Predictive Machine Learning Algorithms for 30-day Hospital Readmission in Diabetes Cohort

Sarah Polzer and Alexis Barrett

George Mason University

Abstract

Hospital readmission is preventable, a common measure for quality of care, and harmful to patients. Data scientists worldwide have built algorithms to discover predictive factors for hospital readmission, however many argue that these algorithms have underperformed. In this study, the Diabetes 130-US hospitals for years 1999-2008 Data Set and four open source machine learning classification models were evaluated and compared. A number of methods were used including characterizing models with tree visualizations and feature importance, observing prediction metrics, and calculating statistical performance measures. The interpretation of these models supported previous opinions for hospital readmission predictive algorithms.

1. Introduction and Background

The Centers for Medicare & Medicaid Services (CMS) consider hospital readmission within 30 days preventable and readmission has become a common measure for quality of care. In 2012, CMS changed Medicare payout so that hospitals can profit during a patient's first admission but do not receive any reimbursement for a patient readmission (HRRP, 2019). Similar penalties have been implemented in other countries. At the same time, Meaningful Use required use of electronic health records with a clinical decision support system, or CDSS, in all healthcare facilities (2019).

Since all clinical data is now digitized, the path is clear for data scientists to build predictive algorithms that can be implemented in clinical settings, however, according to Jiang, Chin, Qu, and Tsiu, most computational algorithms have underperformed (2018). Use of underperforming models could put patients at health risk or encourage unnecessary procedures, so models should be optimized for reliability. Understanding the factors that predict hospital readmission is incredibly valuable to physicians and hospital administrators, as readmission is detrimental to patients and costly to hospitals.

In efforts to create a framework for readmission prediction, researchers have designed datasets specifically for predictive analysis of readmission rates. Researchers at Virginia Commonwealth University created a dataset relating to Diabetes patients that was sent to the University of California Irvine Machine Learning repository. The dataset was called the Diabetes 130-US hospitals for years 1999-2008 Data Set (Strack et al., 2014). The dataset has been cleaned and analyzed by numerous data scientists who have attempted to find the best predictor of readmission rates using different means of classification including Decision Trees, XGBoost, and Random Forest. The findings of are present in online publications, journal articles, and in numerous Github repositories.

The purpose of this research is to evaluate the claims of Chin, Qu, and Tsi, that "most computational algorithms have underperformed" (2018). The models will be characterized with visualizations and statistical performance measures using Python to assess whether or not their classifiers are predictive and useful. This assessment can serve as a framework to aid the decision making process of hospital administrators before investing in and implementing new predictive systems.

2. Methods

a. Data

The Diabetes 130-US hospitals for years 1999-2008 Data Set was collected by a team from VCU led by Beata Strack. The dataset consists of records of 100,000 hospital encounters of 40,000 unique diabetes patients. The dataset has over 50 attributes including demographics, diagnoses, lab results, medication, and readmission data. The dataset was built to perform predictive analytics using machine learning algorithms with readmission as a target variable (2014).

Github was searched for open source repositories which had used Strack's dataset to apply machine learning analytics. Two repositories were found that applied Python's popular Scikit-learn machine learning packages. Github user Maximilian Kurscheidt is a Masters Graduate student in Amsterdam (2018). Kursheidt cleaned and analyzed the dataset with Decision Tree and Random Forest classifiers. A second Github user with an open source repository, Shane Wong, is a Masters Candidate at NCI and Tsinghua University in Beijing, China. Wong applied XGBoost and Random Forest classification algorithms after cleaning the dataset (2019). Both researchers applied statistical analysis and limited visualization. The classification tasks were followed by analysis of the models' qualities using confusion matrices. These repositories will be cloned as a basis for further analysis, visualization, and communication of results.

b. Visual and Statistical Analysis

Kurscheidt and Wong's code bases were updated with additional visualization and analytical techniques across the four models. The classifiers built were analyzed in this study using Python. Six Python visualization packages were used: graphviz, PyDotPlus, dtreeviz, matplotlib, and Seaborn's Heatmap function (2019).

Kursheidt's decision tree was visualized with dtreeviz and his random forest with PyDotPlus.Wong's more advanced Random Forest and XGBoost classifiers were not visualized. The most important features for each model were sorted and compared, and Matplotlib was used to visualize these important features in the dataset. Kurscheidt and Wong included confusion matrices in their projects to gauge the quality of their models. Their confusion matrices were visualized using Seaborn's Heatmap function. Kurscheidt's model was fit to produce a two by two heatmap matrix which reflected the binary classification task performed. Seaborn's Heatmap function was used to visualize Wong's three by three confusion matrices because his confusion matrices were three by three (2019).

Kurscheidt and Wong's classifiers, most predictive attributes, and review level findings were visualized to enable the reader to understand Kurscheidt and Wong's methodologies, results, and quality of work. The findings were summarized in an interactive dashboard using Tableau.

3. Analysis

Kurscheidt made a decision tree model that predicted 30-day readmission (2018). The decision tree evaluates each feature and the range of values in the dataset for that feature and performs a loss function on each. It then chooses the split with the minimum loss and proceeds for each level of the decision tree until it reaches the designated maximum level of splits, which is chosen by the programmer to eliminate overfitting.

Dtreeviz is a “fancy tree” package that was developed based on the visualizations created in Stephanie Yee and Tony Chu’s R2D3 project (2019). The tree displays the feature name at each split as well as a histogram of the feature distribution across the two target values: No readmission within 30 days or Readmission within 30 days. Since a decision tree uses a voting system to predict the classification at the last layer, the fancy visualization shows pie charts to display the portions of each class which predict readmission or no readmission. The size of the pie chart represents the support for that classification.

The results of this model, based on analyzing the tree, tells the user that the model could only predict readmission for one of the eight classes produced by the tree, and in this case, the predictor had minimal support across the model and the voting was only slightly above 50%. This suggests that the model did not do a good job of identifying any attributes which unified the group of admissions which led to readmission.

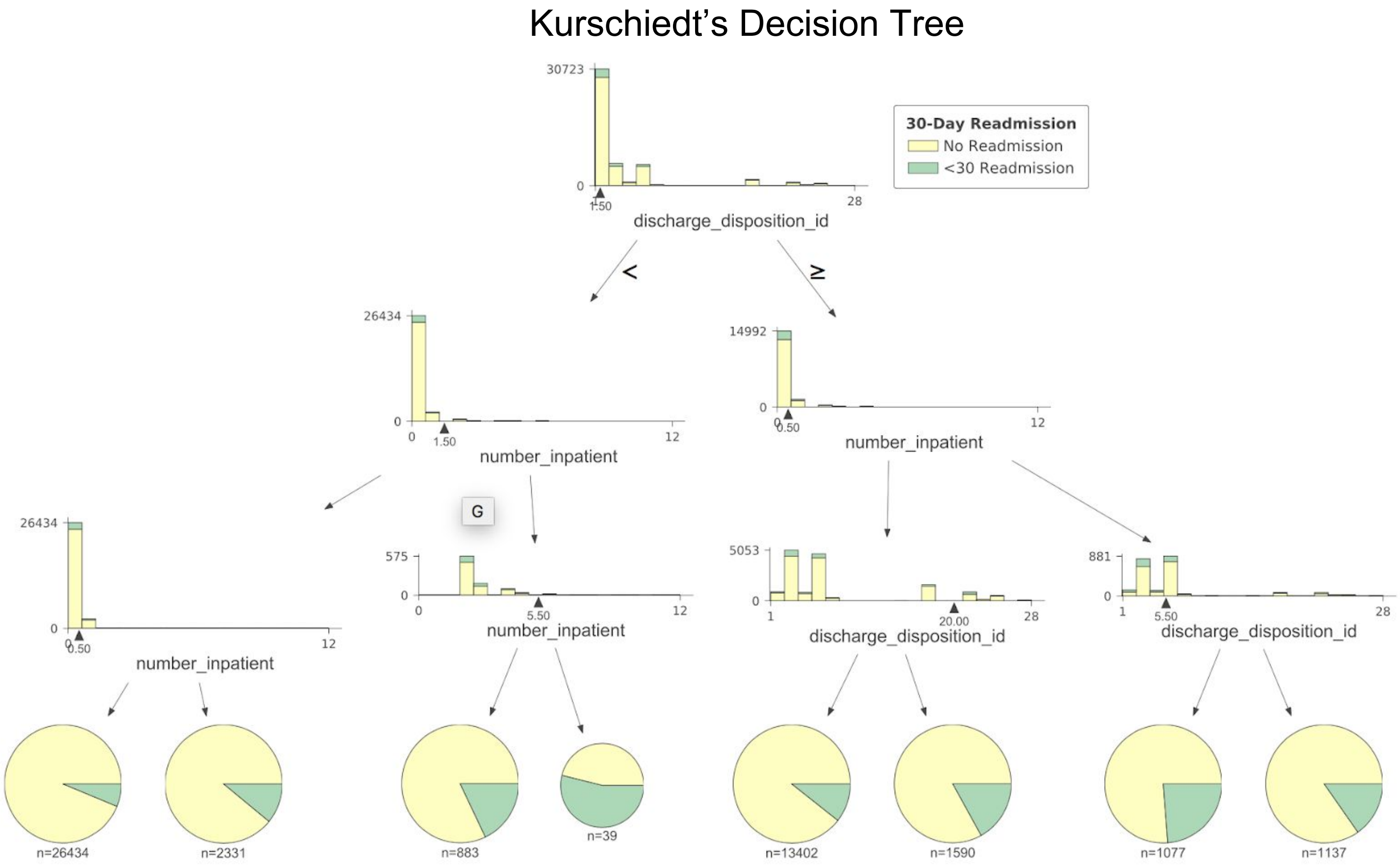

Figure 1: Kurscheidts's decision tree

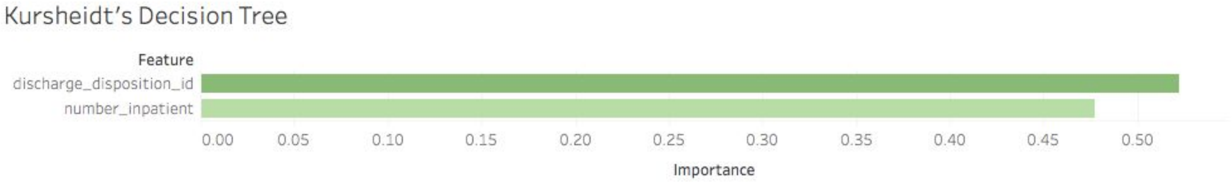

Figure 2: Kurscheidt's Decision Tree Most Important Features

Kurscheidt also made a random forest classifier (2019). A random forest classifier uses multiple decision trees, in this case 30 trees, and takes the majority vote for each split. There is no obvious way to visualize an average of 30 trees, so an estimator function was used prior to visualization. The random forest also used the Gini function as opposed to entropy as a loss function (2018). The random forest visualization is present below.

In analyzing this PyDotPlus tree, it is not nearly as effective as the fancy tree. The features on which the tree is split are displayed only by column number, not on name, which makes it unhelpful to an end user who is not familiar with the dataset. The inclusion of the gini value is only helpful to a data scientist who is familiar with the expected values of the function, again, not an end user. The color scheme of the forest tree is helpful to recognize the classification of each branch, and this classification is backed by the support values shown on the bottom of each box.

In evaluating this model based on the tree, it is found, similarly to Kursheidt's decision tree, that only one branch leads to readmission prediction and this class has the lowest support of any other branch. Again, it is possible to characterize the quality of the model by evaluating the tree but the visual is not self explanatory.

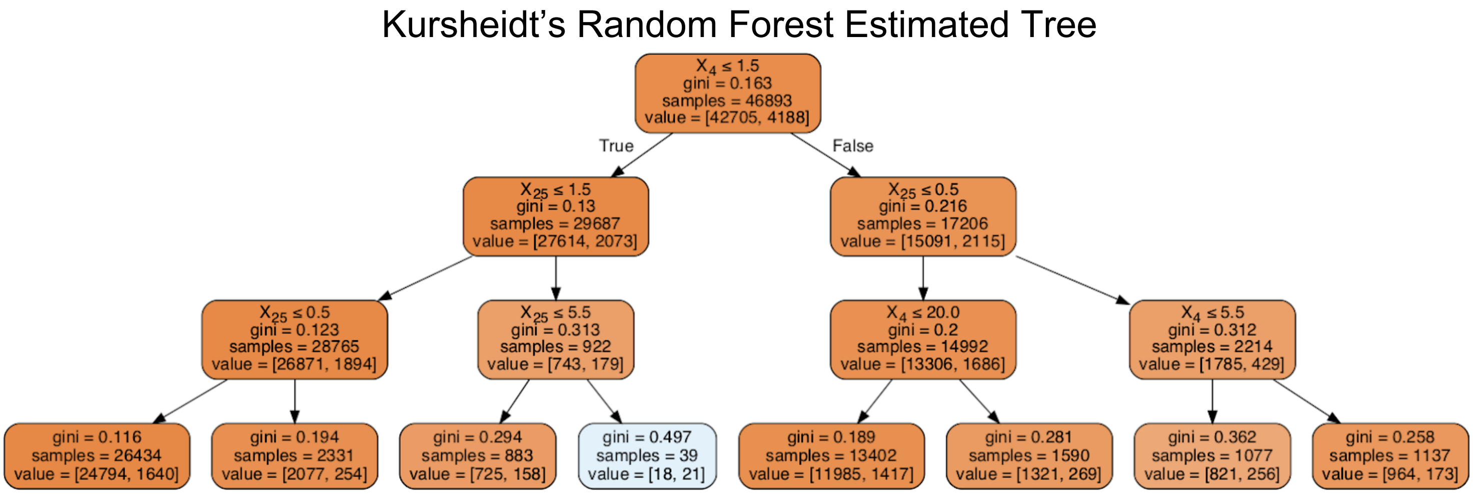

Figure 3: Kurscheidt's random forest

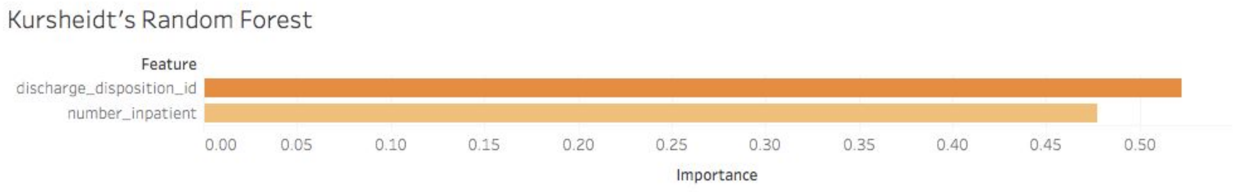

Figure 4: Kurscheidt's Random Forest Most Important Features

Wong determined the most predictive features in the dataset using XGBoost and Random Forest using Python (2019). His most predictive features are present in barchart below. The most predictive features differed based on classifier. These charts would be of value to physicians who can validate the suggested importance for each ranking with their own patient care experience.

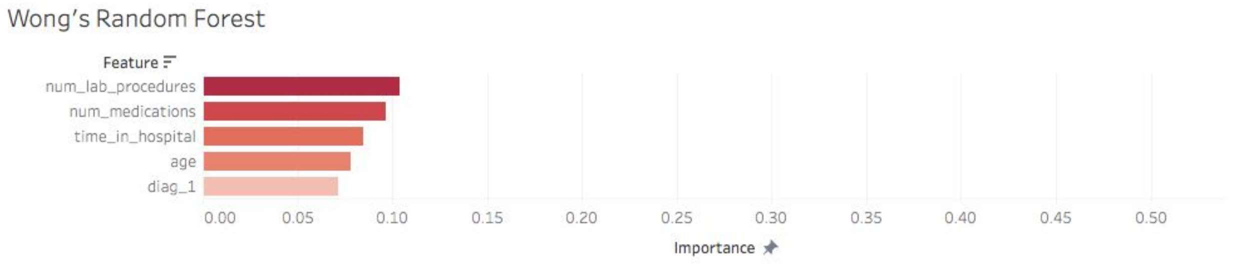

Figure 5: Wong's Random Forest Important Features

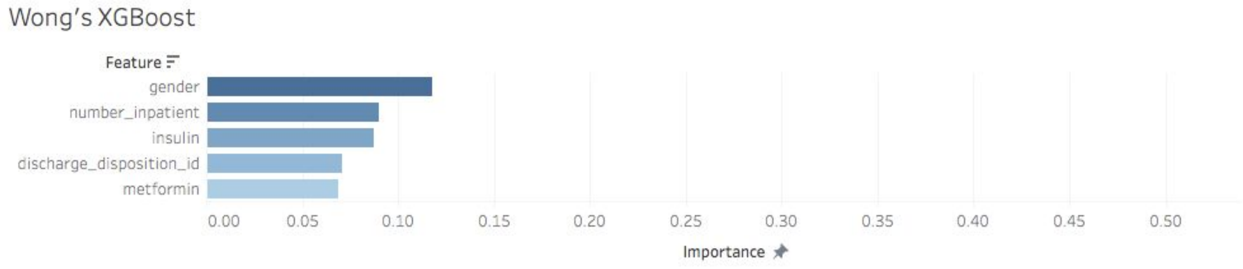

Figure 6: Wong's XGBoost Important Features

Kurscheidt's decision tree and random forest and Wong's random forest and XGBoost model were used to train and test the diabetes dataset. From the results of classifying the testing data, confusion matrices were created. Confusion matrices have an x-axis of actual values and a y-axis of predicted values. The results of testing the data produce values that fall into one of four categories: true positives, false positives, false positives, and false negatives. If a classification model is of good quality, there will be a large number of true positives and a large number of true negatives. If a quality confusion matrix is visualized with a heatmap, the squares should be darker in the diagonal from the upper left to the lower right corner.

Two heatmaps were created to visualize the results of Kurscheidt's decision tree and random forest models. The plots were created with the Seaborn package in Python (2019). Kurscheidt produced models that had confusion matrix heat maps with a dark upper right corner, indicating that his models produced True Positives with high accuracy. However, the light color of the 2nd and 3rd quadrants had signify that his models were flawed. The heatmaps are displayed for Kurscheidt's models in Figures 7 and 8.

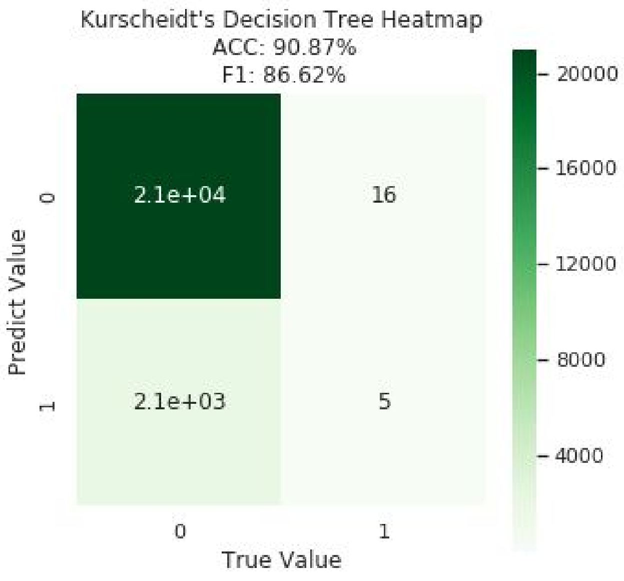

Figure 7: Kurscheidts's Decision Tree's Heatmap

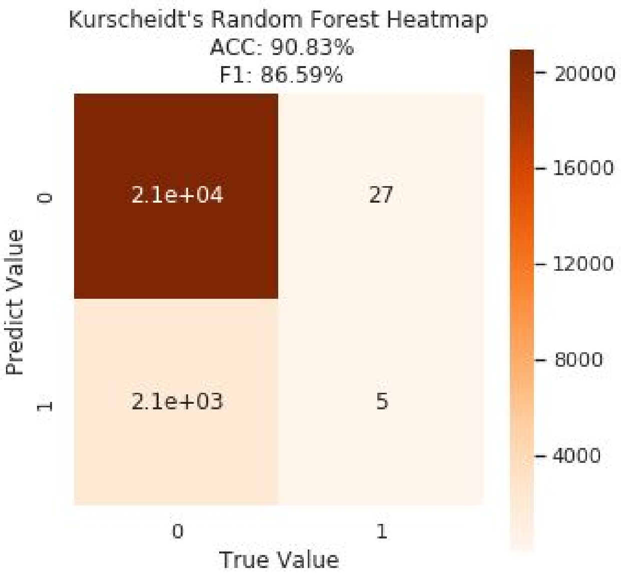

Figure 8: Kurscheidt's Random Forest's Heatmap

Heatmaps were created to visualize the confusion matrices produced by Wong's XGBoost and random forest models. They are present in Figures 9 and 10. In each of the heatmaps, the squares were not distinguishably darker in the diagonal from the upper left corner to the lower right corner. The upper left corners of the Heatmaps were dark, indicating that Wong's models were able to accurately classify whether patients were not readmitted after 30 days. However, the lower right corners of his heatmaps were light, indicating that his models were not able to accurately classify whether patients were readmitted after 30 days. Additionally, the accuracies of the models were labeled on each of the heatmaps, and the accuracy values were just over 50%. Wong's models classified readmission unproportionally and had low accuracy values, indicating his models were not of high quality.

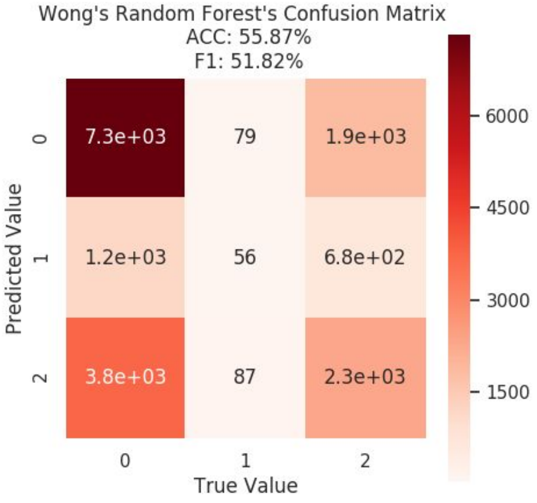

Figure 9: Wong's Random Forest's Heatmap

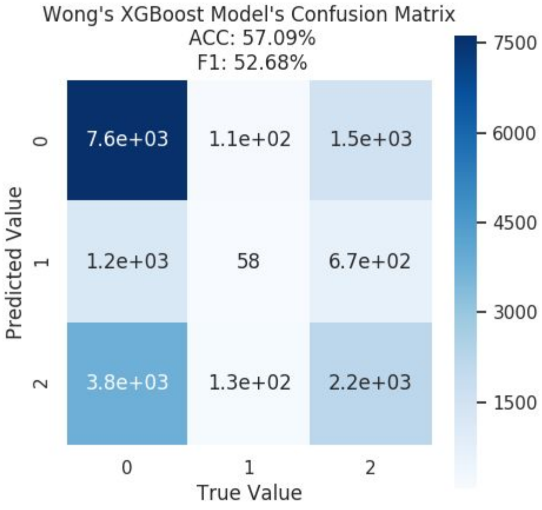

Figure 10: Wong's XGBoost's Heatmap

4. Summary and Conclusion

Hospital Readmission is preventable and can have a devastating impact on patients. Data Scientists have attempted to use hospital readmission data to form classification models to find factors that determine readmission. Two Data Scientists created four models to classify the Diabetes 130-US hospitals for years 1999-2008 Data Set that were found on Github. They used Decision Tree, Random Forest, and XGBoost models. Their models produced confusion matrices that, when visualized, did not look like confusion matrix visualizations created by quality classification models. Their models also had accuracy values that were low. According to Jiang, Chin, Qu, and Tsiu, “most computational algorithms [related to hospital readmission] have underperformed” (2018). The classifications models produced by Kurscheidt and Wong indeed underperformed.

Bibliography

Artetxe, A., Beristain, A., & Graña, M. (2018). Predictive models for hospital readmission risk:A systematic review of methods. *Computer Methods and Programs in Biomedicine*, *164*, 49–64. https://doi.org/10.1016/j.cmpb.2018.06.006

dos Santos, B. S., Steiner, M. T. A., Fenerich, A. T., & Lima, R. H. P. (2019). Data mining and machine learning techniques applied to public health problems: A bibliometric analysis from 2009 to 2018. *Computers & Industrial Engineering*, *138*, 106120. https://doi.org/10.1016/j.cie.2019.106120

Hospital Readmissions Reduction Program (HRRP): CMS. (2019). Retrieved November 16, 2019, from https://www.cms.gov/Medicare/Medicare-Fee-for-Service-Payment/AcuteInpatientPPS/Readmissions-Reduction-Program

Meaningful Use: CDC. (2019, September 10). Retrieved November 16, 2019, from https://www.cdc.gov/ehrmeaningfuluse/introduction.html

Jiang, S., Chin, K.-S., Qu, G., & Tsui, K. L. (2018). An integrated machine learning framework for hospital readmission prediction. *Knowledge-Based Systems*, *146*, 73–90. https://doi.org/10.1016/j.knosys.2018.01.027

parrt/dtreeviz: A python library for decision tree visualization and model interpretation. (2019). Retrieved November 16, 2019, from https://github.com/parrt/dtreeviz

Sarikaya, A., Correll, M., Bartram, L., Tory, M., & Fisher, D. (2019). What Do We Talk About When We Talk About Dashboards? *IEEE Transactions on Visualization and Computer Graphics*, *25*(1), 682–692. https://doi.org/10.1109/TVCG.2018.2864903

seaborn.heatmap—Seaborn 0.9.0 documentation. (2019). Retrieved November 16, 2019, from http://seaborn.pydata.org/generated/seaborn.heatmap.html

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. (2014). Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records [Research article]. https://doi.org/10.1155/2014/781670

Yee, S. & Chu, T. A visual introduction to machine learning. (n.d.). Retrieved November 16, 2019, from http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Data Sources

Kurscheidt, M. (2018). *MadMax93/IPHIE-2018-decision-tree* [Jupyter Notebook]. Retrieved from https://github.com/MadMax93/IPHIE-2018-decision-tree (Original work published 2018)

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. UCI Machine Learning Repository: Diabetes 130-US hospitals for years 1999-2008 Data Set. (2014). Retrieved November 16, 2019, from https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008#

Wong, S. (2019). *Freesinger/readmission_prediction* [Python]. Retrieved from https://github.com/freesinger/readmission_prediction (Original work published 2018)