Everyone wants the best healthcare for themselves and their loved ones. However, making informed comparisons of hospitals is a monumental task. Besides the sheer amount of research necessary, even finding useful metrics to compare often proves to be a fool's errand. Individuals are therefore faced with the problem of comparing hospitals using rating systems available online, most of which are user-unfriendly, opaque in their methodology, and leave no room for individual preference. In this paper, we aim to usefully quantify a measurement of hospital quality that can be tailored to an individual and used to compare hospitals near and far through the user-friendly "HealthSearch" program.

The first part of our model aims to measure a mortality ratio that effectively accounts for what portion of patients incoming to a given hospital might be expected to die due to factors outside the control of that hospital. In other words, it creates a risk-adjusted mortality rate known as a Hospital Standardized Mortality Ratio. We computationally fit a logistic regression to find the mathematical relationships that can predict whether a given patient is likely to die based on seven known features about them including insurance type, length of stay among others. This regression is "trained" on a large dataset from New York State's SPARCS program. Comparing the number of actual deaths to the number predicted by the model yields a measure of where a hospital's mortality rate falls in relation to the average. However, our review of the existing literature showed that this ratio is untenable as the sole basis for ranking hospitals.

To account for this, we next calculated a more holistic and advanced hospital quality metric using a latent variable model. This model takes into account mortality, patient experience, patient satisfaction surveys, and other groups of measures to assign a quality score to a given hospital. It does this by performing a probabilistic computational regression across over fifty measures from the Center for Medicaid and Medicare Services. This regression aims to determine a series of measure loading coefficients, which describe the correlation between a given measure and an assumed "latent" variable, called the group score, that is an unobserved, unmeasured abstract dimension of quality for the group that measure falls into. These scores are then obtained by transforming a dataset of measures for a given hospital using the measure loading coefficients. Finally, we take a weighted average of these group scores with weights determined both by comparative research and user input. This algorithm was coupled with the user-friendly HealthSearch computer program that gives the user a short survey on their healthcare preferences to enable a high degree of personalization and a wide variety of use-cases.

We picked two test cases: the five hospitals closest to a randomly selected location and the best hospitals of all those within 60 miles of that location. We performed both these tests using three sets of weights to model user differences. Having selected Wheaton, IL as our location, we found that Advocate Good Samaritan Hospital was the best in all cases for the first test, and Rochelle Community Hospital was the best in all cases for the second test. To test the sensitivity of our model, we analyzed the regression coefficients, finding that no individual measures have predominant control of the group scores. We also explored the application of this model to an alternate use-case analyzing the regional quality of healthcare in the US.

Our model is built on an exhaustive review of the literature surrounding hospital quality metrics and ranking, with a focus placed on individualizing our metric and incorporating the strongest aspects of existing hospital quality rankings.

HealthSearch: How to Choose the Best Hospital for You

To whom it may concern,

Unfortunately, most people that end up in a hospital don't get to choose where they are treated. However, in the case where a trip to the hospital is premeditated, one would likely prefer to place their wellbeing in the hands of the best medical facility available. While this choice may seem simple, the decision quickly becomes daunting when one begins to consider all of the factors that contribute to overall hospital quality: mortality, number and experience of physicians, timeliness of care, readmission rates, and imaging technology, to name a few. The challenge of selecting a hospital can overwhelm even the most health-conscious individuals, especially when popular metrics such as the hospital standardized mortality ratio are apocryphal as best. Thankfully, our team has created a model that does all the hard work for the patients to facilitate simple and effective decision-making.

Our team created the comprehensive, easy-to-use program HealthSearch, which provides users with hospital ratings that are tailored to their individual preferences and needs. HealthSearch first asks the user for their address, and then asks them to complete a short survey of seven questions. The first six questions evaluate how important qualities pertaining to personal experience and hospital structure are to the user. The last question inquires about the severity of the user's symptoms as to gauge how severe one's illness might potentially be. HealthSearch analyzes the survey responses, processes it with data from nearby hospitals, and outputs a list of the closest hospitals and their quality scores, adjusted to the priorities of the user. The user can also alter the search radius to yield quality scores for hospitals at whatever distance they please. In this way, HealthSearch surpasses other static models by shifting its priorities based on user-preference. Our goal with HealthSearch is to transcend the inherent subjectivity of hospital ranking systems and provide quality scores that are specific to the exact needs and desires of the individual patient.

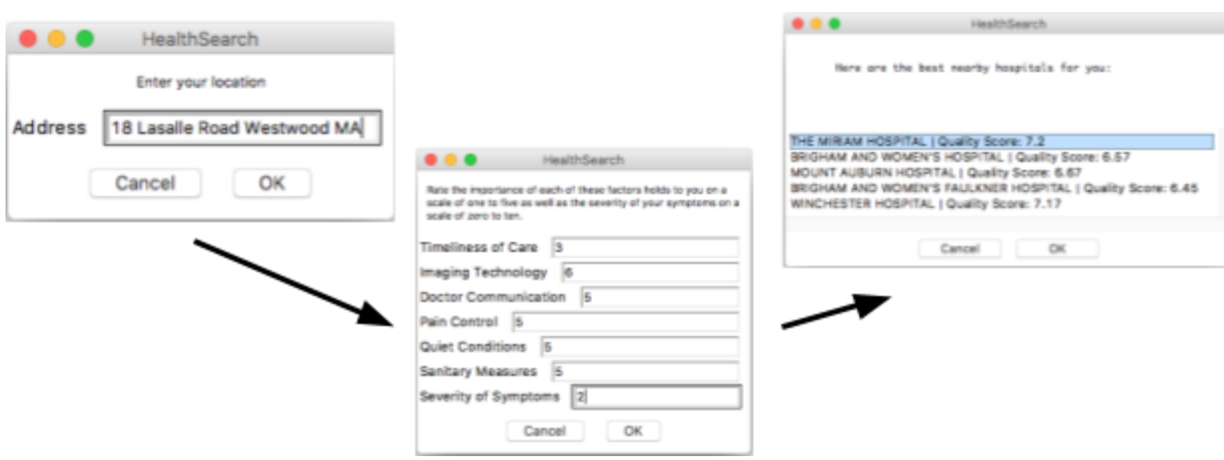


Figure 1: The HealthSearch Application Process

On top of personalizing hospital rankings with HealthSearch, our model has many other attributes that set it apart from existing models. In the table below, we compare aspects of our model to those of four organizations with existing hospital ranking models, all of which are further explained throughout the paper.

	Medicare	US News	Health Insigh t	Leapfrog Group	Health-Search
Over 50 different hospital quality measures utilized	✓	✓			✓
Patient opinion survey data incorporated	✓				✓
Missing data for individual measures replaced by national mean		✓			✓
Fixes missing measure groups by re-weighting remaining groups	✓			✓	✓
Stressed importance on patient experience			✓		✓
Weights adjusted based on user input					✓
User-friendly desktop application					✓

Table 1 - Comparison of Hospital Rating Services

If a patient is looking across the country for hospitals, our model also proves highly versatile, allowing one to find the states and even the counties with the best hospitals. In our paper, we create a heatmap showing the U.S. states with the best average hospital rankings. This ability allows our model to direct patients to the areas of the country where they can expect the most favorable outcomes.

HealthSearch draws on a massive repository of data on more than 3500 United States hospitals from the Center for Medicaid and Medicare Services (CMS). This data is interpreted through an advanced statistical model which aims to identify hidden innate dimensions of quality contained within discrete groupings of hospital measures. This statistical model is fit to the huge CMS dataset using the advanced Factor Analysis model contained with the artificial intelligence library scikit-learn.

While other rating systems reassign weights in the case of missing data, corrupting user choice and unduly weighting the data which is present, HealthSearch instead replaces the missing data with the national mean for that measure, retaining the element of user choice and keeping weights equal.

HealthSearch provides a powerful, robust hospital searching and ranking tool in a clean, functional, and easy-to-use package. It surpasses similar tools by including over 50 hospital quality metrics, putting patient experience at the forefront of our measures, employing the sophisticated latent variable technique to account for potentially hidden variables affecting hospital quality, and offering a personalizable algorithm, accessible through a convenient user-interface, which weights factors based on user preference. We believe HealthSearch can enable patients to take control over their own healthcare and provide them with much-welcomed autonomy in a system which has long treated patients as mere numbers.

Sincerely, Team
2018032

IM²C 2018: The Best Hospital

Team # 2018032

1 - Introduction

Health is the root of a prosperous life, and as such, people want to receive the best care that they possibly can. However, key decisions—in particular, choosing where to receive treatment—often fall on lay individuals who are ill-prepared to make them. A quantitative means of comparing the relative quality of hospitals is highly valuable in aiding the challenges of choosing the "best" hospital. In this paper, we aim to model the quality of hospitals in terms of quantitative performance assessments and the needs of the individual seeking care.

Ultimately, the primary aim of anyone seeking care is to survive and regain health. As a result, hospital-standardized mortality ratios (HSMRs), which control for the underlying health of a hospital's case-mix, are commonly used in comparing hospitals. However, research shows that this metric has significant shortcomings, especially when taken alone (van Gestel et al., 2012), and that it is necessary to account for a multitude of areas (e.g. patient experience, timely and effective care, etc.) in determining quality. Existing hospital quality rankings, however, are far from comprehensive and often disagree due to stressing different measures for importance (Austin et al., 2015). Our model addresses these criticisms by tailoring its focus towards a user's preferences, and by incorporating a variety of the best aspects of existing quality rankings.

To determine the best hospital for a specific individual, we have developed a latent variable model algorithm, care preferences survey, and a user-friendly application. We also assess sample results and done thorough sensitivity analysis of our model. In addition, we explore the utility of our measure in comparing regional health quality trends.

1.1 - The Problem

We were tasked with developing a mathematical model to measure the quality of a hospital, ultimately resulting in information that could allow a non-technical user to make an educated decision as to where to seek treatment. This model should be applicable both when comparing 5 local hospitals or, when willing to travel, choosing out of 50 hospitals.

Summarized, this problem asks 3 things of us:

1. Quantify the quality of a given hospital based on mortality.
2. Quantify hospital quality holistically, including a multitude of factors.
3. Considering these measures of hospital quality, provide a user-friendly system for aiding in picking the best hospital.

1.2 - General Assumptions

- ➔ Hospital quality can be comprehensively modeled through available data concerning surveyed patient experience, structural care metrics (such as timeliness and effectiveness of care, as well as efficient use of imaging), and patient outcomes. ***Justification:*** We cannot mathematically model unmeasured or unmeasurable quantities that may or may not affect a hospital's quality. However, our model's creation of a composite measure potentially accounts for the latent dimension of unmeasured factors (Landrum et al., 2000)

- ➔ Hospital quality relates to the treatment of illnesses that are physical, not mental. ***Justification:*** Our model assesses hospitals' effectiveness in treating patients' physical ailments, as suggested by the problem. Further, mental institutions are usually separate, or at least administratively separate, from the hospitals that we are evaluating.
- ➔ All patients, regardless of demographic, are treated equally by doctors. ***Justification:*** Social discrimination is not accounted for by the existing bodies of data, and thus cannot be included in our model. In addition, hospital ethics rules and internal regulation acts to minimize such discrimination
- ➔ Reported measure data is broadly accurate and free of major errors or omissions. ***Justification:*** The organizations from which we draw data have data-collection regulations and regular audits (Yale/CORE, 2017).

1.3 - General Definitions

Hospital Quality: A quantifiable innate trait of hospitals which can be modeled through its unique and consistent effect on observed metrics for that hospital (i.e. risk-adjusted mortality rates).
Inpatient: A patient who stays in a hospital while under treatment and the most typical individual seeking hospital care.

2 - Modeling Mortality-Based Hospital Quality

For this first section of the model, we set out first to define precisely when a death is considered evitable, and then to determine how hospitals are currently ranked based on mortality. A thorough review of past and current scholarship led us to the conclusion that Hospital Standardized Mortality Ratios (HSMRs)—ratios between observed deaths and deaths predicted based on the inherent risk of incoming patient cases—are highly flawed and not viable as the sole basis for deciding between or ranking hospitals. Acknowledging the problems inherent to HSMRs, we explore the use of a predictive model to find an expected mortality rate for a cohort of patients with treatable conditions and compare that to their observed mortality rate.

2.1 - Specific Assumptions

- ➔ Newborns, patients who leave against medical advice, stillborns, and patients who receive palliative care live or die independent of hospital quality. ***Justification:*** The Canadian Institute for Health Information excluded similar groups of patients from their HSMR study for the same reason.
- ➔ Patients within each age group are just as likely to die as other patients within that age group, all other things being equal. ***Justification:*** Hospitals collect age data for their own analysis. If age groups and not exact ages are sufficient for medical professionals, then it should be sufficient for our use.
- ➔ The independent variables we regress across are linearly independent. ***Justification:*** If an independent variable were linearly dependent, then it would be possible to write them as

some combination of other independent variables, which would create statistical anomalies within our model that are not present.

2.2 - Specific Definitions

Hospital-Standardized Mortality Ratio (β): The ratio of actual deaths to predicted ones within a hospital.
Evitable (Amenable) Death: A death, which, with proper medical treatment being dispensed of in a timely and effective manner, should not have occured from a treatable illness.
Actual Deaths (A_d): The actual number of deaths reported at a hospital.
Probability of Death ($P_{d/p}$): The probability of a given patient dying while in the hospital.

2.3 - Literature Review of Standardized Mortality Ratios

Whether a patient lives or dies is not strictly controlled by the quality of the hospital at which they are treated; it is in fact more often determined by the specific details of their case prior to treatment. When ranking hospitals by mortality, one must differentiate between an evitable and inevitable death. Evitable is defined as "capable of being avoided." This definition gives rise to several questions, as one must then identify how and when death can be avoidable. Ascertaining when a death is evitable, or, in the language of the medical and epidemiological communities, amenable, has proven a difficult task for researchers. The modern consensus is that an amenable death is one where a person with one of 34 specific, treatable conditions should not have died (Nolte and Mckee, 2004). In other words, with proper medical care being dispensed in a timely and effective manner, death should not have occured.

In this vein, many countries, regions, and individual hospitals use what is called a Hospital Standardized Mortality Ratio (hereafter referred to as HSMR) to evaluate medical centers based on death rates. Simply, an HSMR is a ratio of the observed number of deaths to the predicted number of deaths, which are both calculated with respect to the same population of patients. The numerator in the ratio is the number of recorded deaths in a hospital, and the denominator is the sum of the probabilities of patients within the hospital dying (Canadian Institute for Health Information, 2016). If the expected number of deaths is the same is the actual number of deaths, then the HSMR is 1. Numbers that are above 1 indicate that more patients are dying than are expected based on their circumstances, and numbers that are below 1 indicate that patients are dying at a lower rate than expected.

HSMRs, while widely used, have been under much scrutiny for many years. The paper "Hospital Mortality: When Failure is Not a Good Measure of Success" states that the "omission of important clinical information from routinely collected data" can cause serious discrepancies in the calculation of a hospital's HSMR (Shojania et al., 2008). Additionally, GE Rosenthal found that various types of HSMRs may not be excellent indicators of hospital quality (Rosenthal et al., 1998). In perhaps the most scathing paper on HSMRs (Van Gestel et al., 2012), the authors found that HSMRs often do not uniformly account for disease severity, which can lead to discrepancies between HSMRs of different hospitals. Furthermore, they also point out that hospitals who on the whole receive more referrals than give referrals have higher HSMRs. While the logistic issues with HSMRs are mountainous, we in our model attempt to improve upon them.

2.4 - Quantifying Predicted Mortality with Logistic Regression

In order to predict the probabilities of binary outcomes—living and dying, in our case—we use a logistic regression model. Logistic regression is a nonlinear extension of multiple linear regression. It works in the following way: suppose the probability of dying is p, then by definition the probability of staying alive is $1-p$. We define what we call the odds of dying as $p/(1-p)$. Setting that equal to a linear combination of n independent variables $\{x_i\}_{i=1}^n$, whose coefficients $\{\alpha_i\}_{i=1}^n$ are weights that describe how much impact the variable they are multiplied by has on p, gives:

$$\frac{p}{1-p} = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n \tag{1}$$

Unfortunately, the values of p have a larger range than can be interpreted as a probability. In order to restrict them, we take the natural logarithm of the left side of the equation, and solve for p, which is:

$$p = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n)}} \tag{2}$$

The goal of logistic regression is to find out what the values of $\{\alpha_i\}_{i=1}^n$ are. When implementing this regression in a python program, the coefficients are calculated by converting the probability expression into one that makes sense computationally. Firstly, note that that the $\{x_i\}_{i=1}^n$ values become column vectors. The coefficients $\{\alpha_i\}_{i=1}^n$ become the weights w. We can then rephrase our original expression for p into a conditional probability, like so

$$p(y=1|X;w,c) = g(wX_i^T + c) = \frac{1}{1 + e^{-(wX_i^T + c)}} \tag{3}$$

where c is an intercept that was previously denoted by α_0. This equation is an equivalent formulation of the formula for p, except now it is expressed more clearly as a conditional probability for someone dying given a set of independent variables whose scalar multiples are coefficients that express how heavily each variable affects the chance of dying. These coefficients are calculated by completing the following minimization:

$$min(w,c)\frac{1}{2}w^T w + C\sum_{i=1}^{n} log(e^{-y_i(wX_i^T + c)) + 1)}) \tag{4}$$

A full derivation of this equation can be found in the appendix. Once the values of the coefficients are found through computational regression, they can be filled in, and the value of p can be computed for any given patient.

2.5 - METRIC: Hospital-Standardized Mortality Ratio (HSMR)

We arrived at our regression measures by excluding measures that were directionless, difficult or impossible to quantify, overlapped with other measures, or were simply organizational and did not represent relevant data. In addition, we standardized the data into statistical z-scores through the process detailed later in section 3.4. After regressing over these measures, the model gives the predicted probability of death for each patient. Summing over these probabilities yields the expected deaths. Given the number of expected deaths, and the number of actual deaths which occurred at the hospital, we can calculate the HSMR like so

$$\beta_h = \frac{A_h}{\sum_{i=o}^{n} P_{h_i}}$$

[5]

where β_h is the HSMR, P_{h_i} is the predicted probability of death for a given patient, and A_h is the number of actual deaths.

3 - Modeling Holistic Hospital Quality for Decision-Making

In this section, our task was to understand how the "quality" of a hospital can be assessed—what makes a hospital good, bad, or even the best. A hospital's main purpose is to preserve the lives and wellbeing of its patients, and many quantifiable factors can measure the success these objectives. Our model finds underlying scores for various dimensions of quality through a sampling-variation-adjusted latent variable model. These scores are then taken together as a weighted average to yield an overall hospital quality score.

3.1 - Specific Assumptions

- ➔ Each group reflects a single distinct underlying aspect of quality. Each measure contributes to only one group score. ***Justification:*** Though it is possible that measures reflect more than one underlying aspect of quality, measure groups show a high degree of covariance (Yale/CORE, 2017) meaning that the constituent measures significantly correlate with others in their group.
- ➔ Patient surveys properly and fully reflect patient experience in a given hospital. ***Justification:*** Patients are ultimately the final arbiters of whether their experience was positive, so we must assume that this feedback carries a certain degree of objectivity in order to quantify patient experience in our model.
- ➔ Hospital measure group scores follow a distribution with unit variance. ***Justification:*** This is an innocuous assumption resulting from the standardization of measure values, therefore implying that the latent variable is also standardized.

3.2 - Specific Definitions

Standardization (or Normalization): Process of converting a measure into a dimensionless quantity. This quantity is the number of standard deviations a score is above or below the average score.
Measure (y_{khd}): Standardized quantitative value (y) of a metric (k) describing some aspect of a hospital's (h) performance.
Measure Group (d): A group of hospital measures considered conceptually similar.
Weighting (W_d): Weighting considers the importance of a measure group in the whole, and unequal weighting implies some groups contribute more than others.
Latent Variable Score (α): An unobserved, inferred quantity reflecting some latent trait.
Loading (γ_{kd}): In a structural equation model, the regression coefficient between a measure and the measure group's latent variable score.
Z-score: The number of standard deviations above or below the population mean a value is.
Sampling Variation: Variation in a given statistic between samples.

3.3 - Literature Review of Hospital Ranking Systems

Many organizations have already created models to quantify the quality of medical institutions across the nation. Medicare's five star ranking system *Hospital Compare*, for example, assesses the majority of hospitals in the United States. In the Medicare (or CMS, Center for Medicare & Medicaid Services) model, an integral-based latent variable model is used to identify correlations between measure which are grouped together. (Yale/CORE, 2017).

US News & World Report has a different model for ranking hospitals based primarily on four components: structure, outcomes (mortality and readmission), expert opinion, and safety. Their data, as well as Medicare's, is comprised of both quantitative measurements and qualitative surveys. US News does not use a latent variable model; rather their score is calculated by taking the weighted sum of the four group values, which are comprised of z-scored values for each measure (Olmsted et al., 2017).

We found two other organizations that also assess hospital quality with a weighted sum method, from Health Insight and Leapfrog Group. Each organization's weights are shown below, with structure included as effectiveness for the purpose of comparison.

	Medicare	**US News**	**Health Insight**	**Leapfrog Group**
Outcomes	44%	37.5%	10%	50%
Safety	22%	5%	45%	
Timeliness	4%			
Effectiveness	4%	30%		50%
Experience	22%		35%	
Imaging	4%			
Expert Opinion		27.5%		

Table 2 - Hospital Quality Weighting Schemes

As one can see, there is significant variation among not only the weights, but the groups each model chooses to assess. As each model has clear justification for their respective weights, the discontinuity across the board strikes us as alarming, as it brings up the issue of profound subjectivity in hospital ranking systems.

Our model draws from each of the four aforementioned models, external research, and a novel approach to individualizing our quality score to create a hospital ranking system that minimizes subjectivity, maximizes accuracy, and tailors one's hospital search to their specific needs and preferences.

3.4 - Standardizing, Imputing and Grouping Hospital Measure Data

In order to ensure that our model can have real-world usability, we base our holistic hospital quality score on widely-available hospital level data and draw out the mathematical relationships between those available measures through computational regression. This methodology presents two problems: one, many hospitals only have data available with several missing measures, and two, these measures represent a wide variety of measurements and units, where in some cases higher values are better (i.e. flu immunization rates) and in other cases lower values are better (i.e. time to being seen by a doctor in the emergency room). To account for these problems, we use the statistical techniques of value standardization and mean imputation.

Our first step was to organize measures into groups in which the measures would be expected to vary together. Given the available data and grouping analysis done previously by CMS, we decided to separate our measures into seven groups: mortality, readmission, safety, timeliness, effectiveness, patient experience, and imaging technology. The specific measures in each group can be found in the appendix.

Once our measures were grouped, we had to standardize the input values. Value standardization is the process of taking diverse sources of differently scaled and distributed data and converting them into dimensionless quantities that follow the shape of a normal distribution, with a mean of zero and variance of one (unit variance). These dimensionless quantities are known as z-scores, defined as the number of standard deviations above or below the population mean a value is. In our case, the population mean is the average for that measure across all the hospitals in our data source, which is the national mean. The standardized measure is therefore given by

$$y_{khd} = \frac{y_{khd_0} - \mu_{kd}}{\sigma_{kd}} \times \rho_{kd} \qquad [6]$$

where y_{khd} is the standardized measure score of metric k in measure group d for hospital h, μ_{kd} and σ_{kd} are the population mean and standard deviation for that metric, and y_{khd_0} is the initial raw score. This computes the z-score, while ρ_{kd} is a direction scaler ($\rho_{kd} \in 1, -1$) that reverses the direction of the measure if it is a metric where lower values are preferred, making it so that for all standardized measures, higher is better. Through this standardization, we make it possible to compare metrics as disparate as mortality ratios and patient experience reviews within the same model.

Once the measure values are standardized, we fill missing measures through the process of imputation, substituting in the population mean (zero after the standardization). This process aims to prevent those missing measures from skewing the data by minimizing the impact a measure has on the quality score if data is nonexistent. This allows for more consistent and direct hospital comparison. That said, there is a risk of compromising our model's validity by including hospitals where all or nearly all of their measures for each group are just the filled-in average, contributing very little actual information to the model. To avoid this, we imposed a cut-off, requiring that a hospital to have at least three measures in a given group to be included in the latent variable regression for that group.

3.5 - Identifying Trends in Measure Groups with Latent Variable Modeling

We employed a technique known as latent variable modeling (LVM) to estimate a group score for the type of quality represented by each measure group. Each group has a separate LVM as we assume that the measures in that group, when taken together, represent an underlying condition of quality for that area, and that they will usually vary together.

Latent variable models are a type of approach in statistical modeling that assumes that each measure reflects an underlying dimension of quality. Our selection of measure groups reflects this. For example, we define a group called "mortality," containing several HSMRs for different procedures and conditions. LVM accounts for the correlation between these HSMRs; the more they vary with each other, the more influence they have on the derived latent variable. Since these latent variables are unobserved, they are inferred computationally from a probabilistic regression. A review of healthcare quality literature shows that the LVM method is preferred for composite measures (Landrum et al., 2000). LVM is also useful as it can account for sampling variation as detailed in section 3.6.

To derive the latent variable of a measure group for a given hospital, we first need to determine the relationships between each measure in the group and that latent variable, values which we will call the "loading coefficients" for that group. The observations (the group's measures) are assumed to result from a linear transformation of a lower dimensional latent variable and added Gaussian-distributed statistical noise. This variable is distributed according to a Gaussian with zero mean and unit variance. Starting with the dataset, a matrix of standardized measures $\mathbf{Y}_{hd} = \{y_{1hd}, y_{2hd}, \dots, y_{khd}\}$, a simple continuous LVM for $\mathbf{y}_{hd}$ is

$$y_{khd} = \gamma_{kd}\alpha_{hd} + \nu_{kd} + \epsilon_{khd} \tag{7}$$

where α_{hd} is the hospital-specific latent trait for hospital h and group d, ν_{kd} is an arbitrary measure-specific offset value (like a y-intercept), ϵ_{khd} is a random Gaussian-distributed noise term with covariance ψ ($\epsilon_{khd} \sim \mathcal{N}(0, \psi)$), and γ_{kd} is the measure-specific loading coefficient. Equation 7 is called a generative model as it relates how y_{khd} is generated from α_{hd}. Rewriting this equation using the different measure values as columns in the matrix $\mathbf{Y}_{hd}$ (alongside a matrix for each of the loading coefficients $\mathbf{\Gamma}$, offsets $\mathbf{N}$, and noise $\mathbf{E}$), we get

$$\mathbf{Y}_{hd} = \mathbf{\Gamma}_d\alpha_{hd} + \mathbf{N}_d + \mathbf{E}_{hd} \tag{8}$$

Our task is now to compute the loading coefficients in the "factor loading matrix" $\mathbf{\Gamma}_d$. Given a particular α_{hd}, equation 8 implies the probabilistic interpretation

$$p(y_{khd}|\alpha_{hd}) = \mathcal{N}(\gamma_{kd}\alpha_{hd} + \nu_{kd}, \psi)$$

[9]

meaning that measures should follow a distribution with mean $\gamma_{kd}\alpha_{hd} + \nu_{kd}$ and variance ψ.

To get a final probabilistic model for the measure, it is necessary to obtain a prior distribution for the latent variable. A straightforward assumption, based on the clean properties of the Gaussian distribution and the standardization of the measure values, is that $\alpha_{hd} \sim \mathcal{N}(0,1)$. This yields a Gaussian marginal distribution for measure k across different hospitals. One more simple assumption is of the structure of the error covariance ψ. The assumption that the error is distributed with a distinct covariance for each measure ($\epsilon_{khd} \sim \mathcal{N}(0, \sigma^2_{kd})$) yields a classical statistical model called "factor analysis." Given these probabilistic relationships and measure values for a very large population of hospitals, it is possible to calculate the loading coefficients for each measure in each group. Measures with higher calculated loadings have a greater impact on and association with the group score (α_{hd}) than measures with much lower loadings.

The parameters in the loading matrix were obtained by maximum likelihood estimation (MLE) using expectation-maximization (EM), a computational process which attempts to fit regression coefficients to maximize a likelihood function for each measure across every hospital (Pedregosa et al., 2011). As can be seen in equation 8, the loading matrix transforms the latent variable to the observed ones. Once it is determined, the transformation can be reversed and applied to the dataset for any given hospital in order to reduce the multi-factor matrix $\mathbf{Y}_{hd}$ to a single latent variable, the group score α_{hd}.

3.6 - Accounting for Measure Sampling Variation

A key challenge in regressing our LVM is the vast difference in the size of hospitals, meaning that some report measures based on many more cases, a large difference in precision known as "sampling variation." To account for this, we weight the contribution of each hospital h in the computed overall likelihood function for a measure k in group d by the metric

$$\omega_{khd} = \frac{n_{khd}}{\sum\limits_{h=1}^{N_{kd}} n_{khd}} \times N_{kd}$$

[10]

where N_{kd} is the number of hospitals reporting values for that metric and n_{khd} is the "denominator" or sample size for that metric. This metric is then used in the likelihood function L across all hospitals H across all measures K like so:

$$L = \prod_{k \in K} \prod_{h \in H} (L(y_{khd}))^{\omega_{khd}}$$

[11]

which is the function subject to maximization in our computational regression. This means that measures with a grouter precision are given more influence in the determination of the measure loading coefficients

3.7 - Weights for Measure Group Scores

The final and largest problem we face is adjusting the weight given to each measure group in order to provide the most accurate scale for hospital assessment. Our weights are:

Measure Group or Multi-Group (d)	W
Outcome (mortality, readmission)	30% (15%, 15%)
Structure (timeliness, effectiveness)	35% (17.5%, 17.5%)
Imaging	5%
Safety	5%
Patient Experience	25%

Table 3 - Measure Group Weighting Assignments

We assign the largest weight to structure—an amalgamation of timeliness, effectiveness, and imaging technology—because we believe it has the largest bearing on patient recovery and the provision of a positive and efficient hospital experience. We deduce this from Medicare's listed measures within these three groups, which include many preventative safety measures, pain regulation measures, and assessments of hospital efficiency.

We give imaging a weight of 5% due to the fact that it applies to a limited number of patients, and not all hospitals utilize technologically advanced imaging machines. We gave safety a weight of 5% as well because Medicare's safety measures are based on the likelihood of a patient acquiring one of a few specific diseases at the hospital.

We give patient experience a weight of 25%. While patient experience comes secondary to health and successful recovery, it is not to go undervalued. A working paper from the National Bureau of Economic Research (NBER) stresses the strong correlation between patient experience and outcome (Doyle et al., 2017). They write that hospitals with "timely and effective care processes" and "better patient experience" yield "a significant and meaningful decline" in the chance that a patient might die the following year . This correlation allows us to spread the weight fairly evenly among patient experience, structure, and outcome. We give outcome a weight of 30%, because while death prevention is the most important aspect of hospital operation, it is also accounted for by the other measure groups, as suggested by NBER.

If a hospital to be scored reports fewer than three measures in a given group, we consider that group to be "missing," as the data is too sparse to be useful. When this occurs, we re-weight the remaining groups so that they are re-proportioned equally by setting that group's weight to zero, and re-calculating each other group's weight by dividing it by the new total of all weights. The formula for this redistribution is:

$$W_d = \frac{W_{d_0}}{100 - W_{d_{\text{missing}}}}$$

[12]

where W_d is the new weight for group d, W_{d_0} is the initial weight, and $W_{d_{\text{missing}}}$ is the weight of the missing group.

Real-World Use: Tailoring a Quality Score to the User

The largest flaw in the model thus far, and in all existing hospital-ranking models, is that no two patients will prioritize the same aspects of a hospital experience, and thus a uniform set of weights to assign these priorities falters in relevance to the individual patient. In order to combat this, we took our model a step further by incorporating user-specific weights to tailor the hospital ranks to the individual.

3.8 - Re-Weighting Groups According to User Survey

We provide a seven question survey to patients to obtain valuable information as to what their hospital preferences are. Our model uses this information to redesign the weight metric to base new outcome, structure, and experience weights on their relative importance to the user. The first portion of the survey asks the user to rate the importance of six hospital qualities on a scale of one to five, one representing indifference, five representing extreme importance. The first three qualities relate to patient experience: doctor communication, pain mediation, and hospital quietness. The latter three qualities relate to hospital structure: timeliness of hospital, imaging technology, and sanitary measures. We follow the preference survey with asking users to rate the severity of their symptoms from zero to ten, which gives us insight into how crucial a hospital's mortality ratio may be to the recovery of that patient. The vectorial function for these weight adjustments is shown below.

$$W(S, q, E) = (W_O, W_S, W_E)$$

$$W(S, q, E) = \left(\left[58 - 6(E + S - 2) + q\right], \left[16 + 6(S - 1) - \frac{q}{2}\right], \left[16 + 6(E - 1) - \frac{q}{2}\right] \right)$$

[13]

In this function, W_O, W_S, and W_E are the adjusted weights for outcome, structure, and patient experience, respectively. The function inputs S, the mean rating of the three structure questions, E, the mean rating of the three patient experience questions, and q, the symptom severity rating. The 5% weight for safety is excluded from the function, as it remains constant because it accounts for the possibility of a patient acquiring a secondary illness at the hospital, which is independent of user preference. The 5% weight for imaging is also excluded, as it does not factor into user preference significantly enough to be adjusted.

Equation 13 allows us to scale the weights within set ranges to maintain maximum and minimum weights. For a case where severity of symptoms is rated at ten and the mean structure and experience scores are both one, our function outputs weights of 68% for outcome, and 11% for both structure and experience. In the case where symptom severity is ranked at zero and the mean structure and experience scores are both five, our function outputs weights of 20% for outcome, and 35% for structure and experience. Experience and structure survey responses carry more weight in the function because the correlation between the preferences and the measures is much more absolute than that between symptom severity and outcomes.

3.9 - METRIC: Holistic Hospital Quality Score

With the weight W determined for each group d, and with each group's score calculated through LVM, the hospital quality score for a hospital h is given by the weighted average of the group scores:

$$S_h = \frac{\sum_{d \in D} W_d \alpha_{hd}}{\sum_{d \in D} W_d}$$

[14]

In addition, after calculating the summary score for every hospital in our national cohort (represented by the matrix of scores $\mathbf{S}$), our scores were centered around zero with close to half positive and half negative. To make the scores more user-friendly, we rescaled the values to run from zero to ten, centered around five. Each score is now given by

$$S_h = \frac{[S_{h_0} - min(\mathbf{S})] \times 10}{max(\mathbf{S}) - min(\mathbf{S})}$$

[15]

4 - Model Implementation

4.1 - Mixed Sampling and Firth Regression of Evitable Deaths

In order to create our logistic regression, we required complete and thorough hospital discharge data. Robust patient-level data is only available for New York State through their Statewide Planning and Research Cooperative System (SPARCS) program, and thus our model is built from New York data only. However, the SPARCS dataset is large enough that our model is still generalizable to other states and even other countries given equivalent test data.

Newborns and patients who left against medical advice are omitted from the data, as the fact that these patients survive, at least in the discharge data, is irrespective of hospital quality. We account for certainly inevitable deaths by omitting stillborns and patients who received palliative care while in the hospital. More robust assessments of inevitable deaths are possible, but would require much more complete data on patient diagnosis, comorbidity, and age.

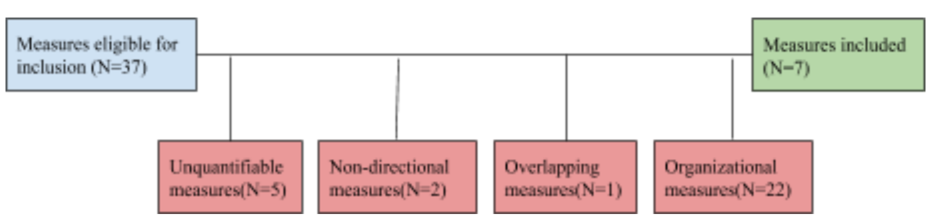


Figure 2 - HSMR Measure Selection Flowchart

The response variables our model regressed over were: age group, gender, race, urgency of admission, whether the procedure was medical or surgical in nature, length of stay, and primary insurance type. The latter of these was included as a measure of social deprivation, since insurance type is known to correlate with both socioeconomic status and patient condition (Weiner et al., 2017). Our criteria in selecting these measures is detailed in Figure 4.1—a list of all the measures eligible for initial inclusion can be found in the appendix. Measures were quantified when necessary, then standardized into statistical z-scores.

When creating our logistic regression to calculate HSMRs, we encountered a serious and unforeseen roadblock. Whereas most studies that attempt to create HSMR regressions use data from intensive care units, which have relatively high mortality, we used data from normal hospitals. This data was highly imbalanced, with deaths making up only 2.9% of patient outcomes, and produced an overfitted model which was highly accurate but entirely useless for predicting deaths.

In order to rectify this, we first employ a Firth Regression which introduces a more effective score function by adding a term that counteracts the first-order term from the asymptotic expansion of the bias of the maximum likelihood estimation (Wang, 2014). We then use a mixed-sampling method which combines the Synthetic Minority Over-Sampling Technique (SMOTE), in which data from the minority class (death) are over-sampled by creating "synthetic" samples from existing data, and the Tomek links removal method, which "cleans" the data by removing points for which there is no example z such that:

$$d(x,y) < d(x,z) \quad \text{or} \quad d(y,z) < d(x,y)$$

Where x and y are two samples and d is the distance between the two samples. These methods exhibited better precision and recall than similar methods, and were demonstrated (Bee WahYap et al., 2014) to perform better than either bagging or boosting methods. We also utilize a class-weighting scheme which increases the penalty for misclassifying the majority class and further improves prediction metrics. The model was created using the python libraries scikit-learn and imbalanced-learn.

4.2 - Factor Analysis of Hospital Measure Group Data

Following the same methodology as our selection of regression variables in section 4.1, we use 59 measures split across 7 groups (as discussed in section 3.4) from the CMS's public datasets used for their *Hospital Compare* system. Every measure, along with its group and loading coefficient, can be seen in the appendix.

For our latent variable model, we first source and standardize every available measure for all Medicare-participating hospital in the United States from publically available datasets using the Socrata API in the python programming language. This data is then grouped and missing values are replaced with the national mean where necessary. Once ready, this is fed as training data to the scikit-learn "FactorAnalysis" class. This performs a probabilistic regression of the loading matrix (the coefficients of which relate each measure to its group's score) through a maximum likelihood estimate. This takes a variety of computational approaches to maximize the likelihood function shown in equation 11.

Once the measure loading coefficients have been fit, the models are all stored. These models then can take in test data in the form of measures for some given hospital and transform them into the latent variable, the group score. The program computes a score for every hospital with available data in the US, and then uses the max and min scores to rescale every score to be between zero and ten.

4.3 - The HealthSearch Application

The final product of our holistic hospital quality analysis is the user-friendly HealthSearch program, as detailed in the attached introductory memo. HealthSearch first asks a user to enter their location, then to rate on a scale of one to five the importance of six different areas of quality, and finally, to rate on a scale of zero to ten the severity of their symptoms. HealthSearch then outputs the quality scores of the hospitals closest to the user within a configurable radius, allowing them to make informed decisions based on their own healthcare priorities.

5 - Results

5.1 - Calculating HSMRs with Patient-Level Data

In order to test the effectiveness of our HSMR model, we calculated the HSMRs of the five hospitals closest to the home of a theoretical individual trying to choose a nearby hospital who lives at 350 5th Avenue, Manhattan, NY:

Hospitals	HSMR
Lenox Hill	1.51
Mount Sinai	1.14
Bellevue Hospital	2.28
NYU Hospital	1.24
Mount Sinai Beth Israel	1.27

Table 4 - Hospital HSMRs

We found that each hospital was at least slightly above average in terms of mortality, which is consistent with their relatively low ratings on Hospital Compare. When comparing the five hospitals, Mount Sinai is the best, with an HSMR of 1.14, and Bellevue comes in last with an HSMR of 2.28. Bellevue’s place at the bottom of the list is clear when considering they have experienced 24,024 deaths over the year, the most deaths of any of the five hospitals, and a relatively small number of surviving patients.

These results are particularly interesting, as they indicate that the top hospitals in one of the wealthiest parts of New York City, which is already one of the wealthiest cities in the world, have HSMRs which demonstrate more people are dying than expected. At first, this may seem counterintuitive; however, there are in fact many logical explanations for why this is. Looking first toward Bellevue Hospital, we see that it has an absurdly high HSMR, but that can be explained by noting that Bellevue Hospital serves some of New York City's most underprivileged populations (NYU, 2012). A possible explanation for the low HSMRs of other hospitals, which in general do not just serve the underprivileged, is that in a large, dense urban area where people are in very close proximity, like midtown Manhattan, disease spreads more quickly, and has the potential to affect more people. Broadly speaking, these results show the deficiencies of HSMRs as a hospital quality metric for comparing hospitals, despite some legitimate (possible) explanations for these results.

5.2 - HealthSearch: Creating Weighted Group Scores with CMS Hospital Data

Using our model, we calculated the quality scores for the five hospitals closest to a randomly selected address: 350 Knoll Street, Wheaton, IL. We first found the five closest hospitals to this address and ranked them by quality. We calculated the quality scores of these hospitals for three different cases: the first using our initial weights, the second based on user input where effectiveness and structure were both rated very high, but symptom severity was rated very low, and the third based on user input where effectiveness and structure were both rated fairly low, but symptom severity was rated extremely high. The five closest hospitals to 350 Knoll St, ranked in order of quality score, are shown below, as well as their distance from the address.

	Initial Score	**Case 1 Score**	**Case 2 Score**	**Distance (mi)**
Advocate Good Samaritan Hospital	6.1	6.0	7.6	9.8
Central Dupage Hospital	5.7	5.9	7.0	1.9
Elmhurst Memorial Hospital	5.6	5.6	7.0	11.1
Edward Hospital	5.3	5.4	6.9	9.3
Adventist Glenoaks	3.8	4.3	5.0	8.7

Table 5 - Five Nearest Hospitals Quality Scores

As is evident in the graphs, the hospitals' ratings are the highest for case 2. This implies that this collection of hospitals has strong mortality ratios, whereas they have lower scores for patient experience.

If a patient is willing to travel farther for their healthcare, they can enter what distance they are willing to travel into HealthSearch, and they will be presented with the five best hospitals within that radius. The hospitals below have the highest ratings within a 60 mile radius.

	Initial Score	**Case 1 Score**	**Case 2 Score**	**Distance (mi)**
Rochelle Community Hospital	6.8	7.3	7.0	56.5

Northwestern Memorial Hospital	6.6	6.3	8.5	31.8
Delnor Community Hospital	6.5	6.3	8.2	11.5
Midwestern Regional Medical Center	6.4	7.2	6.6	67.5
Silver Cross Hospital	6.1	6.3	6.6	31.3

Table 6 - 60 Mile Radius Hospital Rankings

While Rochelle Community Hospital has a higher ranking than Delnor Community Hospital under the initial weight scheme, a patient may choose the latter because the convenience of proximity supersedes a .3 difference in quality. A patient in case one may choose to trek the extra twenty-five miles to go to Rochelle Community Hospital if the one point difference in quality is important to them. As one can see, being presented with this data allows the user to weigh their options and make informed decisions as to which hospital fits their needs the best.

5.3 - Model Extension: Using Quality Metrics to Compare Regional Health

The design intent for our model is to aid in the comparison of several geographically clustered hospitals, but what about across the nation? Pulling from our model's list of the top hospitals in the USA, we found the following correlations between our quality scores and popular review sources online. A few selected examples are shown below, all scaled to a range of 1-10.

Hospital	**Quality Score**	**Facebook score**	**Google score**	**Yelp score**	**Miscellaneous**
Park Place Surgical Hospital	10	9.6	9.6	10	A+ (Better Business Bureau)
Mayo Clinic Hospital	8.1	8.6	n/a	9	7.2 (Consumer Affairs)
New England Baptist Hospital	8.2	9.6	8.8	8	7.4 (Glassdoor)

Table 7 - Hospital Quality Rankings Correlations

As one can see, our model's quality score for a given hospital is generally reflective of the consumer consensus. We also found that of US News's top 13 rated hospitals in the country, 10 fall within our model's top 100 list. Of CNN's top 14 rated hospitals, 7 fall within our top 100. Since we have calculated a score for most American hospitals, we can map the average hospital quality score across every US State.

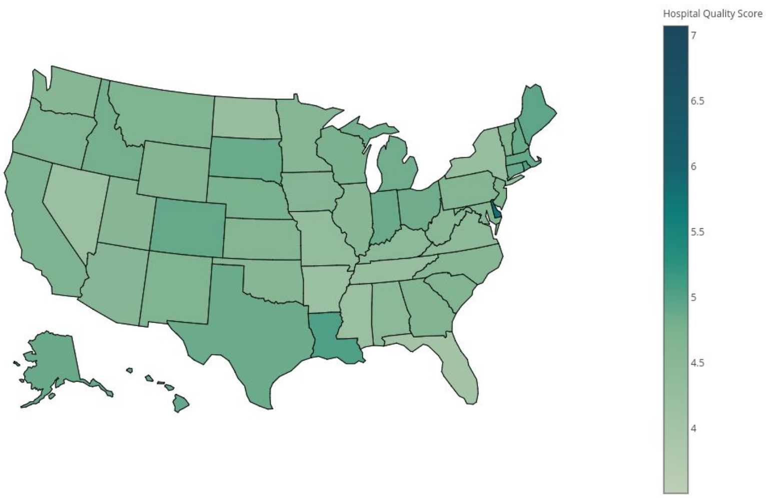


Figure 3 - Choropleth Map of Hospital Quality Score by US State

In Figure 3, the darkest shade represents the highest average quality score, and the lightest shade represents the lowest average quality score. Interestingly, it seems that most states and regions are relatively equal and close to the national average score of 5. Some areas are slightly skewed however, including New England, a region commonly known for its high household income. In addition, the American Southeast, containing states with lower end household incomes, has a lower-skewed average quality. The simplicity and generalizability of our quality metric show how our model can be of use not just to individuals choosing between hospitals, but also to other stakeholders interested in quantifying and comparing healthcare quality.

6 - Sensitivity Analysis

6.1 - Accuracy of Logistic Regression

	Precision	**Recall**	**F1-Score**
Alive	.68	.96	.80
Dead	.90	.41	.56
Avg/Total	.78	.73	.70

Table 8 - Logistic Regression Quality Metrics

Precision is the number of true positives over the number of total positives, recall is the number of true positives over the sum of the true positives and false negatives, and the F1-score is the harmonic mean of both the precision and the recall. Given these values, we can observe that our HSMR model is specific (high negative recall rate) but not very sensitive (low positive recall rate). Our overall precision is high, meaning that our model has relatively strong predictive power.

Variable	**Coefficient**
Age Group	0.76847133
Gender	0.47927852
Race	0.58733511
Length of Stay	1.72857484
Type of Admission	-0.02051861
APR Medical Surgical Description	2.07197458
Primary Insurance	-0.42070641

Table 9 - Logistic Regression Coefficients

Our model shows that whether a procedure is medical or surgical in nature is highly correlated with mortality, as medical procedures are less likely to be necessary to treat life-threatening conditions than surgical procedures. Length of stay is the second most strongly correlated variable. Likely, this is not due to a strong underlying correlation, but rather the fact that length of stay is the variable with the greatest variability, and therefore influences the model to find relationships where there are none. Primary insurance is negatively correlated with mortality, as better insurance implies greater socioeconomic status and allows one to afford more frequent doctor visits, increasing the chance of catching diseases early.

6.2 - LVM Measure Loading Coefficients

Since our LVM is fit through a regression, we computationally determine the relationship between each input measure and the latent variable group score as our first step. This loading coefficient shows the sensitivity of the group score to that particular measure, the larger its magnitude, the more sensitive the model is to it. The full regressed loading coefficients for every measure can be found in the appendix. These values show an interesting trend. Some groups (mortality and experience) show a comparable coefficient for every measure, showing that they all contribute more-or-less equally to the group score. The remaining groups show a mix of behaviors, with some measures being grouped at the high and low end. Some highly specialized measures, such as the interval of usage for specialized imagery, show extremely low coefficients, a logical result of their lack of strong correlation with other measures due to their specific nature.

Overall, the robustness of our LVM is demonstrated in the large variety of distinct measures that contribute to the overall hospital score, and in that no group has one measure with outsized dominance over the final score, since there are at least several measures with high coefficients in every group.

7 - Strengths and Weaknesses

7.1 - Strengths

- Because our HSMR is calculated with respect to a reference population of people for whom death is amenable, many of the problems with HSMRs as they are currently used by many hospitals are rectified.
    - Issues of disease severity raised by Van Gestel et al. do not apply to our model because patients with diseases that are severe enough to limit their lifespan regardless of medical intervention are not in our reference population.
- Our HSMR model uses mixed sampling when regressing across independent variables.
    - Since such a small percentage of patients in hospitals actually die, the probability of death is low regardless of other factors. Because we used mixed sampling to gain insight about the probability of a patient dying, we can more accurately determine the denominator of our HSMR.
- Our model uses latent variable modeling to calculate group scores.
    - Rather than taking a weighted sum of standardized measure values, latent variable modeling detects an underlying dimension of quality this process is preferred for analyzing composite measures, particularly in healthcare (Landrum et al., 2000).
- Our weights are derived from multiple authoritative sources.
    - Rather than choosing weights arbitrarily, we base our weight metric on four existing accredited hospital ranking models, modifying the values based on additional research maximize the accuracy of our weights.
- Our holistic model utilizes over 50 diverse measures of hospital quality.
    - While we cannot account for every variable that impacts hospital quality, our model does account for account for far more measures than most existing hospital ranking models.
- Our model modifies the weight metric based on user input.
    - Unlike all existing models, our model takes patient preference into account to redesign the weight metric to fit the priorities of the patient. This defeats the inherent problem of subjectivity in hospital ranking systems.
- Our model replaces unknown measure data with the national mean.
    - Imputing missing data with the national mean sets the standard deviation of that measure to zero. This processes the quality scores as though the missing measures were not a factor at all, which maintains a reasonable range and does not disrupt the consistency of direct hospital comparison (Olmsted et al., 2017)
- Our model avoids expert bias.
    - Whereas the US News hospital ranking model is based on expert opinion and hospital reputation, with no regard to patient preference, we find that patient experience is more relevant for the purpose of our model. In excluding expert opinion, we avoid the inevitable reputation bias that arises in such surveys.
- Our model has potential global applications.
    - Due to our limited access to data, we were unable to test our model on other regions. However, if one has access to any sort of hospital discharge data that includes the features we regress across, it is possible to apply our model.

7.2 - Weaknesses

- Our model is probabilistic. It is impossible to truly tell up to an arbitrary measure of accuracy whether or not a patient will die with respect to the calculation of our HSMR.
- Symptom severity is not directly correlated to outcome. Our means of customizing the outcome weight to the user is asking them to rate the severity of their symptoms. While the largest impact this can have on the outcome weight is 10%, the correlation between symptom severity and outcome is not airtight.
- Our weights are not 100% accurate. As is the problem with all hospital ranking models, there is no way for us to securely determine the most accurate weights for hospital assessment because of the underlying subjectivity.
- HealthSearch survey obtains a limited amount of information from the user. To maintain user-friendliness, we had to sacrifice the depth of the questions we could ask users.
- Over-sampling the New York State dataset creates overfitting of the minority variable in our HSMR regression. While overfitting was mitigated by class weighting, some overfitting was inevitable. Also, our regression uses a measure for amenable deaths with high specificity, but low sensitivity.

8 - Conclusion and Future Work

Our aim in the study was to create an algorithm and associated user-friendly interface that meaningfully quantifies the quality of hospitals, thereby allowing an individual to make an informed choice of the hospitals available to them. To accomplish this, we first fit a logistic regression to patient-level discharge data so as to assess the expected vs. observed mortality given a particular case-mix. However, our research ultimately showed that these HSMRs fail to satisfactorily model the complex causes of mortality in a hospital setting and that outcomes are only one aspect of a hospital's quality of care. Given this, we developed a holistic latent variable model to assess scores for seven groups with over fifty measures from the CMS. We then take a weighted average of these group scores with weights determined both by comparative research and user input. We assessed two test cases: the five hospitals closest to a randomly selected location and the best hospitals of all those within 60 miles of that location. We performed both these tests using three sets of weights to model user differences. Having selected Wheaton, IL as our location, we found that Advocate Good Samaritan was the best in all cases for the first test, and Rochelle Community was the best in all cases for the second test. To test the sensitivity of our model, we analyzed the regression coefficients, finding that no individual measures have predominant control of the group scores. We also explored the application of this model to an alternate use-case analyzing the regional quality of healthcare in the US.

The largest limitations to further model construction are a lack of comprehensive patient- and hospital-level data. Given access to a large dataset of hospital metrics for a country outside the United States, our model could easily be used to assess their quality. In addition, more hospital metrics or patient-level data could allow for a greater degree of confidence in the robustness of our fit regression. Finally, feedback from users of the HealthSearch program (i.e., post-visit surveys of whether the hospital they attended lived up to our score) could be dynamically incorporated into updated versions of our model.

9 - References

Amenable and preventable deaths statistics (no date) *Amenable and preventable deaths statistics - Statistics Explained*. Available at: http://ec.europa.eu/eurostat/statistics-explained/index.php/Amenable_and_preventable_deaths_statistics.

Austin, J. M., Jha, A. K., Romano, P. S., Singer, S. J., Vogus, T. J., Wachter, R. M. and Pronovost, P. J. (2015) "National Hospital Ratings Systems Share Few Common Scores And May Generate Confusion Instead Of Clarity," *Health Affairs*, 34(3), pp. 423–430. doi: 10.1377/hlthaff.2014.0201.

Data By Hospital - Dartmouth Atlas of Health Care (no date) *The Dartmouth Atlas of Health Care*. Available at: http://www.dartmouthatlas.org/data/hospital/.

Data.Medicare.gov (no date) *Data.Medicare.Gov*. Available at: https://data.medicare.gov/.

Doyle, J., Graves, J. and Gruber, J. (2017) "Evaluating Measures of Hospital Quality." doi: 10.3386/w23166.

Geweke, J., Gowrisankaran, G. and Town, R. (2001) "Bayesian Inference for Hospital Quality in a Selection Model." doi: 10.3386/w8497.

Gorman, A. (2016) *'The older you are, the worse the hospital is for you'*, *CNN*. Cable News Network. Available at: https://www.cnn.com/2016/08/15/health/elderly-hospital-patients/index.html.

Henderson CR. Best Linear Unbiased Estimation and Prediction under a Selection Model. Biometrics 1975;31:423-47.

HCUPnet (no date) *HCUPnet*. Available at: https://hcupnet.ahrq.gov/#setup.

Hosmer, D. W. and Lemeshow, S. (2007) *Confidence interval estimates of an index of quality performance based on logistic regression models*, *Statistics in Medicine*. Wiley-Blackwell. Available at: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.4780141909.

Hosmer, D. W. and Lemeshow, S. (1995) "Confidence interval estimates of an index of quality performance based on logistic regression models," *Statistics in Medicine*, 14(19), pp. 2161–2172. doi: 10.1002/sim.4780141909.

Jha, A. K., Li, Z., Orav, E. J. and Epstein, A. M. (2005) "Care in U.S. Hospitals — The Hospital Quality Alliance Program," *New England Journal of Medicine*, 353(3), pp. 265–274. doi: 10.1056/nejmsa051249.

Jha, A. K., Li, Z., Orav, E. J. and Epstein, A. M. (2005) "Care in U.S. Hospitals — The Hospital Quality Alliance Program," *New England Journal of Medicine*, 353(3), pp. 265–274. doi: 10.1056/nejmsa051249.

Landrum M, Bronskill S, Normand S-L. Analytic Methods for Constructing Cross-Sectional Profiles of Health Care Providers. Health Services and Outcomes Research Methodology 2000;1:23-47.

Luft, H. S. (1986) "Evaluating individual hospital quality through outcome statistics," *JAMA: The Journal of the American Medical Association*, 255(20), pp. 2780–2784. doi: 10.1001/jama.255.20.2780.

Luft, H. S. (1986) *Evaluating Individual Hospital Quality Through Outcome Statistics*, *JAMA*. American Medical Association. Available at: https://jamanetwork.com/journals/jama/article-abstract/404350.

Murrey G. Olmsted et al. (2017)”U.S. News & World Report 2017-18 Best Hospitals: Specialty Rankings.

National Healthcare Quality and Disparities Reports (no date) *NHQDR Web Site - Select State*. Available at: https://nhqrnet.ahrq.gov/inhqrdr/state/select.

“Pattern Recognition & Machine Learning” (1992). doi: 10.1016/c2009-0-22409-3.

Pedregosa, Fabian, Varoquaux, Gaël, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, Olivier, Blondel, Mathieu, Prettenhofer, Peter, Weiss, Ron, Jake, Passos, Cournapeau, David, Brucher, Matthieu, Perrot and Duchesnay (1970) *Scikit-learn: Machine Learning in Python*, *Journal of Machine Learning Research*. Available at: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

Rosenthal, G. E., Shah, A., Way, L. E. and Harper, D. L. (1998) *Variations in standardized hospital mortality rates for six common medical diagnoses: implications for profiling hospital quality.*, *Medical care.* U.S. National Library of Medicine. Available at: https://www.ncbi.nlm.nih.gov/pubmed/9674614.

Shojania, K. G. and Forster, A. J. (2008) “Hospital mortality: when failure is not a good measure of success,” *Canadian Medical Association Journal*, 179(2), pp. 153–157. doi: 10.1503/cmaj.080010.

Ajinkya More (2016) “Survey of resampling techniques for improving classification performance in unbalanced datasets”

Shwartz, M., Ren, J., Peköz, E. A., Wang, X., Cohen, A. B. and Restuccia, J. D. (2008) “Estimating a Composite Measure of Hospital Quality From the Hospital Compare Database,” *Medical Care*, 46(8), pp. 778–785. doi: 10.1097/mlr.0b013e31817893dc.

Wang, X. (2014) “Firth logistic regression for rare variant association tests,” *Frontiers in Genetics*, 5. doi: 10.3389/fgene.2014.00187.

Weiner, A. B., Matulewicz, R. S., Tosoian, J. J., Feinglass, J. M. and Schaeffer, E. M. (2018) “The effect of socioeconomic status, race, and insurance type on newly diagnosed metastatic prostate cancer in the United States (2004–2013),” *Urologic Oncology: Seminars and Original Investigations*, 36(3). doi: 10.1016/j.urolonc.2017.10.023.

Yvette R. B. M. Van Gestel, Lemmens, V. E. P. P., Lingsma, H. F., Ignace H. J. T. De Hingh, Rutten, H. J. T. and Coebergh, J. W. W. (2012) “The Hospital Standardized Mortality Ratio Fallacy,” *Medical Care*, 50(8), pp. 662–667. doi: 10.1097/mlr.0b013e31824ebd9f.

A review on Gaussian Process Latent Variable Models (2016) *CAAI Transactions on Intelligence Technology*. No longer published by Elsevier. Available at: https://www.sciencedirect.com/science/article/pii/S2468232216300828.

10 - Appendix

Table of Contents

Graphs and Displays **2**
Histogram of Quality Scores Across All US CMS-Participating Hospitals 2
Column Chart of Quality Scores Across All US CMS-Participating Hospitals 3
Full Size HealthSearch Screenshots 4
HSMR Patient-Level Included and Excluded Measures 6

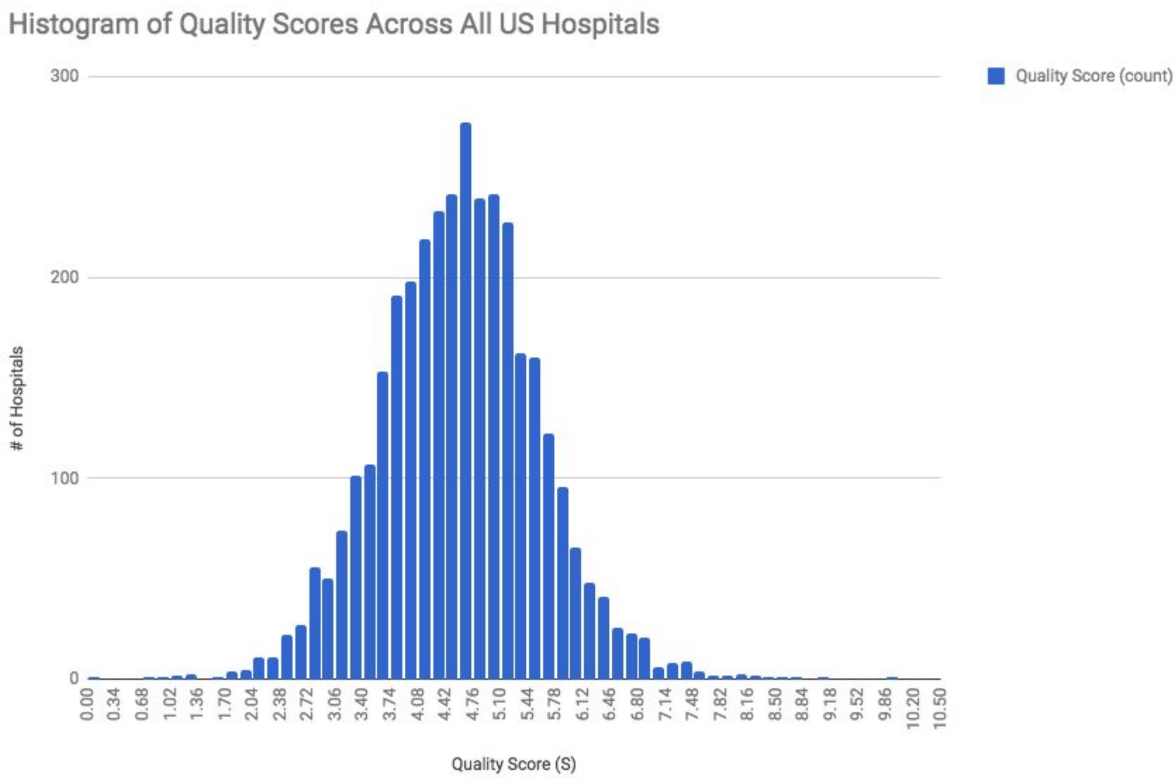
Tables **7**
Measure Loading Coefficients 7
Top 50 United States Hospitals with Measure Group Scores 11
US Ranked Statewide Average Quality and Measure Group Scores 14

Programs **16**
Latent 16
Latent Variable Model 16
Sobol Sensitivity Analysis 16

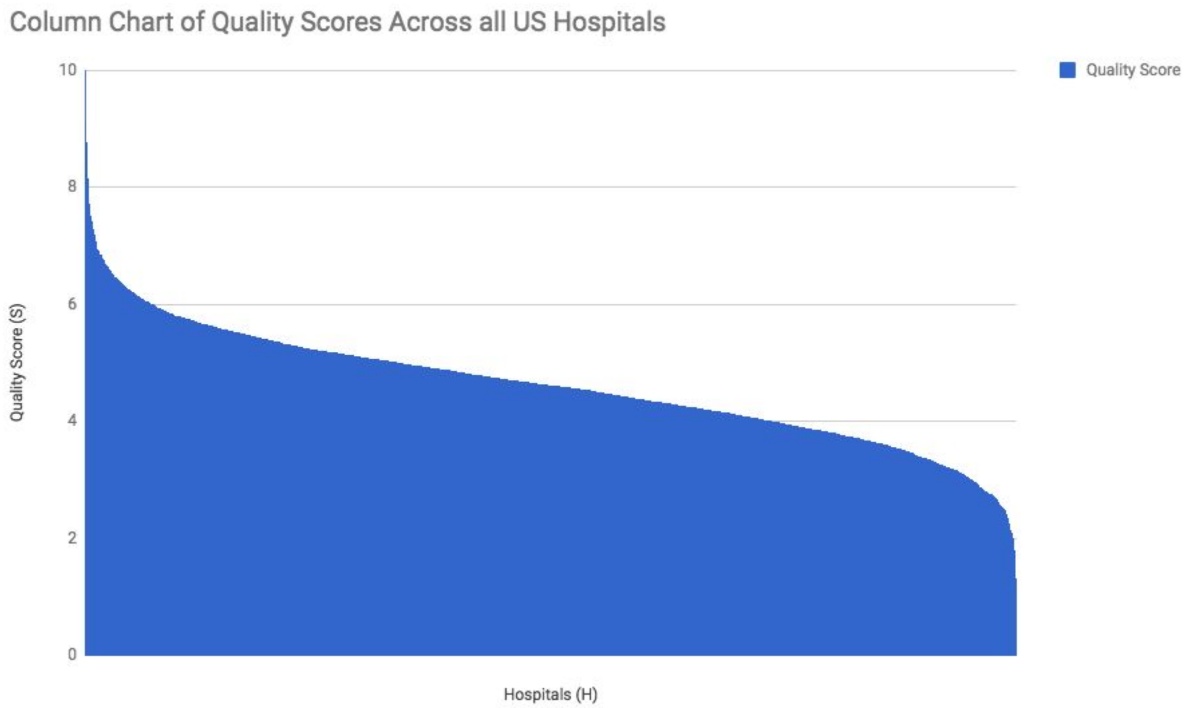
Derivations **17**
Derivation of Scikit-Learn's Logistic Regression Minimization Function 17

Graphs and Displays

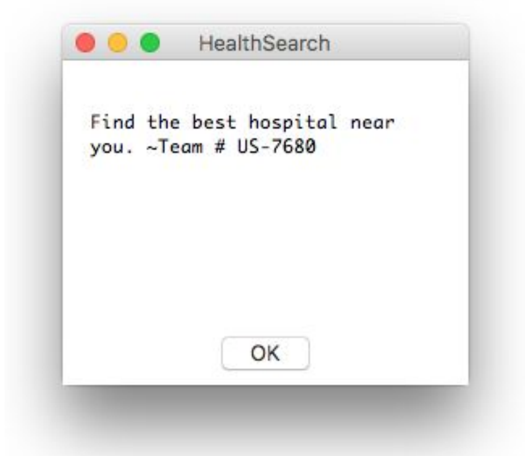
Histogram of Quality Scores Across All US CMS-Participating Hospitals

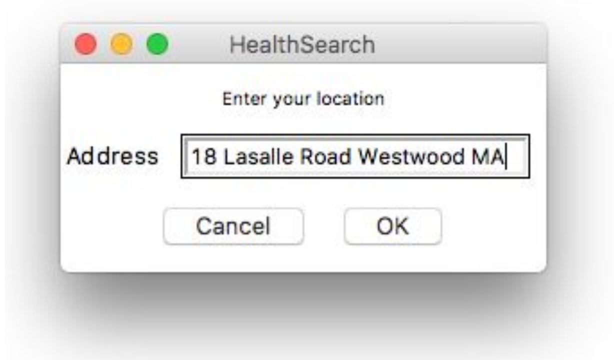


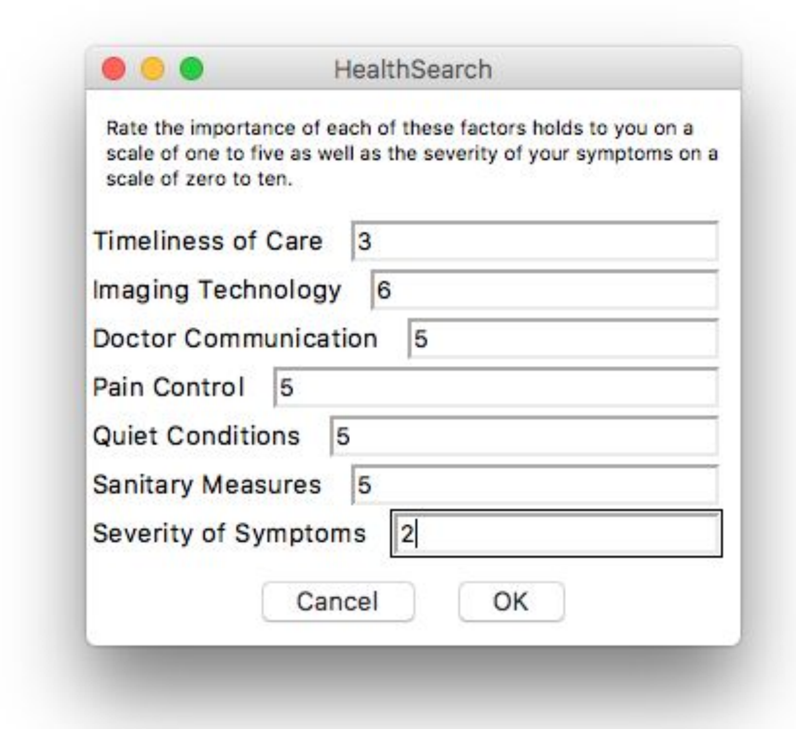
Column Chart of Quality Scores Across All US CMS-Participating Hospitals

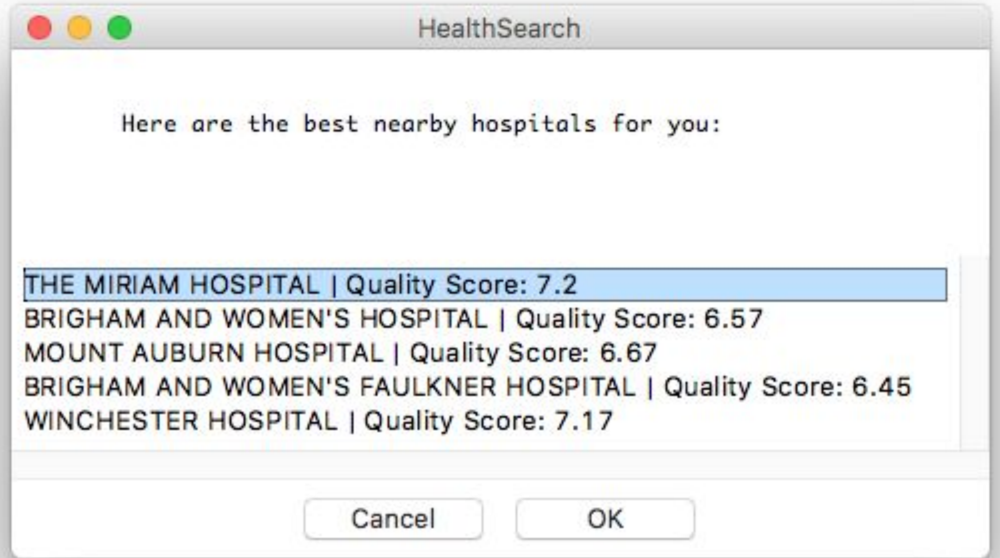


Full Size HealthSearch Screenshots








HealthSearch
Here are the best nearby hospitals for you:
THE MIRIAM HOSPITAL | Quality Score: 7.2
BRIGHAM AND WOMEN'S HOSPITAL | Quality Score: 6.57
MOUNT AUBURN HOSPITAL | Quality Score: 6.67
BRIGHAM AND WOMEN'S FAULKNER HOSPITAL | Quality Score: 6.45
WINCHESTER HOSPITAL | Quality Score: 7.17
Cancel
OK

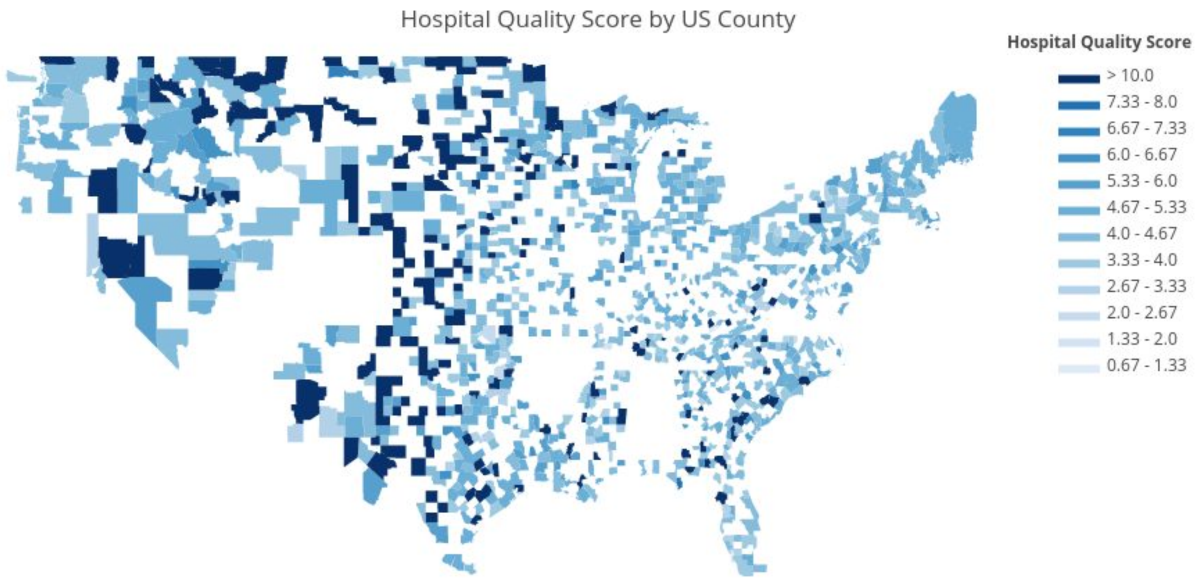
HSMR Patient-Level Included and Excluded Measures

37 preliminary measures:
Health Service Area, Hospital County, Operating Certificate Number, Facility ID, Facility Name, Age Group, Zip Code, Gender, Race, Ethnicity, Length of Stay, Type of Admission, Discharge Year, CCS Diagnosis Code, CCS Diagnosis Description, CCS Procedure Code, CCS Procedure Description, APR DRG Code, APR MDC Code, APR MDC Description, APR Severity of Illness Code, APR Severity of Illness Description, APR Risk of Mortality, APR Medical Surgical Description, Payment Typology 1, Payment Typology 2, Payment Typology 3, Attending Provider License Number, Operating Provider License Number, Other Provider Licence Number, Birth Weight, Abortion Edit Indicator, Emergency Department Indicator, Total Charges, Total Costs.

7 final measures:
Age Group
Gender
Race
Length of Stay
Type of Admission
Payment Typology 1
APR Medical Surgical Description

Choropleth Map of Hospital Quality Score by US County

NOTE: the empty counties are due to discrepancies between county names in the CMS's data and the proper IDs of those counties



Tables

Measure Loading Coefficients

Measure ID	Measure Description	Loading Coefficient
	Mortality Measure Group	
MORT_30_AMI	Acute Myocardial Infarction (AMI) 30-Day Mortality Rate	0.4559688173
MORT_30_CABG	Coronary Artery Bypass Graft (CABG) 30-Day Mortality Rate	0.2692612166
MORT_30_COPD	Chronic Obstructive Pulmonary Disease (COPD) 30-Day Mortality Rate	0.5598636774
MORT_30_HF	Heart Failure (HF) 30-Day Mortality Rate	0.6199874176
MORT_30_PN	Pneumonia (PN) 30-Day Mortality Rate	0.6116240919
MORT_30_STK	Acute Ischemic Stroke (STK) 30-Day Mortality Rate	0.4808452953
PSI_4_SURG_COMP	Death Among Surgical Patients with Serious Treatable Complications	0.2332128044
	Safety Measure Group	
HAI_1_SIR	Central-Line Associated Bloodstream Infection (CLABSI)	-0.532278529
HAI_2_SIR	Catheter-Associated Urinary Tract Infection (CAUTI)	-0.3171571143
HAI_3_SIR	Surgical Site Infection from colon surgery (SSI-colon)	-0.1301562865
HAI_4_SIR	Surgical Site Infection from abdominal hysterectomy (SSI-abdominal hysterectomy)	-0.100550491
HAI_5_SIR	MRSA Bacteremia	-0.2903372275
HAI_6_SIR	Clostridium Difficile (C. difficile)	-0.0811050103
COMP_HIP_KNEE	Hospital-Level Risk-Standardized Complication Rate (RSCR) Following Elective Primary Total Hip Arthroplasty (THA) and Total Knee Arthroplasty (TKA)	-0.06344172506
PSI_90_SAFETY	Safety Complication/Patient Safety for Selected Indicators (PSI)	-0.1125190002

	Readmission Measure Group	
READM_30_CABG	Coronary Artery Bypass Graft (CABG) 30-Day Readmission Rate	0.1756228287
READM_30_COPD	Chronic Obstructive Pulmonary Disease (COPD) 30-Day Readmission Rate	0.5333992563
READM_30_HIP_KNEE	Hospital-Level 30-Day All-Cause Risk-Standardized Readmission Rate (RSRR) Following Elective Total Hip Arthroplasty (THA)/Total Knee Arthroplasty (TKA)	0.2830105668
READM_30_PN	Pneumonia (PN) 30-Day Readmission Rate	0.7025639523
READM_30_STK	Stroke (STK) 30-Day Readmission Rate	0.5114217353
READM_30_HOSP_WIDE	Hospital-Wide All-Cause Unplanned Readmission	0.8473689034
EDAC_30_AMI	Excess Days in Acute Care (EDAC) after hospitalization for Acute Myocardial Infarction (AMI)	0.432724863
EDAC_30_HF	Excess Days in Acute Care (EDAC) after hospitalization for Heart Failure (HF)	0.5782404234
OP_32	Facility 7-Day Risk Standardized Hospital Visit Rate after Outpatient Colonoscopy	0.03753119584
	Experience Measure Group	
H_CLEAN_LINEAR_SCORE	Cleanliness of Hospital Environment (Q8)	0.7019662916
H_COMP_1_LINEAR_SCORE	Nurse Communication (Q1, Q2, Q3)	0.9229018521
H_COMP_2_LINEAR_SCORE	Doctor Communication (Q5, Q6, Q7)	0.7902457714
H_COMP_3_LINEAR_SCORE	Responsiveness of Hospital Staff (Q4, Q11)	0.8602306169
H_COMP_4_LINEAR_SCORE	Pain Management (Q13, Q14)	0.8608832404
H_COMP_5_LINEAR_SCORE	Communication About Medicines (Q16, Q17)	0.8345479566

H_COMP_6_LINEAR_SCORE	Discharge Information (Q19, Q20)	0.6824900621
H_HSP_RATING_LINEAR_SCORE	Overall Rating of Hospital (Q21)	0.9123077224
H_QUIET_LINEAR_SCORE	Quietness of Hospital Environment (Q9)	0.6202164868
H_RECMND_LINEAR_SCORE	Willingness to Recommend Hospital (Q22)	0.8444709184
H_COMP_7_LINEAR_SCORE	HCAHPS 3 Item Care Transition Measure (CTM-3)	0.8799908892
	Effectiveness Measure Group	
IMM-2	Influenza Immunization	-0.1322432369
IMM_3_OP_27_FAC_ADHPCT	Healthcare Personnel Influenza Vaccination	-0.1776777911
OP_4	Aspirin at Arrival	-0.1162971076
OP_22	Emergency Department (ED)-Patient Left Without Being Seen	-0.1415832491
OP_23	Emergency Department (ED)-Head Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) Scan Results for Acute Ischemic Stroke or Hemorrhagic Stroke Who Received Head CT or MRI Scan Interpretation Within 45 Minutes of Arrival	0.002028776703
PC_01	Elective Delivery	-0.1371216267
VTE_6	Hospital Acquired Potentially-Preventable Venous Thromboembolism	-0.6197955686
OP_29	Endoscopy/Poly Surveillance-Appropriate Follow-up Interval for Normal Colonoscopy in Average Risk Patients	-0.550189449
OP_30	Endoscopy/Poly Surveillance: Colonoscopy Interval for Patients with a History of Adenomatous Polyps – Avoidance of Inappropriate Use	-0.004585708482
	Timeliness Measure Group	
ED_1b	Median Time from Emergency Department (ED) Arrival to ED Departure for Admitted ED Patients	-0.997067134

ED_2b	Admit Decision Time to Emergency Department (ED) Departure Time for Admitted Patients	-0.874543608
OP_3b	Median Time to Transfer to Another Facility for Acute Coronary Intervention	-0.0516621176
OP_5	Median Time to Electrocardiography (ECG)	-0.1294382782
OP_18b	Median Time from Emergency Department (ED) Arrival to ED Departure for Discharged ED Patients	-0.702087092
OP_20	Door to Diagnostic Evaluation by a Qualified Medical Professional	-0.4885595005
OP_21	Emergency Department (ED)-Median Time to Pain Management for Long Bone Fracture	-0.373008447
	Imaging Measure Group	
OP_8	Magnetic Resonance Imaging (MRI) Lumbar Spine for Low Back Pain	-0.06472648744
OP_10	Abdomen Computed Tomography (CT) Use of Contrast Material	-0.6555213966
OP_11	Thorax Computed Tomography (CT) Use of Contrast Material	-0.6676661047
OP_13	Cardiac Imaging for Preoperative Risk Assessment for Non-Cardiac Low-Risk Surgery	0.04250642878
OP_14	Simultaneous Use of Brain Computed Tomography (CT) and Sinus CT	0.0513168415

Top 50 United States Hospitals with Measure Group Scores

Hospital Name	Quality Score	Mortality Score	Safety Score	Readmission Score	Experience Score	Effectiveness Score	Timeliness Score	Imaging Score
PARK PLACE SURGICAL HOSPITAL	10	No Score	2.28	6.63	9.39	6.28	No Score	No Score
KERN MEDICAL CENTER	9.14	No Score	2.09	6.49	4.53	5.9	9.2	0.54
PREMIER SURGICAL INSTITUTE	8.77	No Score	2.12	7.7	8.91	3.24	No Score	3.21
ST FRANCIS HOSPITAL, ROSLYN	8.58	7.34	2.73	6.04	7.27	3.06	6.42	0.46
CLOVIS COMMUNITY MEDICAL CENTER	8.47	2.85	5.59	5.96	6.68	1.26	10	0.63
OKLAHOMA CENTER FOR ORTHOPAEDIC & MULTI-SP	8.32	No Score	2.22	6.61	8.78	1.45	No Score	9.53
HACKENSACK UNIVERSITY MEDICAL CENTER	8.32	7.67	2.54	5.3	7.24	6.89	4.41	0.53
NEW ENGLAND BAPTIST HOSPITAL	8.15	5.77	1.93	9.06	8.34	3.4	No Score	0.62
MAYO CLINIC HOSPITAL	8.13	8.36	1.27	8.45	8.55	3.09	2.07	0.6
SIOUX FALLS SPECIALTY HOSPITAL LLP	8.08	No Score	2.17	7.3	8.48	4.06	No Score	1.05
GHS PATEWOOD MEMORIAL HOSPITAL	7.95	No Score	1.71	7.29	8.83	2.9	No Score	1.52
BEACHAM MEMORIAL HOSPITAL	7.88	6.43	2.35	6.57	9.55	2	No Score	1.12
OHIOHEALTH SHELBY HOSPITAL	7.72	6.22	2.45	7.58	7.99	No Score	No Score	No Score
STANISLAUS SURGICAL HOSPITAL	7.68	No Score	2.22	7	9.09	0.64	No Score	4.63
NEW YORK-PRESBYTERIAN HOSPITAL	7.59	9.03	2.31	5.97	6.29	0.74	5.95	1.62
CENTRAL LOUISIANA SURGICAL HOSPITAL	7.59	No Score	2.11	6.71	9.24	1.5	No Score	2.05

MCCURTAIN MEMORIAL HOSPITAL	7.54	5.59	2.35	5.57	No Score	6.11	2.87	6.64
AUXILIO MUTUO HOSPITAL	7.51	2.84	2.34	6.42	No Score	3.3	6.11	No Score
FRANCISCAN HEALTH CARMEL	7.48	No Score	2.25	7.07	9.26	0.55	No Score	No Score
CARE REGIONAL MEDICAL CENTER	7.47	5.37	2.35	8.06	No Score	4.4	3.15	1.26
TYRONE HOSPITAL	7.46	5.25	2.43	6.58	8.71	7.88	1.19	0.62
SURGICAL INSTITUTE OF READING	7.44	No Score	2.27	6.4	9.13	1.68	No Score	No Score
PALO VERDE HOSPITAL	7.41	4.54	2.36	6.72	No Score	5.88	3.16	4.02
SACRED HEART HOSPITAL ON THE GULF	7.36	6.12	2.35	7.23	8.07	6.65	1.39	1.07
HIMA SAN PABLO BAYAMON	7.36	5.39	2.94	4.97	No Score	2.51	5.73	No Score
PRESBYTERIAN INTERCOMMUNITY HOSPITAL	7.34	7.62	2.69	5.91	7.34	5.66	2.59	1.56
COMMUNITY REGIONAL MEDICAL CENTER	7.33	5.12	3.89	4.23	5.86	0.79	9.34	0.75
FRANKLIN WOODS COMMUNITY HOSPITAL	7.27	6.56	2.34	7.29	7.65	6.29	1.61	1.12
SAINT BARNABAS MEDICAL CENTER	7.25	7.7	1.29	5.68	6.88	6.22	3.04	1.53
PRESBYTERIAN COMMUNITY HOSP	7.24	4.22	1.53	5.78	No Score	No Score	5.55	No Score
CHRISTUS SPOHN HOSPITAL ALICE	7.23	5.58	2.15	7.23	6.67	4.95	3.33	4.59
OHIO VALLEY MEDICAL CENTER, LLC	7.19	No Score	2.27	6.54	8.71	2.17	No Score	1.71
SUMMIT HEALTHCARE REGIONAL MEDICAL CENTER	7.18	5.7	2.7	8.19	7.09	5.07	2.33	2.37
MASSACHUSETTS GENERAL HOSPITAL	7.17	7.21	2.21	5.57	7.43	1.07	5.27	0.4
SURGICAL HOSPITAL AT SOUTHWOODS	7.16	No Score	2.28	5.91	9.57	0.64	No Score	2.32

MERCY WILLARD HOSPITAL	7.13	5.59	2.35	6.88	9.04	4.81	1.38	0.64
CHESTER COUNTY HOSPITAL	7.13	7.87	2.22	8.1	7.72	1.18	2.7	0.48
CALDWELL MEMORIAL HOSPITAL, INC	7.08	6.69	2.35	6.23	8.34	2.37	No Score	No Score
SUGAR LAND SURGICAL HOSPITAL LLP	7.02	No Score	2.36	6.46	9.59	0.61	2.09	3.24
IU HEALTH WEST HOSPITAL	7	7.05	5.32	8.07	7.16	1.45	2.99	0.5
REDLANDS COMMUNITY HOSPITAL	6.99	4.69	0.88	7.54	6.74	4.22	4.22	2.05
MAYO CLINIC HOSPITAL ROCHESTER	6.95	7.63	1.65	6.96	7.97	3.33	1.99	0.51
DOCTOR'S MEMORIAL HOSPITAL INC	6.94	5.25	2.18	6.45	6.96	7.99	2.44	0.8
NEWARK BETH ISRAEL MEDICAL CENTER	6.94	5.83	1.32	3.52	6.01	8.12	5.13	1.49
SAN LUKE'S MEMORIAL HOSPITAL INC	6.93	4.54	2.16	4.9	No Score	2.28	5.98	No Score
RONALD REAGAN U C L A MEDICAL CENTER	6.92	7.56	2.23	4.7	7.67	1.44	4.43	1.92
LOMA LINDA UNIVERSITY MEDICAL CENTER	6.91	6.23	4.18	3.95	6.39	1.58	6.77	2.37
OAKLEAF SURGICAL HOSPITAL	6.91	No Score	2.29	6.79	8.9	0.56	No Score	No Score
EMORY UNIVERSITY HOSPITAL	6.9	6.68	3.01	6.41	6.91	2.37	4.26	0.97

US Ranked Statewide Average Quality and Measure Group Scores

State Name	Quality Score	Mortality Score	Safety Score	Readmission Score	Experience Score	Effectiveness Score	Timeliness Score	Imaging Score	Number of Hospitals
Delaware	5.83	5.85	2.59	6.58	6.85	2.01	3.2	0.45	5
Rhode Island	5.06	5.83	2.77	6.47	6.67	1.64	2.35	0.64	11
Louisiana	5.05	5.19	2.67	6.34	6.96	1.92	1.91	2.28	76
Maine	4.95	4.63	2.27	6.84	7.13	1.06	2.33	0.72	30
Massachusetts	4.91	6.17	2	5.92	6.57	1.25	2.61	0.66	56
Hawaii	4.9	4.92	1.96	7.2	6.65	1.59	2.21	1.01	12
Colorado	4.9	4.92	2.21	7.39	6.92	1.52	1.66	1.15	47
Connecticut	4.89	5.56	2.52	6.17	6.38	1.62	2.76	0.61	27
Alaska	4.88	4.64	2.46	7.03	6.45	1.56	2.6	0.77	10
South Dakota	4.87	5	2.52	7.51	7.28	1.42	0.94	0.93	18
New Hampshire	4.87	5.03	2.45	6.63	7.06	1.21	2.19	0.59	24
Indiana	4.87	4.89	2.33	6.91	7.06	1.53	1.77	1.07	109
Texas	4.86	5.12	2.29	6.64	6.64	1.86	2.05	1.58	262
Ohio	4.83	5.45	2.1	6.55	6.84	1.52	1.91	0.85	132
Michigan	4.82	5.21	2.32	6.56	6.8	1.47	2.08	1.24	117
Idaho	4.8	4.62	1.82	7.35	6.9	1.66	1.65	0.74	22
Nebraska	4.77	4.78	2.33	6.89	7.13	1.59	1.76	0.96	43
Wisconsin	4.76	4.83	2.3	7.09	7.39	1.27	1.41	0.71	101
New Jersey	4.73	5.7	2.44	5.52	5.85	2.74	2.87	0.83	62
California	4.71	5.31	2.52	6.44	5.84	1.79	2.85	1.07	273
New Mexico	4.68	4.75	2.49	6.82	6.05	2.19	2.3	1.17	29
Montana	4.68	4.57	2.13	7.13	6.88	1.41	1.65	1.39	22
Oregon	4.67	4.42	2.28	7	6.85	1.8	1.87	0.74	49
Vermont	4.66	4.46	2.33	6.75	6.96	1.15	2.18	0.52	12
Wyoming	4.62	4.39	2.18	6.89	6.84	2.39	1.53	1.42	14
Georgia	4.62	4.89	2.55	6.43	6.28	1.82	2.34	1.31	99
South Carolina	4.62	4.73	2.41	6.54	6.67	1.63	2.12	0.86	52

Iowa	4.61	4.41	2.25	6.9	7.15	1.52	1.58	1.23	78
Pennsylvania	4.6	5.27	2.23	6.34	6.52	1.65	2.04	1.14	143
Kansas	4.6	4.7	2.2	6.76	7.1	1.68	1.24	1.93	59
Maryland	4.6	5.24	2.74	6.27	5.66	1.54	3.22	0.54	44
Minnesota	4.59	5.23	2.28	6.7	7.15	1.36	1.42	0.62	73
North Carolina	4.59	4.7	2.38	6.67	6.59	1.41	2.24	0.76	91
Washington DC	4.58	5.49	3.21	5.51	4.74	3	3.73	1.3	7
Arizona	4.58	5.26	2.21	6.89	6.1	1.49	2.22	1.06	55
Oklahoma	4.56	4.81	2.38	6.59	6.76	1.89	1.51	1.9	79
Washington	4.54	4.39	2.1	7.14	6.41	1.63	2.16	0.72	61
Utah	4.53	4.71	2.35	7.33	6.82	1.27	1.52	0.69	30
Illinois	4.53	5.36	2.19	6.22	6.55	1.51	2	1.18	149
West Virginia	4.51	5.03	2.13	6.13	6.42	2.19	2	1.35	36
Kentucky	4.47	4.82	2.26	5.91	6.88	1.69	1.89	1.49	79
Alabama	4.46	4.64	2.33	6.3	6.6	1.97	1.86	1.58	78
Virginia	4.4	5.03	2.18	6.32	6.52	1.36	2.1	0.82	76
Missouri	4.34	4.92	2.34	6.29	6.63	1.44	1.84	0.93	83
New York	4.26	5.17	2.68	5.33	5.58	1.85	3.18	0.83	147
Tennessee	4.25	4.66	2.33	6.27	6.46	1.92	1.82	1.03	91
North Dakota	4.24	4.62	2.35	6.97	6.58	1.54	1.51	0.8	17
Mississippi	4.19	4.57	2.58	5.91	6.68	2.11	1.52	1.72	62
Nevada	4.17	4.45	2.43	5.87	5.66	2.78	2.52	0.94	25
Arkansas	4.16	4.1	2.48	6.03	6.68	2.19	1.63	1.46	54
Florida	3.98	5.19	2.48	5.64	5.85	1.74	2.24	0.98	164

Programs

Latent Variable Model and HospitalSearch GUI

```
# IMMC 2018 Latent Variable Model Calculation
# Team # 2018032

import googlemaps
import json
import numpy as np
import itertools
import operator
import math
import pandas as pd
from factor_analyzer import FactorAnalyzer
from scipy import stats
from sodapy import Socrata
from openpyxl import load_workbook
from sklearn import linear_model, preprocessing, decomposition

train_features = ['MORT_30_AMI', 'MORT_30_CABG', 'MORT_30_COPD', 'MORT_30_HF', 'MORT_30_PN',
'MORT_30_STK', 'PSI_4_SURG_COMP', 'HAI-1', 'HAI-2', 'HAI-3', 'HAI-4', 'HAI-5', 'HAI-6',
'COMP_HIP_KNEE', 'PSI_90_SAFETY', 'READM_30_CABG', 'READM_30_COPD', 'READM_30_HIP_KNEE',
'READM_30_PN', 'READM_30_STK', 'READM_30_HOSP_WIDE', 'EDAC_30_AMI', 'EDAC_30_HF', 'OP_32',
'H_CLEAN_LINEAR_SCORE', 'H_COMP_1_LINEAR_SCORE', 'H_COMP_2_LINEAR_SCORE',
'H_COMP_3_LINEAR_SCORE', 'H_COMP_4_LINEAR_SCORE', 'H_COMP_5_LINEAR_SCORE',
'H_COMP_6_LINEAR_SCORE', 'H_HSP_RATING_LINEAR_SCORE', 'H_QUIET_LINEAR_SCORE',
'H_RECMND_LINEAR_SCORE', 'H_COMP_7_LINEAR_SCORE', 'IMM-2', 'IMM_3_OP_27_FAC_ADHPCT', 'OP_4',
'OP_22', 'OP_23', 'PC_01', 'VTE_6', 'OP_29', 'OP_30', 'OP_33', 'ED_1b', 'ED_2b', 'OP_3b',
'OP_5', 'OP_18b', 'OP_20', 'OP_21', 'OP_8', 'OP_10', 'OP_11', 'OP_13', 'OP_14']
direction_features = {'MORT_30_AMI': -1.0, 'MORT_30_CABG': -1.0, 'MORT_30_COPD': -1.0,
'MORT_30_HF': -1.0, 'MORT_30_PN': -1.0, 'MORT_30_STK': -1.0, 'PSI_4_SURG_COMP': -1.0,
'HAI_1_SIR': -1.0, 'HAI_2_SIR': -1.0, 'HAI_3_SIR': -1.0, 'HAI_4_SIR': -1.0, 'HAI_5_SIR': -1.0,
'HAI_6_SIR': -1.0, 'COMP_HIP_KNEE': -1.0, 'PSI_90_SAFETY': -1.0, 'READM_30_CABG': -1.0,
'READM_30_COPD': -1.0, 'READM_30_HIP_KNEE': -1.0, 'READM_30_PN': -1.0, 'READM_30_STK': -1.0,
'READM_30_HOSP_WIDE': -1.0, 'EDAC_30_AMI': -1.0, 'EDAC_30_HF': -1.0, 'OP_32': -1.0,
'H_CLEAN_LINEAR_SCORE': 1.0, 'H_COMP_1_LINEAR_SCORE': 1.0, 'H_COMP_2_LINEAR_SCORE': 1.0,
'H_COMP_3_LINEAR_SCORE': 1.0, 'H_COMP_4_LINEAR_SCORE': 1.0, 'H_COMP_5_LINEAR_SCORE': 1.0,
'H_COMP_6_LINEAR_SCORE': 1.0, 'H_HSP_RATING_LINEAR_SCORE': 1.0, 'H_QUIET_LINEAR_SCORE': 1.0,
'H_RECMND_LINEAR_SCORE': 1.0, 'H_COMP_7_LINEAR_SCORE': 1.0, 'IMM-2': 1.0,
'IMM_3_OP_27_FAC_ADHPCT': 1.0, 'OP_4': 1.0, 'OP_22': -1.0, 'OP_23': 1.0, 'PC_01': -1.0,
'VTE_6': -1.0, 'OP_29': 1.0, 'OP_30': 1.0, 'OP_33': 1.0, 'ED_1b': -1.0, 'ED_2b': -1.0,
'OP_3b': -1.0, 'OP_5': -1.0, 'OP_18b': -1.0, 'OP_20': -1.0, 'OP_21': -1.0, 'OP_8': -1.0,
'OP_10': -1.0, 'OP_11': -1.0, 'OP_13': -1.0, 'OP_14': -1.0}

measure_dbs = {"MORTPSI": "ukfj-tt6v", "HAI": "ppaw-hhm5", "READMEDAC": "r32h-32z5", "HCAHPS":
"rmgi-5fhi", "TIMEEFFECT": "3z8n-wcgr", "IMAGE": "72af-b2t9"}

measure_groups = {"mortality": ["MORT_30_AMI", "MORT_30_CABG", "MORT_30_COPD", "MORT_30_HF",
"MORT_30_PN", "MORT_30_STK", "PSI_4_SURG_COMP"],
```

```
                              "safety": ["HAI_1_SIR", "HAI_2_SIR", "HAI_3_SIR", "HAI_4_SIR",
"HAI_5_SIR", "HAI_6_SIR", "COMP_HIP_KNEE", "PSI_90_SAFETY"],
                              "readmission": ["READM_30_CABG", "READM_30_COPD",
"READM_30_HIP_KNEE", "READM_30_PN", "READM_30_STK", "READM_30_HOSP_WIDE", "EDAC_30_AMI",
"EDAC_30_HF", "OP_32"],
                              "experience": ["H_CLEAN_LINEAR_SCORE", "H_COMP_1_LINEAR_SCORE",
"H_COMP_2_LINEAR_SCORE", "H_COMP_3_LINEAR_SCORE", "H_COMP_4_LINEAR_SCORE",
"H_COMP_5_LINEAR_SCORE", "H_COMP_6_LINEAR_SCORE", "H_HSP_RATING_LINEAR_SCORE",
"H_QUIET_LINEAR_SCORE", "H_RECMND_LINEAR_SCORE", "H_COMP_7_LINEAR_SCORE"],
                              "effectiveness": ["IMM-2", "IMM_3_OP_27_FAC_ADHPCT", "OP_4",
"OP_22", "OP_23", "PC_01", "VTE_6", "OP_29", "OP_30", "OP_33"],
                              "timeliness": ["ED_1b", "ED_2b", "OP_3b", "OP_5", "OP_18b",
"OP_20", "OP_21"],
                              "imaging": ["OP_8", "OP_10", "OP_11", "OP_13", "OP_14"]
                              }

measure_ids = {"MORTPSI": ["MORT_30_AMI", "MORT_30_CABG", "MORT_30_COPD", "MORT_30_HF",
"MORT_30_PN", "MORT_30_STK", "PSI_4_SURG_COMP", "COMP_HIP_KNEE", "PSI_90_SAFETY"],
                              "HAI": ["HAI_1_SIR", "HAI_2_SIR", "HAI_3_SIR", "HAI_4_SIR",
"HAI_5_SIR", "HAI_6_SIR"],
                              "READMEDAC": ["READM_30_CABG", "READM_30_COPD",
"READM_30_HIP_KNEE", "READM_30_PN", "READM_30_STK", "READM_30_HOSP_WIDE", "EDAC_30_AMI",
"EDAC_30_HF", "OP_32"],
                              "HCAHPS": ["H_CLEAN_LINEAR_SCORE", "H_COMP_1_LINEAR_SCORE",
"H_COMP_2_LINEAR_SCORE", "H_COMP_3_LINEAR_SCORE", "H_COMP_4_LINEAR_SCORE",
"H_COMP_5_LINEAR_SCORE", "H_COMP_6_LINEAR_SCORE", "H_HSP_RATING_LINEAR_SCORE",
"H_QUIET_LINEAR_SCORE", "H_RECMND_LINEAR_SCORE", "H_COMP_7_LINEAR_SCORE"],
                              "TIMEEFFECT": ["IMM-2", "IMM_3_OP_27_FAC_ADHPCT", "OP_4",
"OP_22", "OP_23", "PC_01", "VTE_6", "OP_29", "OP_30", "OP_33", "ED_1b", "ED_2b", "OP_3b",
"OP_5", "OP_18b", "OP_20", "OP_21"],
                              "IMAGE": ["OP_8", "OP_10", "OP_11", "OP_13", "OP_14"]
                              }

weights = {"mortality": 15,
                     "safety": 5,
                     "readmission": 15,
                     "experience": 25,
                     "effectiveness": 17.5,
                     "timeliness": 17.5,
                     "imaging": 5
                     }

def zify_scipy(d):
    keys, vals = zip(*d.items())
    return dict(zip(keys, stats.zscore(vals, ddof=1)))

def find_db(measure_id):
        global measure_ids
        for db in measure_ids:
                if measure_id in measure_ids[db]:
```

```
				return db
		return None

def get_measure(hospital, measure_id):
		global measure_ids
		global measure_dbs
		db = find_db(measure_id)
		if db == "HCAHPS":
			id_col = "hcahps_measure_id"
			data_col = "hcahps_linear_mean_value"
		else:
			id_col = "measure_id"
			data_col = "score"
		row = medicare_client.get(measure_dbs[db], where=f"hospital_name = \"{hospital}\" AND
{id_col} = \"{measure_id}\"", limit=1)
		if row:
			return row[0][data_col]
		else:
			return None

def get_all_measures(hospital):
		global measure_ids
		global measure_dbs
		vals = {}
		for db in measure_ids:
			if db == "HCAHPS":
				id_col = "hcahps_measure_id"
				data_col = "hcahps_linear_mean_value"
			else:
				id_col = "measure_id"
				data_col = "score"
			rows = medicare_client.get(measure_dbs[db], where=f"hospital_name =
\"{hospital}\"", limit=200)
			for i in measure_ids[db]:
				for row in rows:
					if row[id_col] == i:
						vals[i] = row[data_col]
		return vals

def get_all_denoms(hospital):
		global measure_ids
		global measure_dbs
		vals = {}
		for db in measure_ids:
			if db == "HCAHPS":
				id_col = "hcahps_measure_id"
				data_col = "number_of_completed_surveys"
			elif db == "MORTPSI":
				id_col = "measure_id"
				data_col = "denominator"
```

```
            elif db == "HAI":
                continue
            elif db == "READMEDAC":
                id_col = "measure_id"
                data_col = "denominator"
            elif db == "TIMEEFFECT":
                id_col = "measure_id"
                data_col = "sample"
            elif db == "IMAGE":
                continue
            rows = medicare_client.get(measure_dbs[db], where=f"hospital_name =
\"{hospital}\"", limit=200)
            for i in measure_ids[db]:
                for row in rows:
                    if row[id_col] == i:
                        try:
                            vals[i] = float(row[data_col])
                        except ValueError:
                            vals[i] = None
        return vals

def get_denom_data(limit):
        denom = {}
        hospitals = [hospital["hospital_name"] for hospital in medicare_client.get("rbry-mqwu",
where="hospital_overall_rating != \"Not Available\"", limit=limit)]
        count = 1
        for hospital in hospitals:
            denom[hospital] = get_all_denoms(hospital)
            print(f"{hospital} METRICS COLLATED | {count} / {len(hospitals)}")
            count += 1
        return denom

def hospital_metrics(hospital):
        global measure_ids
        global measure_groups
        metrics = {}
        vals = get_all_measures(hospital)
        for group in measure_groups:
            metrics[group] = {}
            for measure in measure_groups[group]:
                if measure in vals.keys():
                    metrics[group][measure] = vals[measure]
                else:
                    metrics[group][measure] = None
        return metrics

def get_data(limit):
        sample = {}
        hospitals = [hospital["hospital_name"] for hospital in medicare_client.get("rbry-mqwu",
where="hospital_overall_rating != \"Not Available\"", limit=limit)]
```

```
        count = 1
        for hospital in hospitals:
                sample[hospital] = hospital_metrics(hospital)
                print(f"{hospital} METRICS COLLATED | {count} / {len(hospitals)}")
                count += 1
        return sample

def save_data(name, data):
        with open(name, 'w') as fp:
                json.dump(data, fp)
        return

def convert_data(hospitals):
        global measure_groups
        temp = {}
        converted = {}
        for group in measure_groups:
                for measure in measure_groups[group]:
                        temp[measure] = {}
                        for hospital in hospitals:
                                if hospitals[hospital][group][measure] != None:
                                        temp[measure][hospital] =
hospitals[hospital][group][measure]
        for measure in temp:
                for hospital in temp[measure]:
                        if temp[measure][hospital] == None or temp[measure][hospital] == "Not
Available":
                                temp[measure][hospital] = None
                        else:
                                temp[measure][hospital] = float(temp[measure][hospital])
                if temp[measure]:
                        pass
                        #temp[measure] = zify_scipy(temp[measure])
                else:
                        temp[measure] = {}
        for hospital in hospitals:
                converted[hospital] = {}
                for measure in temp:
                        if hospital in temp[measure].keys():
                                converted[hospital][measure] = temp[measure][hospital]
                        else:
                                converted[hospital][measure] = None
        return converted

denom_info = {}

def sample_variation_weight(measure, hospital):
        global hospital_denoms
        global denom_info
        global measure_ids
```

```
        if measure in measure_ids["HAI"] or measure in measure_ids["IMAGE"]:
                return 1
        if measure in denom_info.keys():
                nkhd = hospital_denoms[hospital][measure]
                if nkhd == None:
                        return 1
                return (nkhd / denom_info[measure]["sum"]) * denom_info[measure]["count"]

        else:
                Snkhd = []
                for h in hospital_denoms:
                        if measure in hospital_denoms[h].keys() and hospital_denoms[h][measure]
!= None:
                                Snkhd.append(hospital_denoms[h][measure])
                nkhd = hospital_denoms[hospital][measure]
                if nkhd == None:
                        return 1
                denom_info[measure] = {}
                denom_info[measure]["sum"] = sum(Snkhd)
                denom_info[measure]["count"] = len(Snkhd)
                return (nkhd / denom_info[measure]["sum"]) * denom_info[measure]["count"]

def training_data(group, hospitals):
        global direction_features
        global measure_groups
        train = []
        for hospital in hospitals:
                h = []
                no_vals = True
                min_vals = 3
                for measure in measure_groups[group]:
                        if hospitals[hospital][measure] != None:
                                min_vals -= 1
                                if min_vals < 1:
                                        no_vals = False
                                h.append((hospitals[hospital][measure] *
sample_variation_weight(measure, hospital) * direction_features[measure]).real)
                        else:
                                h.append(np.nan)
                if not no_vals:
                        train.append(h)
        if train:
                return train

def test_data(hospital, group, hospitals):
        global direction_features
        global measure_groups
        test = []
        h = []
        for measure in measure_groups[group]:
```

```
            if hospitals[hospital][measure] != None:
                h.append(hospitals[hospital][measure] * direction_features[measure])
            else:
                h.append(np.nan)
        test.append(h)
        return test

def make_preprocessing(group, hospitals):
        train = training_data(group, hospitals)

        imp = preprocessing.Imputer(missing_values="NaN", strategy="mean", axis=0).fit(train)
        train = imp.transform(train)

        scaler = preprocessing.StandardScaler().fit(train)
        train = scaler.transform(train)
        return [train, scaler, imp]

def factor_analyzer(train):
        FA = decomposition.FactorAnalysis(n_components=1, svd_method="lapack")
        return FA.fit(train)

def make_models():
        global measure_groups
        FA_measure_groups = {}
        for group in measure_groups:
                pp = make_preprocessing(group, hospitals)
                FA_measure_groups[group] = {"training_data":pp[0], "scaler":pp[1],
"imputer":pp[2], "factor_analyzer":factor_analyzer(pp[0])}
        return FA_measure_groups

def group_score(hospital, group, hospitals, models, override_min=False):
        d = test_data(hospital, group, hospitals)
        no_vals = True
        min_vals = 3
        for i in d[0]:
                if not math.isnan(i):
                        min_vals -= 1
                        if min_vals < 1 or override_min:
                                no_vals = False
        if no_vals:
                return "No Score"
        d = models[group]["imputer"].transform(d)
        d = models[group]["scaler"].transform(d)
        return models[group]["factor_analyzer"].transform(d)[0][0]

def summary_score(hospital, hospitals, models):
        global measure_groups
        global weights
        scores = {}
        for group in measure_groups:
```

```
                scores[group] = group_score(hospital, group, hospitals, models)
        s = 0
        ss_names = []
        wf = []
        ss = []
        for group in scores:
                if scores[group] == "No Score":
                        wf.append(0)
                        ss.append(-999)
                else:
                        s += 1
                        wf.append(weights[group])
                        ss.append(scores[group])
                ss_names.append(group)
        if s < 3 or (scores["mortality"] == "No Score" and scores["safety"] == "No Score" and
scores["readmission"] == "No Score"):
                return ["No Score", scores]
        for i in range(len(ss_names)):
                if ss[i] == -999:
                        recalc = group_score(hospital, ss_names[i], hospitals, models,
override_min=True)
                        if recalc != "No Score":
                                wf[i] = weights[ss_names[i]]
                                ss[i] = recalc
                                scores[ss_names[i]] = ss[i]
                        else:
                                scores[ss_names[i]] = "No Score"
                else:
                        scores[ss_names[i]] = ss[i]

        wf = [i/sum(wf) for i in wf]
        return [np.average(ss, weights=wf), scores]

gmaps = googlemaps.Client(key='AIzaSyB3yHil_Mals5ls4Nelxwy0ClhTroXsCbk') #gmaps API key
medicare_client = Socrata("data.medicare.gov", "v1BUR3g8iQ3vzlC44sPvnXbju")

models = make_models()

##GUI##
import easygui

def nearby_hospital_scores_gui(location, hscores, googlemaps=True):
        global gmaps
        models = make_models()
        geo = gmaps.geocode(location)
        nhos = []
        if googlemaps:
                nearby = gmaps.places(query="hospital", type='hospital',
location=geo[0]["geometry"]["location"], language='en-US', radius=50000)
                for h in nearby["results"]:
```

```
                        hospital_info = medicare_client.get("rbry-mqwu", where=f"hospital_name
LIKE \"{h['name'].upper()}\"", limit=1)
                        if hospital_info and hospital_info[0]["hospital_name"] in
hscores.keys():
                                nhos.append([hospital_info[0]["hospital_name"],
hscores[hospital_info[0]["hospital_name"]]])
        return nhos

def gui(hospitals):
        easygui.msgbox('Find the best hospital near you. ~Team # 2018032', 'HealthSearch')
        msg = "Enter your location"
        title = "HealthSearch"

        fieldNames = ["Address"]
        fieldValues = []  # we start with blanks for the values
        fieldValues = easygui.multenterbox(msg,title,fieldNames)

        msg = "Rate the importance of each of these factors holds to you on a scale of one to
five as well as the severity of your symptoms on a scale of zero to ten."
        surveyNames = ["Timeliness of Care", "Imaging Technology", "Doctor Communication",
"Pain Control", "Quiet Conditions", "Sanitary Measures", "Severity of Symptoms"]
        surveyValues = []  # we start with blanks for the values
        surveyValues = easygui.multenterbox(msg,title,surveyNames)

        nhos = nearby_hospital_scores_gui(fieldValues[0], hospital_all_scores(hospitals))
        msg = "Here are the best nearby hospitals for you:"
        choices = [str(l[0]) + " | Quality Score: " + str(round(l[1][0])) for l in nhos]
        choice  = easygui.choicebox(msg, title, choices)

gui(hospitals)
```

HSMR Logistic Regression

```
# IMMC 2018 HSMR Logistic Regression Model Calculation
# Team # 2018032

import sklearn
import pandas
from sklearn.metrics import classification_report
from sklearn.preprocessing import MinMaxScaler
import pandas_ml as pdml
from imblearn.combine import SMOTETomek
import time
t0 = time.time()
import numpy
from sodapy import Socrata

client = Socrata("health.data.ny.gov", "Y4nGvKjaPPX0Hs04klcxbEKsN",
username="crooksnoah@gmail.com", password="Slypie11!")

l = 1645096
results = pandas.read_csv("D:\Hospital_Inpatient_Discharges__SPARCS_De-Identified___2016.csv")
#client.get("y93g-4rqn", limit = l)

e = 0
# Convert to pandas DataFrame
aresults = results

#results_df.filter(items=["Age Groups", "apr_drg_description", "apr_mdc_description", "APR
Medical Surgical Description", "apr_risk_of_mortality",  "Facility Name", "
patient_disposition", "Payment Typology 1", "Race","Type of Admission"])
aresults = aresults.rename(columns={"Patient Disposition":"y"})

print aresults
test1 = aresults.loc[(aresults["Facility Id"] == 1450.0)]
test1 =
test1.drop(test1.columns[[0,1,2,3,4,6,9,13,13,14,15,16,17,18,19,22,23,24,24,27,28,30,31,33,34,
36, 35, 20, 21, 32, 29]], axis=1)
print test1.columns
print test1
test1 = test1.reset_index()
locate1 = test1.loc
del test1['index']
for i in xrange(0,len(test1)):
    if locate1[i,"y"] == "Expired":
        locate1[i,"y"] = 1

    else:
        locate1[i,"y"] = 0
    if locate1[i, "Gender"] == "M":
        locate1[i,"Gender"] = 1
    else:
        locate1[i,"Gender"] = 0
    if locate1[i, "APR Medical Surgical Description"] == "Medical":
        locate1[i, "APR Medical Surgical Description"] = 1
    else:
        locate1[i, "APR Medical Surgical Description"] = 0
    if locate1[i,"Age Group"] == "0 to 17":
        locate1[i,"Age Group"] = 0
    elif locate1[i,"Age Group"] == "18 to 29":
        locate1[i,"Age Group"] = 1
    elif locate1[i,"Age Group"] == "30 to 49":
```

```
        locate1[i,"Age Group"] = 2
    else:
        locate1[i,"Age Group"] = 3
    if locate1[i, "Race"] == "White":
        locate1[i, "Race"] = 0
    else:
        locate1[i,"Race"] = 1
    if locate1[i,"Type of Admission"] == "Elective":
        locate1[i,"Type of Admission"] = 0
    elif locate1[i,"Type of Admission"] == "Emergency":
        locate1[i,"Type of Admission"] = 2
    else:
        locate1[i,"Type of Admission"] = 1
    if locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 0
    elif locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 1
    elif locate1[i, "Payment Typology 1"] == "Blue Cross/Blue Shield":
        locate1[i, "Payment Typology 1"] = 2
    else:
        locate1[i, "Payment Typology 1"] = 3
    if locate1[i,"y"] == 1:
        e = e+1
print e
print "1"
test2 = aresults.loc[(aresults["Facility Id"] == 1456.0)]
test2 =
test2.drop(test2.columns[[0,1,2,3,4,6,9,13,13,14,15,16,17,18,19,22,23,24,24,27,28,30,31,33,34,
36, 35, 20, 21, 32, 29]], axis=1)

test2 = test2.reset_index()
locate1 = test2.loc
del test2['index']
for i in xrange(0,len(test2)):
    if locate1[i,"y"] == "Expired":
        locate1[i,"y"] = 1

    else:
        locate1[i,"y"] = 0
    if locate1[i, "Gender"] == "M":
        locate1[i,"Gender"] = 1
    else:
        locate1[i,"Gender"] = 0
    if locate1[i, "APR Medical Surgical Description"] == "Medical":
        locate1[i, "APR Medical Surgical Description"] = 1
    else:
        locate1[i, "APR Medical Surgical Description"] = 0
    if locate1[i,"Age Group"] == "0 to 17":
        locate1[i,"Age Group"] = 0
    elif locate1[i,"Age Group"] == "18 to 29":
        locate1[i,"Age Group"] = 1
    elif locate1[i,"Age Group"] == "30 to 49":
        locate1[i,"Age Group"] = 2
    else:
        locate1[i,"Age Group"] = 3
    if locate1[i, "Race"] == "White":
        locate1[i, "Race"] = 0
    else:
        locate1[i,"Race"] = 1
    if locate1[i,"Type of Admission"] == "Elective":
        locate1[i,"Type of Admission"] = 0
```

```
    elif locate1[i,"Type of Admission"] == "Emergency":
        locate1[i,"Type of Admission"] = 2
    else:
        locate1[i,"Type of Admission"] = 1
    if locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 0
    elif locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 1
    elif locate1[i, "Payment Typology 1"] == "Blue Cross/Blue Shield":
        locate1[i, "Payment Typology 1"] = 2
    else:
        locate1[i, "Payment Typology 1"] = 3
print "2"
test3 = aresults.loc[(aresults["Facility Id"] == 1438.0)]
test3 =
test3.drop(test3.columns[[0,1,2,3,4,6,9,13,13,14,15,16,17,18,19,22,23,24,24,27,28,30,31,33,34,
36, 35, 20, 21, 32, 29]], axis=1)

test3 = test3.reset_index()
locate1 = test3.loc
del test3['index']
for i in xrange(0,len(test3)):
    if locate1[i,"y"] == "Expired":
        locate1[i,"y"] = 1

    else:
        locate1[i,"y"] = 0
    if locate1[i, "Gender"] == "M":
        locate1[i,"Gender"] = 1
    else:
        locate1[i,"Gender"] = 0
    if locate1[i, "APR Medical Surgical Description"] == "Medical":
        locate1[i, "APR Medical Surgical Description"] = 1
    else:
        locate1[i, "APR Medical Surgical Description"] = 0
    if locate1[i,"Age Group"] == "0 to 17":
        locate1[i,"Age Group"] = 0
    elif locate1[i,"Age Group"] == "18 to 29":
        locate1[i,"Age Group"] = 1
    elif locate1[i,"Age Group"] == "30 to 49":
        locate1[i,"Age Group"] = 2
    else:
        locate1[i,"Age Group"] = 3
    if locate1[i, "Race"] == "White":
        locate1[i, "Race"] = 0
    else:
        locate1[i,"Race"] = 1
    if locate1[i,"Type of Admission"] == "Elective":
        locate1[i,"Type of Admission"] = 0
    elif locate1[i,"Type of Admission"] == "Emergency":
        locate1[i,"Type of Admission"] = 2
    else:
        locate1[i,"Type of Admission"] = 1
    if locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 0
    elif locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 1
    elif locate1[i, "Payment Typology 1"] == "Blue Cross/Blue Shield":
        locate1[i, "Payment Typology 1"] = 2
    else:
        locate1[i, "Payment Typology 1"] = 3
```

```
print "3"
test4 = aresults.loc[(aresults["Facility Id"] == 1463.0)]
test4 =
test4.drop(test4.columns[[0,1,2,3,4,6,9,13,13,14,15,16,17,18,19,22,23,24,24,27,28,30,31,33,34,
36, 35, 20, 21, 32, 29]], axis=1)

test4 = test4.reset_index()
locate1 = test4.loc
del test4['index']
for i in xrange(0,len(test4)):
    if locate1[i,"y"] == "Expired":
        locate1[i,"y"] = 1

    else:
        locate1[i,"y"] = 0
    if locate1[i, "Gender"] == "M":
        locate1[i,"Gender"] = 1
    else:
        locate1[i,"Gender"] = 0
    if locate1[i, "APR Medical Surgical Description"] == "Medical":
        locate1[i, "APR Medical Surgical Description"] = 1
    else:
        locate1[i, "APR Medical Surgical Description"] = 0
    if locate1[i,"Age Group"] == "0 to 17":
        locate1[i,"Age Group"] = 0
    elif locate1[i,"Age Group"] == "18 to 29":
        locate1[i,"Age Group"] = 1
    elif locate1[i,"Age Group"] == "30 to 49":
        locate1[i,"Age Group"] = 2
    else:
        locate1[i,"Age Group"] = 3
    if locate1[i, "Race"] == "White":
        locate1[i, "Race"] = 0
    else:
        locate1[i,"Race"] = 1
    if locate1[i,"Type of Admission"] == "Elective":
        locate1[i,"Type of Admission"] = 0
    elif locate1[i,"Type of Admission"] == "Emergency":
        locate1[i,"Type of Admission"] = 2
    else:
        locate1[i,"Type of Admission"] = 1
    if locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 0
    elif locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 1
    elif locate1[i, "Payment Typology 1"] == "Blue Cross/Blue Shield":
        locate1[i, "Payment Typology 1"] = 2
    else:
        locate1[i, "Payment Typology 1"] = 3
print "4"
test5 = aresults.loc[(aresults["Facility Id"] == 1439.0)]
test5 =
test5.drop(test5.columns[[0,1,2,3,4,6,9,13,13,14,15,16,17,18,19,22,23,24,24,27,28,30,31,33,34,
36, 35, 20, 21, 32, 29]], axis=1)
test5 = test5.reset_index()
locate1 = test5.loc
del test5['index']
for i in xrange(0,len(test5)):
    if locate1[i,"y"] == "Expired":
        locate1[i,"y"] = 1
```

```
    else:
        locate1[i,"y"] = 0
    if locate1[i, "Gender"] == "M":
        locate1[i,"Gender"] = 1
    else:
        locate1[i,"Gender"] = 0
    if locate1[i, "APR Medical Surgical Description"] == "Medical":
        locate1[i, "APR Medical Surgical Description"] = 1
    else:
        locate1[i, "APR Medical Surgical Description"] = 0
    if locate1[i,"Age Group"] == "0 to 17":
        locate1[i,"Age Group"] = 0
    elif locate1[i,"Age Group"] == "18 to 29":
        locate1[i,"Age Group"] = 1
    elif locate1[i,"Age Group"] == "30 to 49":
        locate1[i,"Age Group"] = 2
    else:
        locate1[i,"Age Group"] = 3
    if locate1[i, "Race"] == "White":
        locate1[i, "Race"] = 0
    else:
        locate1[i,"Race"] = 1
    if locate1[i,"Type of Admission"] == "Elective":
        locate1[i,"Type of Admission"] = 0
    elif locate1[i,"Type of Admission"] == "Emergency":
        locate1[i,"Type of Admission"] = 2
    else:
        locate1[i,"Type of Admission"] = 1
    if locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 0
    elif locate1[i, "Payment Typology 1"] == "Medicare":
       locate1[i, "Payment Typology 1"] = 1
    elif locate1[i, "Payment Typology 1"] == "Blue Cross/Blue Shield":
        locate1[i, "Payment Typology 1"] = 2
    else:
        locate1[i, "Payment Typology 1"] = 3
print "5"
aresults =
aresults.drop(aresults.columns[[0,1,2,3,4,6,9,13,13,14,15,16,17,18,19,22,23,24,24,27,28,30,31,
33,34,36, 35, 20, 21, 32, 29]], axis=1)
#results_df = results_df.drop(results_df.columns[[5,6,7,8,10,12,13,15,16,18]], axis=1)

aresults = aresults.loc[(aresults["Type of Admission" ]!="Newborn")]
aresults = aresults.loc[(aresults["y"] != "Left Against Medical Advice")]
aresults = aresults.dropna()
aresults = aresults.reset_index()
del aresults['index']



#test1 = locate[:"Facility Name"] == "New York Presbyterian Hospital - Columbia Presbyterian
Center" | "New York Presbyterian Hospital - New York Weill Cornell Center"
#print test1
aresults = aresults.drop(aresults.index[100001:])
print aresults
for i in xrange(0,len(aresults)):
    if aresults.loc[i,"y"] == "Expired":
        aresults.loc[i,"y"] = 1
    else:
        aresults.loc[i,"y"] = 0
    if aresults.loc[i, "Gender"] == "M":
```

```
        aresults.loc[i,"Gender"] = 1
    else:
        aresults.loc[i,"Gender"] = 0
    if aresults.loc[i, "APR Medical Surgical Description"] == "Medical":
        aresults.loc[i, "APR Medical Surgical Description"] = 1
    else:
        aresults.loc[i, "APR Medical Surgical Description"] = 0
    if aresults.loc[i,"Age Group"] == "0 to 17":
        aresults.loc[i,"Age Group"] = 0
    elif aresults.loc[i,"Age Group"] == "18 to 29":
       aresults.loc[i,"Age Group"] = 1
    elif aresults.loc[i,"Age Group"] == "30 to 49":
        aresults.loc[i,"Age Group"] = 2
    else:
        aresults.loc[i,"Age Group"] = 3
    if aresults.loc[i, "Race"] == "White":
        aresults.loc[i, "Race"] = 0
    else:
        aresults.loc[i,"Race"] = 1
    if aresults.loc[i,"Type of Admission"] == "Elective":
        aresults.loc[i,"Type of Admission"] = 0
    elif aresults.loc[i,"Type of Admission"] == "Emergency":
        aresults.loc[i,"Type of Admission"] = 2
    else:
        aresults.loc[i,"Type of Admission"] = 1
    if aresults.loc[i, "Payment Typology 1"] == "Medicare":
       aresults.loc[i, "Payment Typology 1"] = 0
    elif aresults.loc[i, "Payment Typology 1"] == "Medicare":
       aresults.loc[i, "Payment Typology 1"] = 1
    elif aresults.loc[i, "Payment Typology 1"] == "Blue Cross/Blue Shield":
        aresults.loc[i, "Payment Typology 1"] = 2
    else:
        aresults.loc[i, "Payment Typology 1"] = 3
print "aresults"
#print df
test1 = test1.replace("120 +", 120)
test2 = test2.replace("120 +", 120)
test3 = test3.replace("120 +", 120)
test4 = test4.replace("120 +", 120)
test5 = test5.replace("120 +", 120)
scaler = MinMaxScaler()

test1 = pdml.ModelFrame(scaler.fit_transform(test1), columns = test1.columns)
test2 = pdml.ModelFrame(scaler.fit_transform(test2), columns = test2.columns)
test3 = pdml.ModelFrame(scaler.fit_transform(test3), columns = test3.columns)
test4 = pdml.ModelFrame(scaler.fit_transform(test4), columns = test4.columns)
test5 = pdml.ModelFrame(scaler.fit_transform(test5), columns = test5.columns)



bresults = aresults.replace("120 +", "120")

bresults = pdml.ModelFrame(scaler.fit_transform(bresults), columns = bresults.columns)

import matplotlib.pyplot as plt
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
from sklearn.cross_validation import train_test_split
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
```

```
##creates heatmap to test independence
#sns.heatmap(bresults.corr())
#plt.show()



X = (bresults.as_matrix(bresults.columns[[0,1,2,3,5,6]]))
y = (bresults.as_matrix(bresults.columns[[4]]))
y = y.astype(int)
X = X.astype(int)
X_test1 = test1.as_matrix(test1.columns[[0,1,2,3,5,6]])
X_test1 = X_test1.astype(int)
y_test1 = test1.as_matrix(test1.columns[[4]])
print y_test1
y_test1 =y_test1.astype(int)
X_test2 = test2.as_matrix(test2.columns[[0,1,2,3,5,6]])
X_test3 = test3.as_matrix(test3.columns[[0,1,2,3,5,6]])
X_test4 = test4.as_matrix(test4.columns[[0,1,2,3,5,6]])
X_test5 = test5.as_matrix(test5.columns[[0,1,2,3,5,6]])
y_test2 = test2.as_matrix(test2.columns[[4]])
y_test3 = test3.as_matrix(test3.columns[[4]])
y_test4 = test4.as_matrix(test4.columns[[4]])
y_test5 = test5.as_matrix(test5.columns[[4]])
y=numpy.ravel(y)
y_test2 =y_test2.astype(int)
y_test3 =y_test3.astype(int)
y_test4 =y_test4.astype(int)
y_test5 =y_test5.astype(int)
X_test2 = X_test2.astype(int)
X_test3 = X_test3.astype(int)
X_test4 = X_test4.astype(int)
X_test5 = X_test5.astype(int)
#print X
#print y
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

ros = SMOTETomek( ratio="minority", random_state=42, k=5)
X_res, y_res = ros.fit_sample(X_train, y_train)

classifier = LogisticRegression(random_state=0,C=1, penalty="l2", solver = "sag",
class_weight= {1:1, 0:3})
classifier.fit(X_res, y_res)
y_pred1 = classifier.predict(X_test1)
print y_pred1
y_pred2 = classifier.predict(X_test2)
y_pred3 = classifier.predict(X_test3)
y_pred4 = classifier.predict(X_test4)
y_pred5 = classifier.predict(X_test5)
from sklearn.metrics import confusion_matrix

confusion_matrix1 = confusion_matrix(y_test1, y_pred1)
confusion_matrix2 = confusion_matrix(y_test2, y_pred2)
confusion_matrix3 = confusion_matrix(y_test3, y_pred3)
confusion_matrix4 = confusion_matrix(y_test4, y_pred4)
confusion_matrix5 = confusion_matrix(y_test5, y_pred5)
print confusion_matrix1
print confusion_matrix2
print confusion_matrix3
print confusion_matrix4
```

```
print confusion_matrix5
prob1 = classifier.predict_proba(X_test1)
prob2 = classifier.predict_proba(X_test2)
prob3 = classifier.predict_proba(X_test3)
prob4 = classifier.predict_proba(X_test4)
prob5 = classifier.predict_proba(X_test5)
print(classification_report(y_test1, y_pred1, target_names=["alive", "dead"]))

roc = sklearn.metrics.roc_auc_score(y_test1,y_pred1)
print sum(prob1)
print sum(prob2)
print sum(prob3)
print sum(prob4)
print sum(prob5)
print roc

print('Accuracy of logistic regression classifier on test set:
{:.2f}'.format(classifier.score(X_res, y_res)))
```

Derivations

Derivation of Scikit-Learn's Logistic Regression Minimization Function

Adapted from a 2015 Stackexchange answer by user "YuppieNetworking"
(https://stats.stackexchange.com/questions/186830/what-is-scikit-learns-logisticregression-minimizing)

Let ((X_i, y_i) $\forall i \in \{1...n\}$ be pairs of (features, class) where X_i is a column vector. The class $y_i \in \{1, -1\}$ consists exclusively of those values in the set. We wish to model the following probability:

$$p(y=1|X;w,c) = g(wX_i^T + c) = \frac{1}{1 + e^(-(wX_i^T + c))}.$$

where c, w are weights and the intercept of the logistic regression model.

To obtain the optimal w, c, it is necessary to maximize the likelihood given the database of labeled data. The optimization problem is:

$$\underset{\theta}{argmax} \quad \mathcal{L}(w, c; X_1, \ldots, X_m)$$

$$= \prod_{i, y_i = 1} p(y=1|X_i; w, c) \prod_{i, y_i = -1} p(y=-1|X_i; w, c)$$

$\langle$There are only two classes, so $p(y=-1|\ldots) = 1 - p(y=1|\ldots)\rangle$

$$= \prod_{i, y_i = 1} p(y=1|X_i; w, c) \prod_{i, y_i = -1} (1 - p(y=1|X_i; w, c))$$

$\langle$Definition of $p\rangle$

$$= \prod_{i, y_i = 1} g(X_i^T w + c) \prod_{i, y_i = -1} (1 - g(X_i^T w + c))$$

$\langle$Useful property: $1 - g(z) = g(-z)\rangle$

$$= \prod_{i, y_i = 1} g(X_i^T w + c) \prod_{i, y_i = -1} g(-(X_i^T w + c))$$

$\langle$Handy trick of using +1/-1 classes: multiply by y_i to have a common product$\rangle$

$$= \prod_{i=1}^{m} g(y_i(X_i^T w + c))$$

Next, we apply the logarithm function, which monotonically increases (see any real analysis textbook), and convert the maximization problem into a minimization problem, by multiplying by negative one, we see that:

$$\underset{\theta}{argmin} \quad -\log\left(\prod_{i=1}^{m} g(y_i(X_i^T w + c))\right)$$

$\langle\log(a \cdot b) = \log a + \log b\rangle$

$$= -\sum_{i=1}^{m} \log g(y_i(X_i^T w + c))$$

$\langle$definition of $g\rangle$

$$= -\sum_{i=1}^{m} \log \frac{1}{1 + \exp(-y_i(X_i^T w + c))}$$

$\langle \log(a/b) = \log a - \log b \rangle$

$$= -\sum_{i=1}^{m} \log 1 - \log(1 + \exp(-y_i(X_i^T w + c)))$$

$\langle \log 1$ is a constant, so it can be ignored$\rangle$

$$= -\sum_{i=1}^{m} - \log(1 + \exp(-y_i(X_i^T w + c)))$$

$$= \sum_{i=1}^{m} \log(\exp(-y_i(X_i^T w + c)) + 1)\,,$$