U.S. Hospital Rating Model

*Based on patient survey and complications data from CMS*

June 5th, 2019

Marin Gow, Michael Diarra, Venku Buragadda, Zain Iqbal, Zain Jafri

MScA, University of Chicago

Table of Contents

[Background 3](#_Toc10620075)

[Literature Review 4](#_Toc10620076)

[Hypothesis 4](#_Toc10620077)

[Data Overview 5](#_Toc10620078)

[Datasets 5](#_Toc10620079)

[Data Exploration 5](#_Toc10620080)

[Data Preparation 9](#_Toc10620081)

[Data Imputation 9](#_Toc10620082)

[Models Overview 9](#_Toc10620083)

[Classification Models Used 10](#_Toc10620084)

[Latent Dirichlet Allocation (LDA) 10](#_Toc10620085)

[K-Nearest Neighbors (KNN) 10](#_Toc10620086)

[Support Vector Machine (SVM) 10](#_Toc10620087)

[Gradient Boosted Classifier 10](#_Toc10620088)

[Random Forest 10](#_Toc10620089)

[Regression Models Used 11](#_Toc10620090)

[Linear Regression 11](#_Toc10620091)

[Gradient Boosted Regression 11](#_Toc10620092)

[Random Forest Regressor 11](#_Toc10620093)

[XGBoost 11](#_Toc10620094)

[Model Results 12](#_Toc10620095)

[Classification Model Results 12](#_Toc10620096)

[Regression Model Results 12](#_Toc10620097)

[Model Selection 13](#_Toc10620098)

[Model Validation 13](#_Toc10620099)

[K-Fold Cross Validation 13](#_Toc10620100)

[Accuracy Paradox 14](#_Toc10620101)

[Area Under The Curve Receiver Operating Characteristics Curve (AUC ROC Curve) 15](#_Toc10620102)

[Conclusion 16](#_Toc10620103)

[Recommendation 16](#_Toc10620104)

[Challenges 16](#_Toc10620105)

[Model Improvements 17](#_Toc10620106)

[Citations 17](#_Toc10620107)

# Background

Over the past ten years, the power of consumers has grown significantly due to the rise of mobile and internet-based applications allowing consumers to evaluate the quality of the goods and services they purchase. This has been the case in nearly every industry from restaurants, entertainment, travel and retail with large platforms quickly arising, including Tripadvisor, Yelp, and Consumer reports (among others). The impact of this trend has been significant: giving consumers previously unseen levels of influence and power to direct the purchasing decisions of their fellow consumers. In addition, businesses have adapted their conduct accordingly, often giving outsized attention to these new platforms and the feedback of consumers.

During the same time period (and in some cases even earlier), the major healthcare regulatory body – the Centers for Medicare & Medicaid Services (CMS) – has also invested significantly in developing programs to measure

This broad trend has, however, failed to materialize within the United States healthcare industry. Over the past two decades, the major regulatory body – the Centers for Medicare & Medicaid Services (CMS) – and other stakeholders, including federal/local governments, not-for-profits, and private sector organizations, have made significant investments in developing programs to measure and improve the quality and performance of both healthcare providers and payers. Programs CMS has launched include, but are not limited to, the following:

* *Hospital Consumer Assessment of Healthcare Providers and Systems (HCAHPS):* A patient satisfaction survey required by CMS for all hospitals in the United States
* *Hospital Compare:* A consumer-oriented website that provides data on the performance of hospitals’ delivery of care, including the following:
  + Prevalence of complications and deaths (among other measures) and;
  + An overall Star Rating for the quality of care delivered at the hospital

Despite these efforts, there are significant limitations and challenges in the industry, which have limited the effectiveness of these programs. Some major challenges include the following:

* Data collection and reporting is not timely with data and reports often lagging anywhere from 1-3 years (e.g., many of the measures used to calculate a hospital’s Star Rating are often 3 years old by the time the rating is reported)
* Reporting is not easily accessible or interpretable by consumers, including a lack of transparency into the measurement and calculation of quality and performance metrics

In addition to the challenges around data accessibility and timeliness, there are also significant market challenges that exist in the US healthcare industry. Namely, the dominance of a third-party payer model within the US healthcare sector often obscures the purchasing decision and creates perverse incentives for all many stakeholders and parties to support accessibility and transparency of the performance of healthcare providers. Also given healthcare is often viewed as a *necessity* vs. consumer good or service, real limitations exist on the scope of any quality or performance ranking program; i.e., they are often limited to non-emergency services. However, these trends are starting to change as government and private sector forces start to shift the current market environment from one that is built on a Fee-for-Service (FFS) model to a Fee-for-Value (FFV) model; and the demands for transparency increase.

With this context in mind, we see a growing need for a more accessible, transparent and timely reporting the quality and performance of healthcare providers in the US. We sought out to develop a predictive model, which utilizes the *HCAHPS Patient Survey* and *Hospital Compare Prevalence of Complications and Deaths* data sets to predict the overall quality and performance of hospitals within the United States. Based upon our data set, we had the opportunity to predict two, related measures from our data sets:

* *Hospital Star Rating:* A categorical measure of overall hospital quality and performance on a scale of 1-5 with 5 being the highest
* *Hospital Overall Recommend Score:* A numerical measure of hospital quality performance on a scale of 0-100 with 100 being the highest and 57 the lowest recorded value

We explored using both measures to develop a classification model for the *Hospital Star Rating* measure and a regression model for the *Hospital Overall Recommend Score*. This presented a unique opportunity to evaluate both measures and the use of a classification or a regression model for our business case. Ultimately, our goal is to use our recommended final model for a more timely application patients can use to determine which hospitals they utilize for non-emergency care.

# Literature Review

Given today’s trend toward consumerism across industries *(Deloitte 2019),* it is unsurprising that many hospitals and healthcare organizations have begun to use predictive modeling to rate hospitals based on a variety of factors. The number of touchpoints at which patient data is collected has continued to grow. These touchpoints include pre-care appointment scheduling and phone calls, medical records and other data taken down during the visit, survey and follow-up care data collected after the visit, billing information, a wealth of demographic and lifestyle information submitted to insurance and care coordination, and more. *(Anisingaraju et al., 2019)*

One 2018 study performed a patient sentiment analysis using social media and comparing to hospital HCAPS star ratings (the same measure targeted by our classification models). The study counted positive reviews left by patients on hospitals’ Facebook pages and concluded that positive comments were associated with both higher star ratings and higher willingness to recommend the hospital to others. *(Huppertz, 2018)*

In addition to research, machine learning is being used by professional services firms to help hospitals deliver precision care. One example is HEALTH[at]SCALE, a company founded in 2015 that specializes in using machine learning models to predict individual care needs and match patients with the correct providers and services at the right time.

# Hypothesis

Our initial hypothesis was that for our business case the *Hospital Star Rating* Classification Model would perform better and would be an easier model to implement and interpret from a business perspective (i.e., we predicted *Hospital X* had a 3-star rating). Furthermore, we estimated we would be able to achieve an accuracy of 80% (i.e., our predicted score would be correct 80% of the time). In terms of the *Hospital Overall Recommend Score* Regression Model, we felt that, while this model would be valuable, from a business perspective the resultant score would not be as easy for consumers to interpret. From a modeling perspective, our initial hypothesis was that we could achieve an RMSE of 2 (i.e., we could predict the final score for a given hospital within 2 points on a scale of 0-100 (57-100 were the scores present in the data).

# Data Overview

## Datasets

Our data source is from CMS which is government body that runs Medicare and Medicaid and audits the performance of the U.S. hospitals and healthcare organizations.

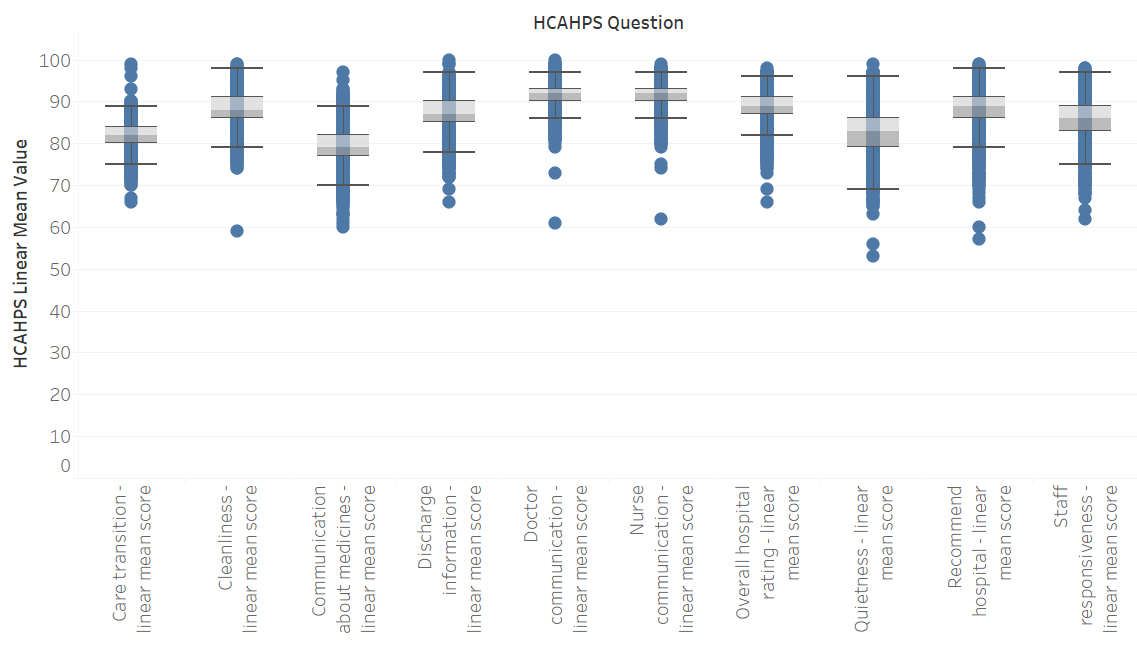
We selected two datasets for the project:

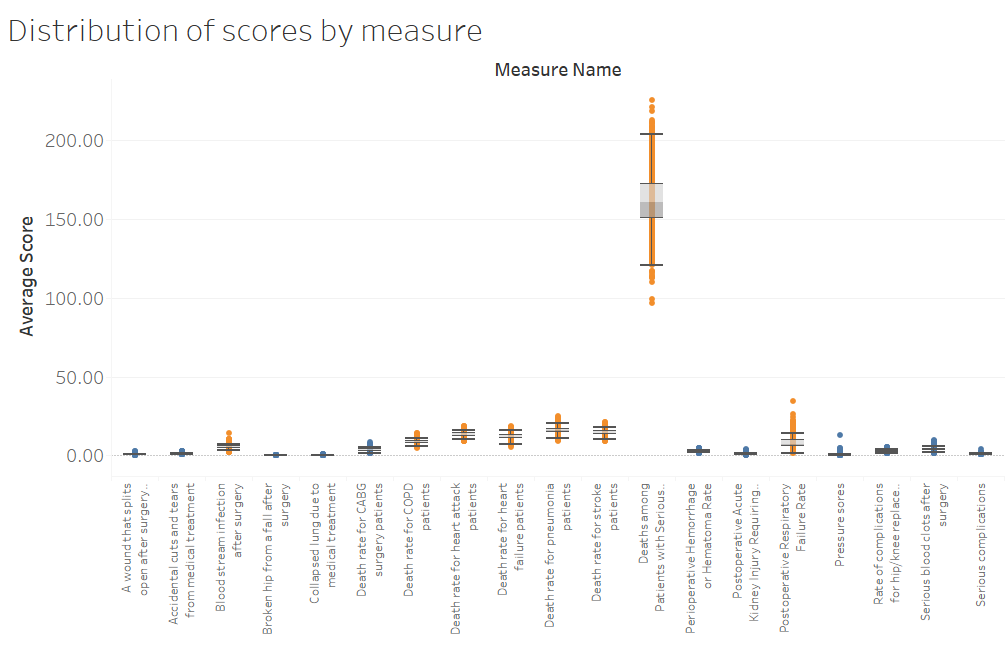
1. Patients Survey scores: A list of hospital ratings for the Hospital Consumer Assessment of Healthcare Providers and Systems (HCAHPS). HCAHPS is a national, standardized survey of hospital patients about their experiences during a recent inpatient hospital stay.
2. Complications and Deaths: Complications and deaths - provider data. This data set includes provider-level data for the hip/knee complication measure, the CMS Patient Safety Indicators, and 30-day death rates.

## Data Exploration

Our preliminary data exploration included the following:

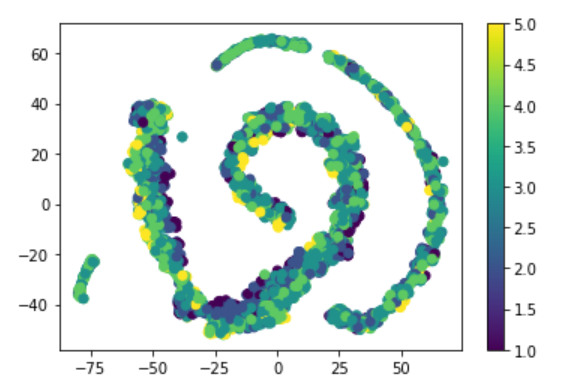
1. Count of records
   1. Survey dataset: We have approx. 250K records in survey dataset, as all the surveys are results for each hospital are in row format. Later we converted these rows in columns as our predictors.
   2. Complications dataset: We have approx. 100K records in the complication dataset where each different complication score for each of the hospital is in row format.
2. Count of records by state
   1. Survey dataset: Texas, California, Florida, Illinois, Pennsylvania are the top 5 states were most of the hospitals were surveyed.
   2. Complications dataset:  Texas, California, Florida, Illinois, New York, are the top 5 states that has most hospitals scores.
3. Boxplot of each of the datasets. From our boxplots, we were able to determine that the survey data was relatively evenly distributed, while the complications data had a significant outlier in the rate of Deaths Among Patients with Serious Complications. Because of this, we knew standardization of our data would be important for distance-based models such as SVM.



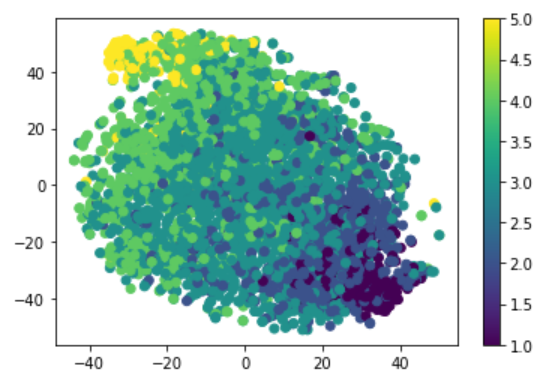


1. Apply tSNE: We applied tSNE on the combined dataset to check for natural clustering in the data. The data points were color-coded by hospital star rating to bring to light any correlation between clustering and star rating. There was some clustering in the data before standardization, but it seemed unrelated to star rating. After standardization, the data formed one large cluster, but there was a clearly visible pattern with the highest rated hospitals appearing in the upper left-hand corner and the lowest rated hospitals in the bottom right hand.

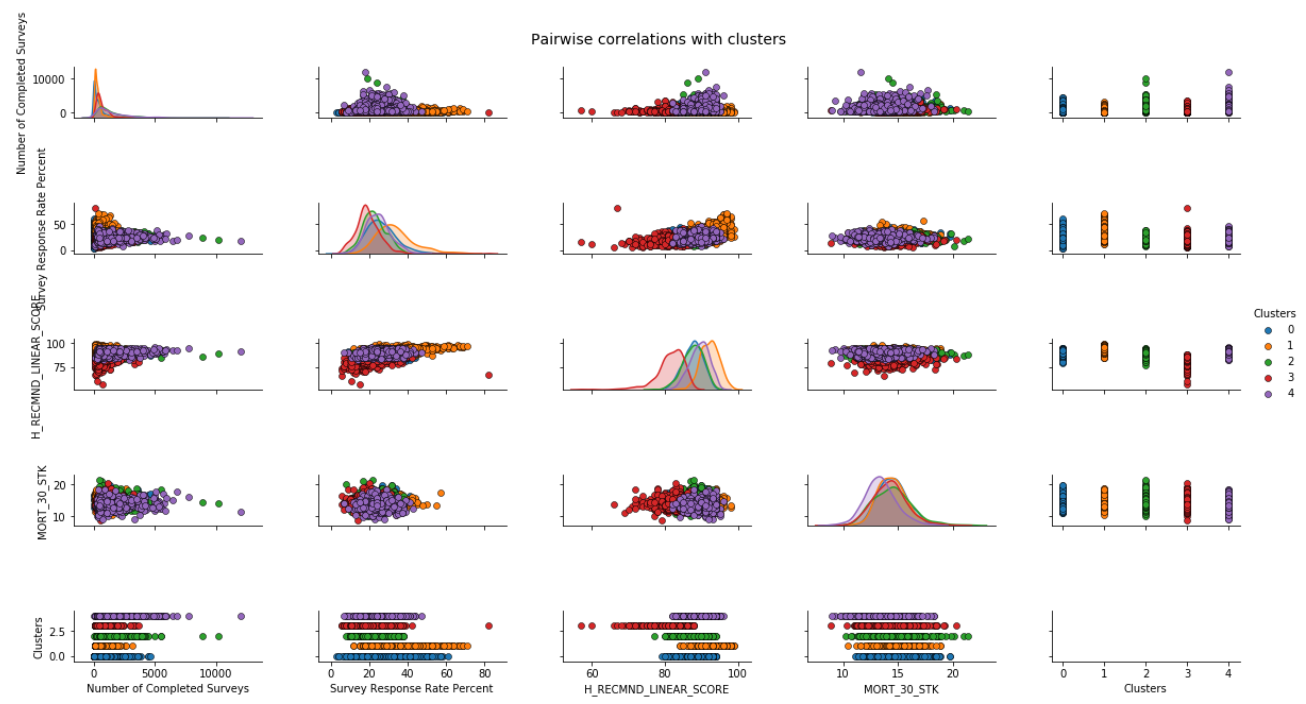
*Before standardization; some clustering present, but no clear pattern in star ratings*



*After standardization; data forms one large cluster, but there is a clear pattern in location of star ratings*



1. K-Means: We produced a pairwise grid of our K-Means clusters (k=5) to examine the relationships between our clusters and our variables. The clusters appeared to have moderate relationships with some variables, while with others there was no relationship. Even where there was a visible relationship, there was significant overlap between clusters. For this reason, we decided not to include K-Means clusters as an input to our final model.



## Data Preparation

Overall, the datasets selected from CMS required relatively little cleanup. However, several steps were taken to format the data for analysis:

1. The Survey dataset was unpivoted to change the grain from question-response to hospital-score. The original data contained one line per survey response per hospital; our final format contained one line per hospital and one column per survey question.
2. The two datasets were tied together using Provider ID, a consistent hospital identifier used across both the Complications and Surveys datasets.
3. Records missing the selected target variables (star rating and recommended linear score) were removed from the dataset.

## Data Imputation

As part of the data preparation, records without a value for the target variables (star rating and recommended linear score) were removed from the dataset. This left a number of null values in the Complications dataset which were filled using three methods:

1. Mean values
2. Linear regression using Python’s FancyImpute module
3. KNN using Python’s FancyImpute module

Each of these imputation methods were used to output a separate filled dataset. Each dataset was run through correlation analysis to ensure that the filled values did not significantly alter relationships between the predictor variables.

Each filled dataset was run through all the predictive models. Some models showed slight differences in performance between the mean, linear regression, and KNN imputed datasets; others produced the same result across all three. Overall, linear regression performed equal to or better than the other imputation methods across all models.

# Models Overview

We ran eight classification models and four regression models for a total of 12 models. An overview of each model is provided in the sections below.

Selecting an adequate prediction model was a critical part of our project. Each of our 12 models was run on three slightly different datasets (filled using mean values, linear regression and KNN respectively) for a total of 36 model runs. We then proceeded to select the dataset for which the models yielded the best prediction capabilities. In this section, we describe the twelve models that we ran before describing how we selected the Gradient Boosted Regressor as our best model.

# Classification Models Used

We run the following eight classification models: The Latent Dirichlet allocation (LDA) model, the K-Nearest Neighbors model, three Support Vector Machine models, the Gradient Boosted Classifier, and the Random Forest classifier. Below is a brief description of these classification models.

## Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation is a probabilistic model used to assign clusters to data with many natural groupings. It is frequently applied in NLP and other cases where hard groupings are less useful than fuzzy groups. *(Raschka, 2019)*

## K-Nearest Neighbors (KNN)

The KNN algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the *inertia* or within-cluster sum-of-squares (see below). This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields. The k-means algorithm divides a set of N samples X into K disjoint clusters C, each described by the mean μj of the samples in the cluster. The means are commonly called the cluster “centroids”; note that they are not, in general, points from X, although they live in the same space. The K-means algorithm aims to choose centroids that minimize the inertia, or within-cluster sum-of-squares criterion. (*Pedregosa et al., 2019)*

## Support Vector Machine (SVM)

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss. *(Gandhi, 2019)*

## Gradient Boosted Classifier

Gradient boosting decision tree (GBDT) is a widely-used machine learning algorithm, due to its efficiency, accuracy, and interpretability. GBDT achieves state-of-the-art performances in many machine learning tasks, such as multi-class classification [2], click prediction , and learning to rank. *(Ke et al., 2019).* Boosting is an ensemble learning technique. Conceptually, these techniques involve: 1. learning base learners; 2. using all of the models to come to a final prediction. Additive modelling is at the foundation of Boosting algorithms. The idea is simple- form a complex function by adding together a bunch of simpler terms. In Gradient Boosting, a number of simpler models are added together to give a complex final model. *(Mahto, 2019*, *Ke et al., 2019)*.

## Random Forest

The random forests classifier is an ensemble learning technique. With the random forests classifier each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model. *(Pedregosa et al., 2019)*

# Regression Models Used

We run the following four regression models: The Linear Regression model, the Gradient Boosted Regression model, the Random Forest Regressor, and the XGBoost model. Below is a brief description of these regression models.

## Linear Regression

Simple linear regression has a form Y = aX + b where Y and X are called dependent and independent variables. These terms are interchangeably used with a response and explanatory variable. Multiple linear regression extends the equation with a greater number of independent variables such as Y= aX + bX1 + c where Y = Weight, X = Height, and X1 = Gender. If the goal is a prediction or error reduction, linear regression can be used to fit a predictive model to an observed dataset. Moreover, the technique can be applied to quantify the strength of the relationship if the goal is to explain variation in Y that can be attributed to variation in the Xs. *(Kim, 2019, Pedregosa et al., 2019).*

## Gradient Boosted Regression

Gradient Tree Boosting or Gradient Boosted Regression Trees (GBRT) is a generalization of boosting to arbitrary differentiable loss functions. GBRT is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems. Gradient Tree Boosting models are used in a variety of areas including Web search ranking and ecology. The advantages of GBRT are:

* Natural handling of data of mixed type (= heterogeneous features)
* Predictive power
* Robustness to outliers in output space (via robust loss functions)

*(Pedregosa et al., 2019)*

## Random Forest Regressor

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The underlying principles and methods are for the most part similar to the one described above for the Random Forest Classifier. *(Fabisch et al., 2019)*

## XGBoost

In their paper, Chen and Guestrin describe a scalable end- to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. They propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, the authors provide insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost can scale beyond billions of examples using far fewer resources than existing systems and yield better results. *(Chen and Guestrin, 2019)*

# Model Results

After fitting the above models, we tested each model using the following techniques:

## Classification Model Results

We tested each classification model by computing each model fit accuracy, precision, recall and F1 score on the testing dataset.

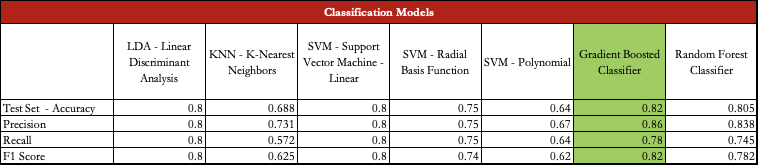
*Accuracy* is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage.

*Precision* is the number of positive predictions divided by the total number of positive class values predicted

*Recall* is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.

*F1 Score* is 2\*((precision\*recall)/(precision+recall)). The F1 score conveys the balance between the precision and the recall.

Below is a summary of the computed values:



Based on the above results we pre-identify the Gradient Boosted Classifier as a good prediction model for our specific problem subject to further testing.

## Regression Model Results

We tested each regression using the RMSE on the testing dataset.

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit (Statistics How To, 2019).

Below is a summary of the computed values:



## Model Selection

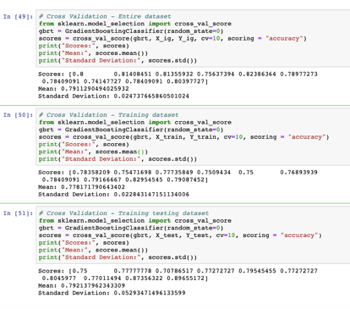
While most of our regression models were performing well based on these initial computed RMSE values, we decided to retain best classification model, the Gradient Boosted Classifier, for further generalizability testing. This decision was based on the business case for our model, which is ultimately meant to provide more up-to-date hospital ratings to potential patients. Our team felt that a star rating was easier to understand for a typical consumer than a score, which is difficult to interpret without context.

# Model Validation

Our main goal was to ensure that our selected model was generalizable and not overfitting or underfitting. To that end, we run the following tests:

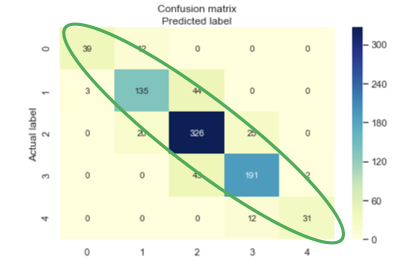
## K-Fold Cross Validation

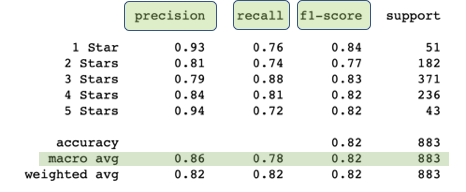
Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation (Brownlee, 2019). We checked the validity of our accuracy score by running a 10-folds cross-validation on the entire dataset, the test dataset, and the training datasets. On comparing our results, we saw roughly equal performance between the folds, which indicates that our model is generalizable.



## Accuracy Paradox

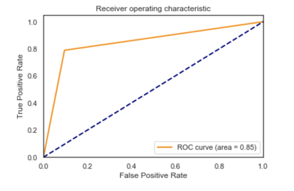
The accuracy paradox is the paradoxical finding that accuracy is not a good metric for predictive models when classifying in predictive analytics. We checked for the accuracy paradox by running a confusion matrix and a Sklearn classification report. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. (GeeksforGeeks, 2019). Below is the result of our computations:





## Area Under The Curve Receiver Operating Characteristics Curve (AUC ROC Curve)

AUC - ROC curve is a performance measurement for classification problems at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher AUC indicates better model performance (Narkhede, 2019). Below is the AUC ROC curve for our Gradient Boost classification model:



Based on the above test, we concluded that our Gradient Boosted Classification model is performing good predictions for our use case and is generalizable.

# Conclusion

## Recommendation

Using Hospital Complication Scores and Patient Surveys the selected Gradient Boosted Classification model predicts Hospital Rating Scores with an accuracy of greater than 80%. The recommended model provides a more timely application patients can use to determine which hospitals they utilize for non-emergency care. This approach is an improvement from the CMS star rating system since CMS star rating evaluation is based solely on patient surveys and does not include hospital complication score performance. In addition, the CMS star ratings lag by 12-18 months and do not include the hospital’s most recent data.

## Challenges

The project team faced the following challenges while attempting to accurately predict the hospital ratings:

1. *Missing data:* Roughly 5% of the records included missing values. As mentioned in the data cleaning section we explored three different imputation methods: mean, KNN, and iterative regression. In addition to feature engineering, selecting the correct imputation method required testing all the models using all three methods to determine which produced the best accuracy score or lowest RMSE.
2. *Model overfitting/underfitting:* Over fitting or under fitting the model is a potential challenge with machine learning models. For the Random Forest Classification and Regression models, the team noticed that the cross validation accuracy on the training dataset was greater than the cross validation accuracy on the testing dataset resulting in an over fitting error. The team adjusted the model parameters to ensure the difference in accuracy between testing and training cross validation was low enough to avoid over fitting error. The selected Gradient Boosted Classification model did not show any indication of over fitting or under fitting.
3. *Clustering:* The team applied various unsupervised learning methods such as k-means, DBSCAN, and hierarchical clustering, however none of the models were valuable for the selected model. With different combinations of complication features, patient rating scores, and hospital locations none of the models produced a useable output. The team decided to continue the analysis without the application of any clustering methods.
4. *Delay in data reporting:* Typically, CMS reporting data contains a 1-3 year reporting lag time. The model was fit on data valued as of July 1, 2017. Given the objective of the project is to develop a more accurate real time rating methodology, for model develop the only option was to reply on data that’s two years old. For the real work application scenario, once we started receiving real-time feedback from patients and hospital complication reporting – the model will be refined.

## Model Improvements

Given additional data sources, the team concluded two model recommendations which would improve the patient application:

1. *Graph Model Ranking:* The selected Gradient Boosted Classification model predicts the hospital rating score. The team explored graph models to eventually rank the hospitals and provide a recommendation to the user. However, there were not enough hospital or patient features in the data to develop meaningful edges and rank accordingly. An example of graph model ranking was included in the presentation showing the hospitals as nodes with characteristics such as star rating, location, patient count, insurance network, and cost of care. The edges of the sample model include the patient characteristics such as complication, insurance network, and location. Using the PageRank feature hospitals can potentially be ranked and recommended to the user.
2. *Additional Complication Scores:* The model could also improve by adding more complication features to the dataset. Currently, CMS complication scores only include 20 different complication types. With additional complications, the model would be able to better predict rating scores based on the users complication. Weights could be added to the PageRank nodes to deliver a bespoke ranking for the user.

# Citations

**Anisingaraju et al** (2019). Healthcare Analytics: Predicting patient experience with narrative data: a healthcare goldmine - Analytics Magazine. [online] Analytics Magazine. Available at: http://analytics-magazine.org/healthcare-analytics-predicting-patient-experience-with-narrative-data-a-healthcare-goldmine/ [Accessed 3 Jun. 2019].

**Brownlee, J.** (2019). *Classification Accuracy is Not Enough: More Performance Measures You Can Use*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/ [Accessed 4 Jun. 2019].

**Brownlee, J.** (2019). *A Gentle Introduction to k-fold Cross-Validation*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/k-fold-cross-validation/ [Accessed 4 Jun. 2019].

**Chen, T. and Guestrin, C.** (2019). *XGBoost: A Scalable Tree Boosting System*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1603.02754.pdf [Accessed 4 Jun. 2019].

**Deloitte (2019).** Www2.deloitte.com. [online] Available at: https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/consumer-business/consumer-review-8-the-growing-power-of-consumers.pdf [Accessed 5 Jun. 2019].

**Fabisch, A.,** Passos, A., Gollonet, A., Joly, A., Gorgolewski, C., Cournapeau, D., Warde-Farley, D., Duchesnay, E., Fabian Pedragosa, F., Louppe, G., Schreiber, J., Vanderplas, J., Grobler, J., Millman, J., Kastner, K., Buitinck, L., Kumar, M., Blondel, M., Brucher, M., Dawe, N., Losi, P., Prettenhofer, P., Rajagopalan, R., Layton, R., Weiss, R., Ghosh, S., Du, S., Jones, T., Dubourg, V., Michel, V., Fritsch, V. and Li, W. (2019). *3.2.4.3.6. sklearn.ensemble.GradientBoostingRegressor — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html [Accessed 4 Jun. 2019].

**Fabisch, A.,** Passos, A., Gollonet, A., Joly, A., Gorgolewski, C., Cournapeau, D., Warde-Farley, D., Duchesnay, E., Pedragosa, F., Louppe, G., Schreiber, J., Vanderplas, J., Grobler, J., Millman, J., Kastner, K., Buitinck, L., Kumar, M., Blondel, M., Brucher, M., Dawe, N., Losi, P., Prettenhofer, P., Rajagopalan, R., Layton, R., Weiss, R., Ghosh, S., Du, S., Jones, T., Dubourg, V., Michel, V., Fritsch, V. and Li, W. (2019). *3.2.4.3.2. sklearn.ensemble.RandomForestRegressor — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html [Accessed 4 Jun. 2019].

**Gandhi, R.** (2019). *Support Vector Machine — Introduction to Machine Learning Algorithms*. [online] Towards Data Science. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 [Accessed 4 Jun. 2019].

**GeeksforGeeks.** (2019). *Confusion Matrix in Machine Learning - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ [Accessed 4 Jun. 2019].

**Huppertz, P. (2019).** Predicting HCAHPS scores from hospitals' social media pages: A sentiment analysis. - PubMed - NCBI. [online] Ncbi.nlm.nih.gov. Available at: https://www.ncbi.nlm.nih.gov/pubmed/28225448 [Accessed 3 Jun. 2019].

**Ke, G.,** Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T. (2019). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. [online] Papers.nips.cc. Available at: https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf [Accessed 4 Jun. 2019].

**Kim, S.** (2019). *Linear Regression*. [online] Towards Data Science. Available at: https://towardsdatascience.com/regression-analysis-linear-regression-239df26a94ac [Accessed 4 Jun. 2019].

**Mahto, K.** (2019). *Demystifying Maths of Gradient Boosting*. [online] Towards Data Science. Available at: https://towardsdatascience.com/demystifying-maths-of-gradient-boosting-bd5715e82b7c [Accessed 4 Jun. 2019].

**Narkhede, S.** (2019). *Understanding AUC - ROC Curve*. [online] Towards Data Science. Available at: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 [Accessed 4 Jun. 2019].

**Pedregosa, F.,** Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2019). *1.11. Ensemble methods — scikit-learn 0.21.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/ensemble.html#forest [Accessed 4 Jun. 2019].

**Pedregosa, F.,** Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2019). *2.3. Clustering — scikit-learn 0.21.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/clustering.html#k-means [Accessed 4 Jun. 2019].

**Pedregosa, F.,** Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2019). *sklearn.linear\_model.LinearRegression — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html [Accessed 4 Jun. 2019].

**Raschka, S.** (2019). *Linear Discriminant Analysis*. [online] Dr. Sebastian Raschka. Available at: https://sebastianraschka.com/Articles/2014\_python\_lda.html [Accessed 5 Jun. 2019].

**Statistics How To.** (2019). *RMSE: Root Mean Square Error - Statistics How To*. [online] Available at: https://www.statisticshowto.datasciencecentral.com/rmse/ [Accessed 4 Jun. 2019].