 180 days of having their last (available) exam utilizing EHR data. Selecting relevant clinical features and fitting our training set using xgboost, we did not obtain a high performing model. Low performance of our model was possibly due to the immense amount of data needed to implement and train our model on which we decided to only include a portion of to save time and memory for the purpose of this project. Future work to refine our model would include implementing data from the observations and drugs table. Also, utilizing better cloud computing methods to handle large amounts of data to train algorithms faster. Incorporating longitudinal data to map a patient's health at any given time period instead of analyzing all conditions at once may give a more accurate prediction of mortality risk. Additionally, more emphasis on health indexes for patients with important concept codes related to the CCI index may play a significant role in a patient's risk for mortality while also reducing the dimension of our feature set by encoding many health conditions into one index. Assessing the features in our model, we recognized that the way gender and ethnicity were encoded may have played a role in mortality prediction as one value may have been given more weight in our model than another. A fix would be to one hot encode demographics. In the future, reducing the dimensionality of the feature space using health indexes and including features from the other data sources may prove successful in advancing the overall performance of our model while also allowing us to potentially utilize oversampling methods to balance the dataset. Our current method does not distinguish between the two classes signifying that the features selected do not provide much, if any, impact on the health expectancy of a patient following a medical visit.

References

Charlson, M E et al. “A new method of classifying prognostic comorbidity in longitudinal studies: development and validation.” *Journal of chronic diseases* vol. 40,5 (1987): 373-83. doi:10.1016/0021-9681(87)90171-8

Chen, Yw., Li, Yj., Deng, P. *et al.* Learning to predict in-hospital mortality risk in the intensive care unit with attention-based temporal convolution network. *BMC Anesthesiol*22, 119 (2022). https://doi.org/10.1186/s12871-022-01625-5

“Concept: Charlson Comorbidity Index.” *Concept: Charlson Comorbidity Index*, 17 Nov. 2021, http://mchp-appserv.cpe.umanitoba.ca/viewConcept.php?printer=Y&conceptID=1098.

Guo, A., Pasque, M., Loh, F. *et al.* Heart Failure Diagnosis, Readmission, and Mortality Prediction Using Machine Learning and Artificial Intelligence Models. *Curr Epidemiol Rep*7, 212–219 (2020). https://doi.org/10.1007/s40471-020-00259-w

“Machine Learning in Python.” *Scikit-Learn*, https://scikit-learn.org/stable/.

“View Reference: MCHP Concept Dictionary and Glossary for Population-Based Research: Max Rady College of Medicine: University of Manitoba.” *View Reference | MCHP Concept Dictionary and Glossary for Population-Based Research | Max Rady College of Medicine | University of Manitoba*, http://mchp-appserv.cpe.umanitoba.ca/viewReference.php?referencePaperID=71546.