Predicting Readmittance for Hospitalized Diabetic Patients

Brian Cullinan

Machine Learning Course

3/11/2022

[Executive Summary 2](#_Toc97786114)

[Introduction & Problem Statement 2](#_Toc97786115)

[Summary of Findings 2](#_Toc97786116)

[Data & Modeling Approach 3](#_Toc97786117)

[Data Overview 3](#_Toc97786118)

[Modeling Approach 5](#_Toc97786119)

[Detailed Findings 6](#_Toc97786120)

[Recap of Preliminary Results Report: 6](#_Toc97786121)

[Steps Taken Since of Preliminary Results Report: 9](#_Toc97786122)

[Evaluation 16](#_Toc97786123)

[Selection Model Results: 16](#_Toc97786124)

[Recommendations for Future Implementations: 19](#_Toc97786125)

[Appendix 20](#_Toc97786126)

[Source Code: 20](#_Toc97786127)

[Citations: 20](#_Toc97786128)

# 

# Executive Summary

## Introduction & Problem Statement

Hospital patients who are readmitted after being discharged generally have poor medical outcomes with increased costs. (McIlvennan). Predicting which patients are most likely to be readmitted may improve outcomes and reduce costs, by flagging those patients as at-risk and providing them with additional care.

This project looked at diabetic patient encounters from 1999-2008 at 130 hospitals with data on whether or not the patient was readmitted after being discharged. The dataset contains 2 identifiers and 48 other variables describing encounters with diabetic patients, including patient demographics, admission data, various tests and treatments during a patient’s stay, the length of a patient’s stay in the hospital, diagnoses, and medications prescribed for treatment during their stay.

Using machine-learning classification models, this project aimed to answer the research question: *“How well can the data on diabetic patients collected by 130 hospitals predict which patients are likely to be readmitted?”*

## Summary of Findings

The diabetic patient data from 1999-2008 was modeled with a Voting Classifier model. The Voting Classifier model utilizes four different models (Random Forest, Gradient Boosting, GaussianNB, and Neural Network) and then makes a classification prediction through soft voting from the results of each of the models. Further discussion of how this model was chosen is available in the “Detailed Results” section.

A primary objective of this project was to minimize false negative predictions, where a patient was not predicted to be readmitted but actually was readmitted. A false negative is considered to be the worst possible outcome in a medical setting because if a patient is not flagged as being at-risk, then they may not receive the care they need. By flagging at-risk patients and minimizing false negatives, patient outcomes can be improved. An outcome in which a patient is classified as at-risk for being readmitted when they actually are not (false positive) is less adversely impactful to the patient since the downside is additional care administered to avoid a poor outcome as opposed to a missed opportunity to provide life-saving care.

Through validation with a test dataset, the Voting Classifier model was shown to be 61% accurate in predicting if a patient would be readmitted. ([See Figure 11](#_gtqwnl67h25j)). Further, the model is better at avoiding false negatives than false positives: when a patient is predicted to not be readmitted, that prediction is correct slightly more than 70% of the time; when a patient is predicted to be readmitted, that prediction is accurate slightly more than 50% of the time.

While the accuracy is low when predicting that a patient will be readmitted, this is less of an adverse impact as compared to a false positive because it means they will be flagged as at risk and given the care needed to improve their prognosis. More detail on the findings can be found in the “Evaluation” section.

While the prediction results are mediocre, it is hoped that this model may begin to provide valuable information to the hospital and physicians providing care to diabetic patients. Recommendations for future improvements are included at the end of this document.

# Data & Modeling Approach

## Data Overview

This analysis included data clean-up and feature engineering in order to utilize and optimize classification models. The following steps were applied to the source data prior to modeling:

1. Null value treatment:

### *Table 1: Overview of Null Values with Handling Notes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable | # Of Nulls | Null % | # Of Unique Values | Action |
| weight | 68665 | 96.0 | 9 | Variable Removed |
| medical\_specialty | 34477 | 48.2 | 70 | Variable Removed |
| payer\_code | 31043 | 43.4 | 17 | Variable Removed |
| race | 1948 | 2.7 | 5 | Nulls filled with “Unknown” |
| diag\_3 | 1225 | 1.7 | 758 | Allowed to remain null |
| diag\_2 | 294 | 0.4 | 725 | If diag\_2 is null and diag\_3 is not null, replace diag\_2 with diag\_3 and delete value from diag\_3. |
| diag\_1 | 11 | 0.01 | 696 | If diag\_1 is null and diag\_2 is not null, replace diag\_1 with diag\_2 and delete value from diag\_2. Delete any values that are still null. |

1. Other observations and variables were removed from the dataset, including:
   1. 1,545 patients known to be deceased from the discharge type
   2. 3 patients who did not have at least one diagnosis
   3. 3 variables which were not relevant to predicting readmission were dropped: “admission\_source\_id”, “medical\_specialty”, “payer\_code”
2. The following encoding strategies were used:

### *Table 2: Encoding Strategy*

|  |  |  |
| --- | --- | --- |
| Encoder | Variable List | Notes/Details |
| Label Encoded | “Gender”, “Change”, and “DiabetesMedicine”, “Readmitted” | “Readmitted” is the response variable and was encoded to indicate if a patient is readmitted (1) or not (0). The detail of when a patient was readmitted was dropped. |
| Dummy Encoded | “Race”, “Discharge”, “Admission” | To reduce the number of features, both Discharge and Admission were mapped to major groups (i.e., “Discharge\_Home”, or “Discharge\_OutPatient”) |
| Ordinal Encoded | “A1Cresult”, “max\_glu\_serum”, and all medication features | All these features have an order to represent the dosage level or severity of a test result. |

1. Four engineered features will be added to the dataset:

### *Table 3: Feature Engineering Overview*

|  |  |  |
| --- | --- | --- |
| Feature Name | Description | Source Data/Logic |
| “num\_of\_meds” | Sum of the number of medications prescribed | All medication variables |
| “med\_level\_direction” | Indicates whether the net medication level went up or down and by how much | All medication variables |
| “num\_meds\_changed” | The number of medications where dosage was changed | All medication variables |
| “patientVisits” | The sum of visits the patient made to medical facilities in the last year | The sum of visits captured in “number\_emergency”, “number\_inpatient”, “number\_outpatient”. Those variables were dropped and replaced with “patientVisits” |

## Modeling Approach

The research question of this project is a classification question of whether or not patients are predicted to be readmitted to the hospital. To answer this question, five classification models in the Scikit-Learn python library that represent a variety of classification approaches were selected as candidates. These candidates and the reason they were selected are as follows:

1. *Random Forest*: This is an ensemble method of many learners, that is generally a good classification model which provides clear insights into feature importance. Additionally, Random Forest is fairly easy to interpret, therefore helps to maintain the principle of parsimony.
2. *Gradient Boosting*: Boosting is another ensemble method and is therefore in some ways a similar model to Random Forest. It is an attractive option for its ability to combine weak learners into strong learners, and because it also provides clear insights into feature importance.
3. *MLP Classifier (referred to hereafter as “Neural Network”)*: Neural Networks (NN) take a propagation approach that is different from the ensemble methods listed above, which is the first reason for comparison against the ensemble methods above. Additionally, “[Neural Networks (NN’s)] can generalize - After learning from the initial inputs and their relationships, it can infer unseen relationships on unseen data as well, thus making the model generalize and predict on unseen data.” (Mahanta).
4. *K-Nearest Neighbors (KNN):* KNN was selected because it is a simple and intuitive model that offers a good comparison for the rest of the models by taking a very different approach of making predictions based on the closest neighbors. While this model does not generally perform well with large feature sets, and therefore did not have strong potential, it did offer a good comparison for the rest of the models.
5. *Gaussian Naive Bayes (NB)*: While Random Forest, Gradient Boosting, and Neural Networks wereselected in part because of their ability to identify hidden patterns and combinations within the data, GaussianNB was selected as a comparison for the opposite reason in that Naive Bayes model assumes no relationships (independence) between features and therefore, offers a good comparison to the other models.

The modeling approach was to evaluate these models in a multi-step way to narrow down to a best-performing selected model. Later in this model evaluation, as will be discussed in the “Detailed Findings” section, a voting ensemble composed of multiple models was also added as a candidate.

The overall model performances were trained with 80% of the overall records, and then tested using a test data set made up of a random sample of 20% of the overall records. The overall model performance was measured with accuracy scores and Receiver Operating Characteristic (ROC) curves with the Area Under the Curve (AUC) metric. Since false negatives are the outcome that the models tried to minimize, more detailed metrics of model performance in the form of precision, recall, and F1 scores were used. Of these metrics, recall was particularly important as the primary metric to show false negatives. Finally, in the later stages of model evaluation, where a very detailed comparison is called for, confusion matrices were also utilized.

# Detailed Findings

## Recap of Preliminary Results Report:

As discussed in the “Data & Approach” section, to evaluate models a narrowing-down approach was followed where, first, many models as well as feature and data engineering options were considered.

The first step was to choose 5 different classification models to run an initial “quick evaluation” with minimal tuning. The goal of this step was to identify models that are promising for further tuning and improvement. This step also compared the models with and without class balancing of the response variable (“readmitted”) and determined that class balancing should be used going forward. Models in this step were measured with precision, recall, and F1 scores, however since false negatives are the outcome we are trying to minimize, recall is used as the primary metric. Although none of the models performed particularly well, one (KNN) performed poorly enough that it was eliminated from consideration.

### *Table 4: Initial Model Evaluation*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Recall** | **Precision** | **F1** | **Actions/Next Steps** |
| GaussianNB | 0.5649 | 0.5013 | 0.5312 | Evaluate Further |
| RandomForest | 0.5599 | 0.5012 | 0.5289 | Evaluate Further |
| Neural Network | 0.4701 | 0.5336 | 0.4999 | Evaluate Further |
| GradientBoosting | 0.4344 | 0.5529 | 0.4865 | Evaluate Further |
| KNN | 0.2583 | 0.4843 | 0.3369 | Eliminated KNN Model |

For the second step, the remaining models were further evaluated to see if they would benefit from dimensionality reduction of the continuous numeric features using Principal Component Analysis (PCA). The features suited for PCA are: “age” (binned in 10-year increments), “time\_in\_hospital”, “num\_lab\_procedures”, “num\_medications”, “number\_outpatient”, “number\_emergency”, “number\_inpatient”, “number\_diagnoses”, “num\_procedures”, “num\_of\_meds.” The advantage of using PCA is that the models may potentially work with just as much information in a smaller feature set, which is particularly useful for ensemble methods like Random Forest. Using an elbow plot, it was determined that the optimal number of principal components would be 2.

The variables noted above were reduced to two principal components, and the original variables in the dataset were replaced in the dataset. The results show most models perform similarly, but the GaussianNB greatly improves the recall score with PCA. Based on the models generally performing better with PCAs, it was initially determined that the on-going evaluation would utilize PCA. However, as will be discussed later, this decision was later reversed, as it was subsequently discovered that the increase in model performance was less attributable to PCA itself, and more so a result of reducing the overall number of features (detail to follow).

### *Table 5: PCA Evaluation*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **PCA** | **Recall** | **Precision** | **F1** | **Actions/Next Steps** |
| GaussianNB | Yes | 0.7347 | 0.4625 | 0.5677 |  |
| GaussianNB | No | 0.5649 | 0.5013 | 0.5312 | Eliminate Option w/o PCA |
| RandomForest | No | 0.5599 | 0.5012 | 0.5289 | Eliminate Option w/o PCA |
| RandomForest | Yes | 0.5578 | 0.4991 | 0.5268 |  |
| Neural Network | Yes | 0.4722 | 0.5201 | 0.495 |  |
| Neural Network | No | 0.4701 | 0.5336 | 0.4999 | Eliminate Option w/o PCA |
| GradientBoosting | Yes | 0.4494 | 0.5466 | 0.4933 |  |
| GradientBoosting | No | 0.4344 | 0.5529 | 0.4865 | Eliminate Option w/o PCA |

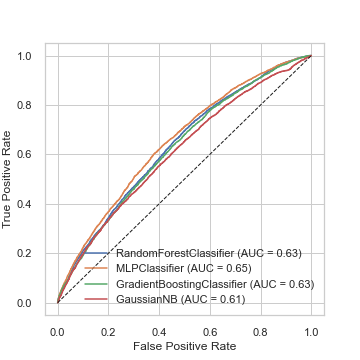
*\*Note: The determination to use PCA was later reversed.*

As a third and final step, the hyperparameters of the remaining models were further tuned, and more detailed information on the performance of each model was collected. At this stage, in addition to using recall, precision, and F1 scores (with particular attention paid to recall scores) the models were also compared using Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) metric. While the goal was to reduce the number of remaining models, no model substantially outperformed the other models, and therefore no models were eliminated. The results showed that the different models all had strengths and weaknesses (as shown by recall and precision scores), but overall model performance (as shown by F1 and AUC scores) were very similar.

### *Table 6: Results presented in the “Preliminary Results Report”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Recall** | **Precision** | **F1** | **AUC** |
| GaussianNB | 0.7445 | 0.4612 | 0.5696 | 0.61 |
| Gradient Boost | 0.6402 | 0.5633 | 0.5993 | 0.63 |
| Random Forest | 0.5841 | 0.5921 | 0.5881 | 0.63 |
| Neural Network (MLP Classifier) | 0.5451 | 0.6107 | 0.576 | 0.65 |

### *Figure 1: ROC Curve/AUC metrics presented in the “Preliminary Results Report”*



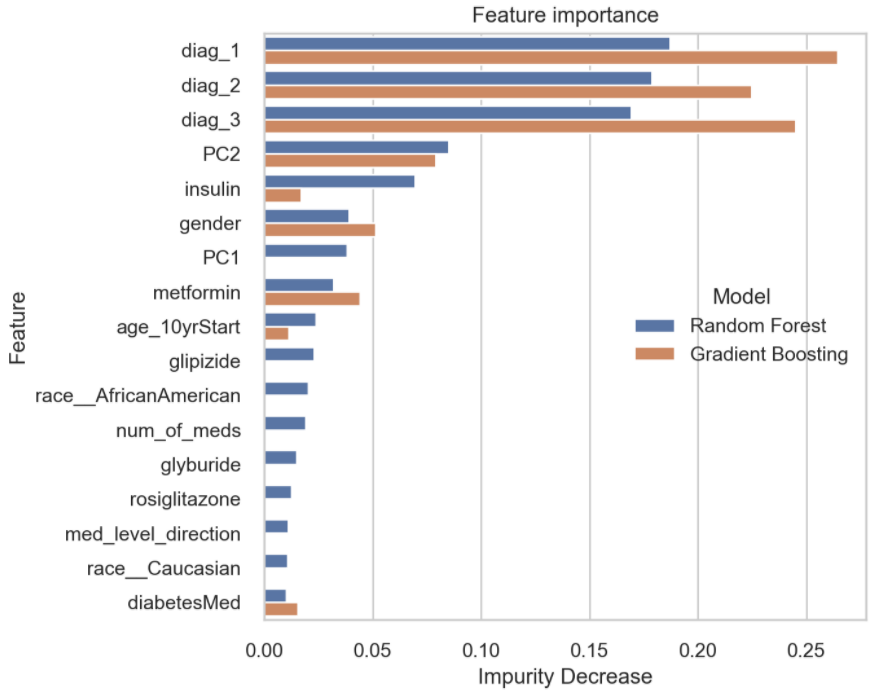
## Steps Taken Since of Preliminary Results Report:

The “Preliminary Results Report” included next steps for further evaluation improvement. The steps outlined in that report were:

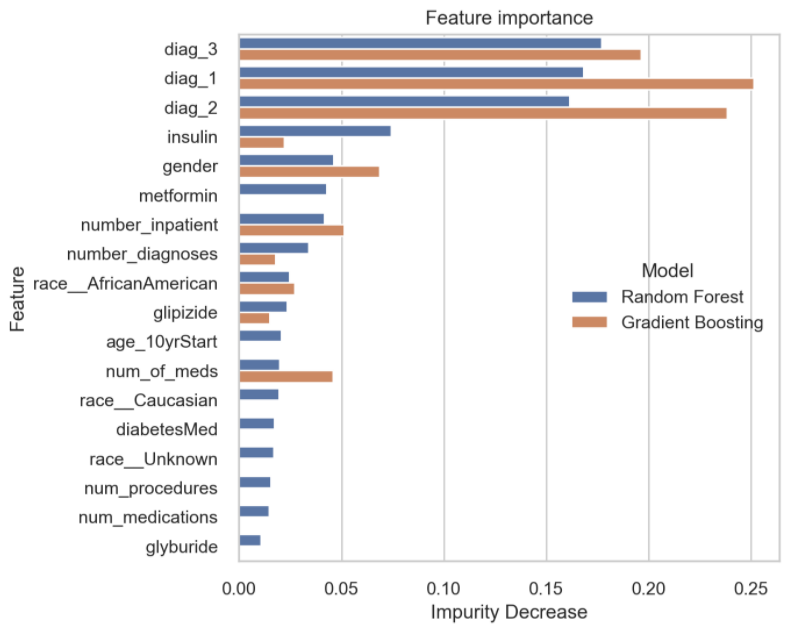
* Feature Engineering Steps:
  + **Change encoding of diagnosis features:** change from a target encoder to a custom approach where diagnosis codes are grouped by major categories of ICD-9 diagnosis codes and then dummy encoded.
  + **Clustering**: Use unsupervised clustering to generate a cluster ID as a feature to show patient profiles.
* Potential Model Improvements:
  + **Reduce Feature Set:** Use the Random Forest and Gradient Boosting feature importance function to reduce the set of predictors to the most important features.
  + **Tweak the prediction threshold:** Experiment with changing the default 50% prediction threshold in order to minimize false negatives.
  + **Ensemble Methods:** Should no single model win out, try to improve prediction accuracy with an ensemble method composed of multiple models.

After the “Preliminary Results Report” was submitted, the first step was to re-evaluate the use of PCA by comparing the results to a reduced feature set. The top features were determined by the Random Forest and Gradient Boost feature importance functions (see plots below), and features that had a Gini impurity decrease of at least 0.01 in either model were selected. The rest were removed as predictors.

### *Figure 2: Feature Importance for PCA Model*



*Figure 3: Feature Importance for non-PCA Mode*l



The result showed that the benefits from the option with PCA were minimized when the overall feature set was reduced. However, given that the model performance was very close with and without PCA, this assertion was frequently re-tested in subsequent evaluation steps, however the option using PCA never outperformed the option without it.

### *Table 7: Model Comparison Using Reduced Feature Set and PCA and Reduced Feature Set*

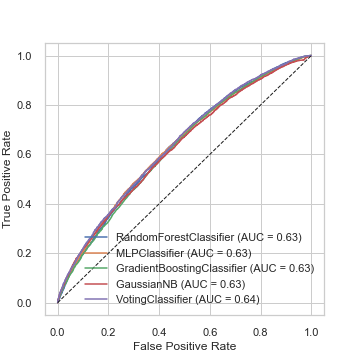
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Group | Accuracy | Precision | Recall | F1 | AUC |
| Neural Network | Reduced Feature Set | 0.617 | 0.5911 | 0.1935 | 0.2915 | 0.63 |
| Random Forest | Reduced Feature Set | 0.5932 | 0.5006 | 0.552 | 0.525 | 0.63 |
| Random Forest | PCA + Reduced Feature Set | 0.5921 | 0.4993 | 0.5499 | 0.5234 | 0.63 |
| Gradient Boost | PCA + Reduced Feature Set | 0.5911 | 0.4983 | 0.5956 | 0.5426 | 0.63 |
| Gradient Boost | Reduced Feature Set | 0.5869 | 0.4939 | 0.5822 | 0.5344 | 0.63 |
| GaussianNB | Reduced Feature Set | 0.5756 | 0.4837 | 0.6277 | 0.5464 | 0.63 |
| Neural Network | PCA + Reduced Feature Set | 0.5646 | 0.4777 | 0.7399 | 0.5806 | 0.64 |
| GaussianNB | PCA + Reduced Feature Set | 0.5617 | 0.4737 | 0.6854 | 0.5602 | 0.62 |

The next step was to add a voting classifier ensemble method for consideration. Since no one model was clearly winning out, the Voting Classifier model would determine whether predictions could be improved by tying the models together in a voting ensemble. The initial results were promising in that it appeared to have potential to combine the best parts of the various models, namely a way to retain overall performance (by AUC score) while also improving recall score. Given this promising outlook, all subsequent evaluations considered the Voting Ensemble as a candidate.

### *Table 8: Model Comparison with Voting Classifier Ensemble*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 | AUC |
| Voting Ensemble | 0.5865 | 0.4938 | 0.6097 | 0.5457 | 0.64 |
| Neural Network | 0.617 | 0.5911 | 0.1935 | 0.2915 | 0.63 |
| Random Forest | 0.5932 | 0.5006 | 0.552 | 0.525 | 0.63 |
| Gradient Boost | 0.5869 | 0.4939 | 0.5822 | 0.5344 | 0.63 |
| GaussianNB | 0.5756 | 0.4837 | 0.6277 | 0.5464 | 0.63 |

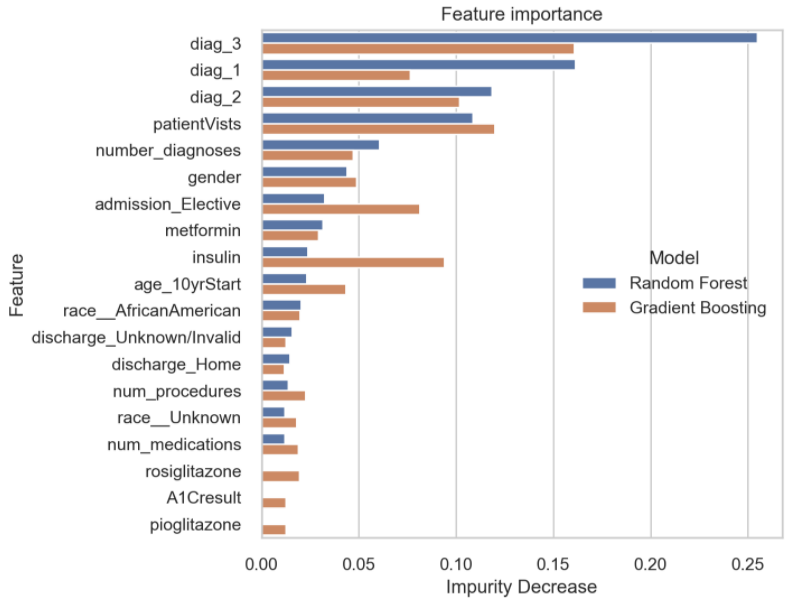
### *Figure 4: ROC Curve/AUC Metrics with Voting Classifier Ensemble*



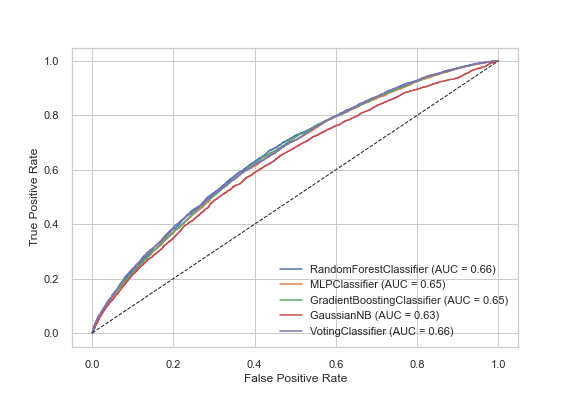
The next step was to revisit the feature engineering approach and try new approaches with three different features. First, “admission\_type\_id” and “discharge\_disposition\_id” were each mapped to a few major groups, such as “emergency”, “elective”, “birth”, and “unknown/invalid” for admission and “Home”, “Other Care Facility”, and “To Outpatient” for discharge types. Then both “admission” and “discharge” were encoded using dummy encoding.

The results showed that these features were selected as important using the Random Forest and Gradient Boost feature importance functions (see “Feature Importance w/ Admission/Discharge”) and had slightly improved AUC scores (see “ROC/AUC w/Admission/Discharge”). Given this, these changes were implemented going forward.

### *Figure 5: Feature Importance w/ Admission/Discharge Feature Engineering:*

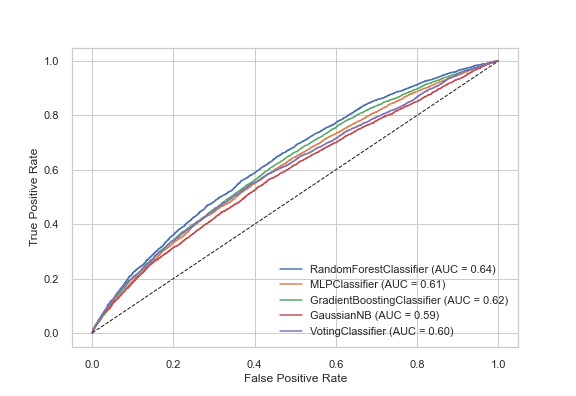


### *Figure 6: ROC Curve/AUC Metrics w/ Admission/Discharge Feature Engineering:*



Next, a similar approach was taken with diagnosis, a primary difference being that there are more categorical values to choose from for diagnosis, and that a standard for major groups exists within the ICD-9 codes, which was utilized (Wikipedia). Once diagnoses were aligned to major ICD-9 groups, it was also encoded using dummy variables. Unlike “admission” and “discharge”, this led to a decrease in AUC scores, so this change was not implemented.

### *Figure 7: ROC Curve/AUC Metrics w/ Admission/Discharge and Diagnosis Feature Engineering*



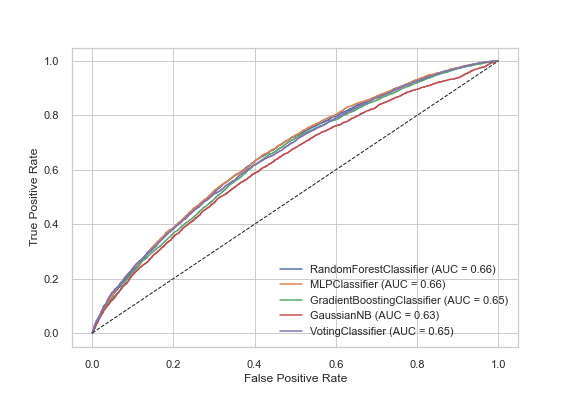
Using the same selected features, and the same model parameters, extensive experimentation was conducted with the prediction parameters. This experimentation altered the default 50% threshold used in predicting whether or not a patient would be predicted to be readmitted. The idea behind this experiment was to avoid false negatives by increasing the threshold for predicting that a patient would not be readmitted. After extensive experimentation, this was abandoned since the default parameters were not improved upon.

The final step before selecting a final model was to fine-tune the model parameters. To do this, Scikit-Learn’s GridSearchCV function was utilized. This function allows many options of multiple parameters to be specified and compares all combinations of those parameters using cross-validation. Parameters may be compared to improve accuracy, precision, or recall scores. Once completed, the function returns a set of the best parameters to be used for increasing whichever score that was specified. The RandomForest, Gradient Boost, and Neural Network models were fine-tuned using this function (there are no tuning parameters for the GaussianNB model), and running the model with best parameters for recall produced the following results:

### *Table 9: Model Performance After Hyperparameter Tuning with GridSearchCV*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 | AUC |
| Random Forest | 0.6229 | 0.5291 | 0.5602 | 0.5442 | 0.66 |
| Gradient Boost | 0.6051 | 0.507 | 0.6283 | 0.5612 | 0.65 |
| GaussianNB | 0.5669 | 0.4739 | 0.7038 | 0.5664 | 0.63 |
| Neural Network | 0.6274 | 0.5368 | 0.5335 | 0.5351 | 0.66 |
| Voting Ensemble | 0.5981 | 0.5 | 0.6466 | 0.5639 | 0.65 |

### *Figure 8: ROC Curve/AUC Metrics After Hyperparameter Tuning with GridSearchCV*



# Evaluation

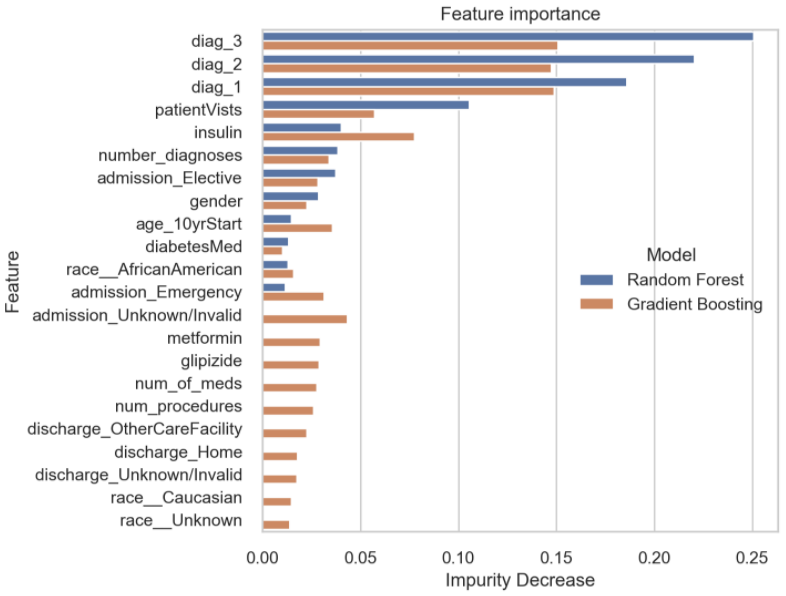
## Selection Model Results:

As described in the “Detailed Results” section, there were many options for features, models, and model configuration that were explored. Some of the early refinement steps, such as using a reduced set of important features, led to meaningful improvements. However, refinement gains quickly diminished and provided only minor improvements to the overall performance. With options for overall improvement largely exhausted, attention was turned into balancing overall model performance with reducing false negatives.

The final model chosen was the ensemble Voting Classification which is composed of all the remaining models: Random Forest, Neural Network, Gradient Boosting, and GaussianNB. The advantage of this model over any used individually was that it was able to leverage the benefits of the individual models. For example, GaussianNB had the best recall score (an indication that false negatives were reduced) but was otherwise the poorest performing overall model. The Voting Classification model allowed GaussianNB to be used with other models that had better overall performance but lower recall scores.

This final ensemble model utilized the features that showed at least a 0.01 Gini impurity decrease from either the Random Forest or Gradient Boost model.

### *Figure 9: Feature Importance Used with Final Model*

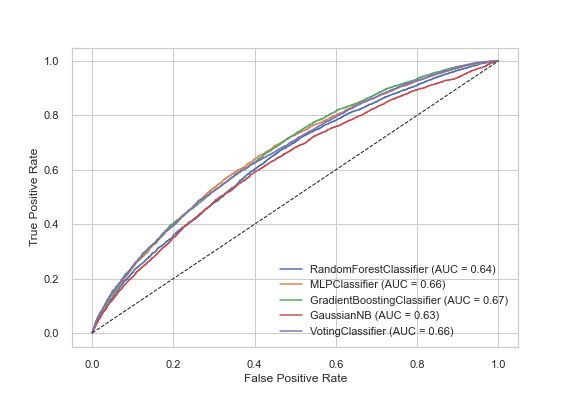


It is important to note that this final model selection used a well-rounded picture of how the model performed as measured by several metrics, most important of which were recall, accuracy, and AUC. It was not the best configured model for any one measurement, but instead the best when considering all of them. The result was a more balanced model that was not quite as good as GaussianNB in terms of recall call but had significantly better overall accuracy.

### *Table 10: Final Model Performance*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 | AUC |
| GaussianNB | 0.5671 | 0.4737 | 0.6939 | 0.563 | 0.63 |
| Random Forest (Tuned for Recall) | 0.5961 | 0.4981 | 0.6327 | 0.5574 | 0.64 |
| **Selected Model: Voting Classifier** | **0.6121** | **0.5144** | **0.6208** | **0.5626** | **0.66** |
| Neural Network | 0.6352 | 0.5505 | 0.504 | 0.5262 | 0.66 |
| Gradient Boost (Tuned for Accuracy) | 0.634 | 0.5541 | 0.4581 | 0.5016 | 0.67 |

### *Figure 10: ROC Curve/AUC Metrics for Final Model*

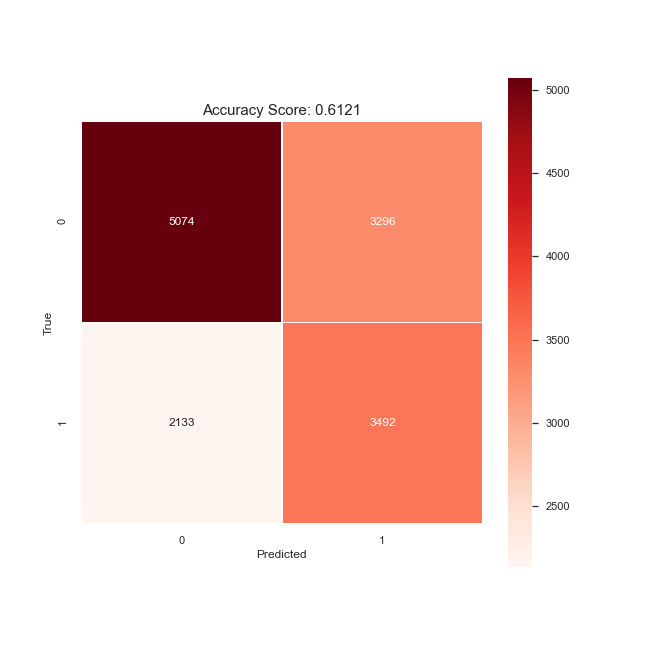


The overall accuracy of the selected model was only just above 61%. However, as has been noted, a heavier importance was placed on reducing false negatives where a patient would be predicted to not be readmitted but were. The model is better at avoiding this case.

Per the confusion matrix below, of the 6,788 patients who were predicted to be readmitted, only 3,492 were, meaning the model is only about 51% accurate when predicting patients will be readmitted. The other 49% would be flagged as at-risk, but not readmitted. Therefore, while this may drive up treatment costs it should not otherwise adversely affect patient outcomes.

In contrast, of the 7,207 patients who were predicted to be readmitted, this prediction proved to be true for 5,074 patients. Meaning that the model is accurate about 71% of the time.

### *Figure 11: Confusion Matrix for Final Model (Voting Classifier Ensemble)*



The results of this project can best be described as mixed. It was hoped that better results would be achieved than the noted 61% overall accuracy with 70% accuracy when predicting not readmitted. Despite optimistic expectations not being met, this model still should be helpful in flagging at-risk patients and as a useful tool in part of a larger evaluation on whether a patient can safely be discharged with minimal risk of readmission.

## Recommendations for Future Implementations:

For the model to truly be useful to practitioners, the overall accuracy needs to be increased. The first priority to improve the overall predictions should be to improve the 70% accuracy for patients who are predicted to not be readmitted. The second priority should be to improve the accuracy of patients predicted to be readmitted. This project was not able to identify concrete steps to do so, but some suggestions are as follows:

1. Consistently collect “weight” data: It is reasonable to hypothesize that weight may be a valuable variable to predict readmittance of diabetic patients, but that variable needed to be dropped due to a high number of null values.
2. Improve data management of diagnosis variables: The three diagnosis features were shown to be the most important features. Given that diagnoses were stored as categorical features with a large number of possibilities, it is quite possible that the true value of that data was never fully realized. Standardizing mapping to major diagnosis may be one way to improve the value delivery of these features.
3. Capture severity: The diagnosis variables capture a diagnosis, but do not contain information of severity. For example, is someone experiencing acute, chronic, or severe symptoms? This may be a valuable predictor.
4. Capture more information on medications: Two suggestions for improving the predictive power of the 23 medicine variables are 1) categorize the medications (i.e., “blood thinners”, “anti-viral”, etc.), and 2) capture dosage data.

# Appendix

## Source Code:

<https://github.com/bcullinan32/ML_Course>

## Citations:

Mahanta, J. (2017, July 10). *Introduction to Neural Networks, Advantages and Applications*. Towards Data Science. <https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207>

McIlvennan, C.K. DNP, Eapen, Z.J. MD, Allen, L.A. MD, “*Hospital Readmissions Reduction Program.”* Circulation. vol. 131, no. 20, 19 May. 2015, pp. 1796-1803. [PubMed.](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4439931/#:~:text=Hospital%20readmissions%20are%20associated%20with%20unfavorable%20patient%20outcomes%20and%20high%20financial%20costs.&text=Causes%20of%20readmissions%20are%20multi,rates%20vary%20substantially%20by%20institution.&text=Historically%2C%20nearly%2020%25%20of%20all,a%20readmission%20within%2030%20days.) [10.1161/CIRCULATIONAHA.114.010270](https://dx.doi.org/10.1161%2FCIRCULATIONAHA.114.010270). Accessed 4 March 2022.

Wikipedia. 2021. *List of ICD-9 codes. 10/12/2021.* <https://en.wikipedia.org/wiki/List_of_ICD-9_codes>