IDS 575 – FINAL REPORT

DIABETES PATIENT READMISSION

Omar Elhayboubi, Shashank Siddagangaiah, Lina Quiceno

Abstract

This paper aims to build a classification model that will predict whether Diabetic patients will be readmitted within 30 days. The Hospital Readmission Program (HRRP) reduces hospital fundings if diabetic patients readmission in less than 30 days rate above average. The ability to intervene early when a patient is at risk of readmission is crucial for hospital management. We can use machine learning models to predict and prioritize these patients. We will develop and evaluate four models in this paper: Naïve Bayes, Logistic Regression, Knn, and SVM. Results will then be compared and analyzed. Finally, the best-performing model will be selected.

Introduction and Motivation

Diabetes is a disease characterized by high sugar levels, which can cause vision problems, lower-limb amputations, heart and kidney disease, or even death.

Diabetes is one of the top ten leading causes of death globally and one of the ones with the highest rates of readmission in less than 30 days to the hospital. Under the "Hospital Readmissions Reduction Program (HRRP)," hospitals must make sure to reduce avoidable readmission in this period of time, or they risk getting reduced the payments by the government for excess readmission. (U.S. Centers for Medicare & Medicaid Services, 2021)

"In recent years, government organizations and healthcare institutions have placed a greater emphasis on 30-day readmission rates to better understand the complexity of their patient populations and enhance quality. Thirty-day readmission rates for hospitalized diabetic patients range from 14.4% to 22.7 percent, which is significantly higher than the rate for all hospitalized patients (8.5–13.5 percent)". [1]

The project's goal is to predict the likelihood of a diabetic patient being readmitted and to identify factors that lead to higher readmission in such patients, which can lead to risk-standardized planned readmission programs for the patients with the highest risk to be readmitted.

To meet our goal, we used classification problems as Logistic Regression, naïve Bayes, SVM, and KNN. Per the nature of the data, which was unbalanced and our goal to predict the readmitted patients (positive labels), we focused our efforts to get a better recall metric.

Related Work

This dataset was first analyzed by the authors of the paper "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records" [2] in which the authors found that "The statistical model suggests that the relationship between the probability of readmission and the HbA1c measurement depends on the primary diagnosis". The study concludes that finding a predictor for readmissions would reduce costs for the individuals with the disease.

Also, this dataset has been shared in the UCI repository, and it has been the subject of different models by students and competitors in universities and the Kaggle community.

Models

We decided to work with four classification problems we learned in class and compared the performance of its PR AUC and Recall metrics.

Naïve Bayes

This is a generative model based on the Naïve Bayes theorem. It is often useful with categorical data and assumes the conditional independence of the features as Naïve Bayes calculates the probabilities of each instance to occur; it works well for classifications problems. We used Naïve Bayes as our base model, the one we are trying to improve.

Logistic Regression

While we use linear Regression for predicting continuous values, we used Logistic Regression to map the confidence of categorical labels occurrences into a probability in the range of [0,1], which is why we can say this is a model based on probabilistic intuition and a good model for binary decisions and classification problems.

Logistic Regression works through Linear Regression. The process starts when we first get the estimations of continuous values through Linear Regression and continues through a link function called Sigmoid. The Logistic Regression maps those continuous values into bounded values between 0 and 1, which are the probabilities of occurrences of our predictions.

$$\text{Map } [-\infty, \infty] \text{ to } [0,1] \text{ by } g(z) = \frac{1}{1+\exp(-z)}.$$

According to the AUC and PR results, Logistic Regression performed the best in our model selection. We used grid-search and 10-fold cross-validation to tune our parameters and find the best model. L2 regularization penalty was found to be the penalty parameter for our model. It is equal to the square of the magnitude of coefficients.

10-fold Cross-Validation basically sets 10% of the data as our test set 10 times and then trains the model on the remaining 90% ten times. Then we choose the best performing model and retrain it on the entire dataset. This helps us find the best model and use different test/train splits.

Support Vector Machine

This classification model based on geometric intuition tries to maximize "margins" that split the plane into positive-classified examples and negatively classified ones. To choose the right margins, the model must choose the right intercept b, which will separate the plane to get the function in which the data points will be evaluated and then classified as positive or negative values (-1,1).

$$\text{For a new data point } x \in \mathbb{R}^n, \text{ evaluate. } z = w^T x + b$$

For non-linear data, Support Vector Machine adds a third dimension which converts planes into a hyperplane making it easier to separate and thus classify data; while this can become computationally expensive, SVM makes use of kernels to transform the data and find the proper decision boundary.

In our case, since we do not have enough computational resources, we used Linear SVM, which is the fastest one to run on both our balanced and unbalanced data.

K-Nearest Neighbors KNN

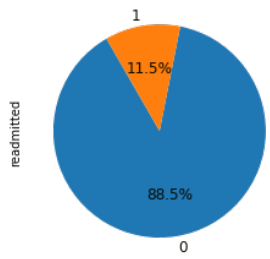
This is a classification model based on a voting system based where the label of the majority k nearest neighbors will make our goal prediction. To choose the right k in the diabetes data set, we performed Cross-validation with ten folds where we iterated the odd numbers from one to fifty, then evaluated the AUC score in each round and chose the best performing k. By default, Euclidian distance is used to train our model. The results showed k=5 as the best parameter.

Experimental Results

Data set Description:
The data collection covers ten years of clinical treatment at 130 US hospitals and integrated delivery networ ks (1999-2008). It has 49 features and about 101766 instances that indicate patient and hospital outcomes.

Data Preprocessing and Feature Engineering

Prediction models are challenging to create since the raw data consists of inconsistent or redundant records with incomplete and high dimensionality features. Therefore, it receives the necessary attention to be converted into a proper format.



The readmission goal variable is divided into two categories: readmitted for less than 30 days and non-admitted for more than 30 days. Initially, it had three categories, No, <30, and >30. Our data is highly imbalanced, with 88.5% labeled as >30 and the rest labeled as <30.

To guarantee that each instance is statistically independent, only the first experience per patient is examined.

The first thing we did for the data engineering section was a check for NA values. We found out that our data has different types of missing values. Most of them had the symbol '?'. Others had 'Unknown/Invalid' and 'unknown.' We ran a report that showed percentages of missing values in each field and decided to completely drop the features that had more than 40% missing. Those fields were:

- Weight with 96% missing values.
- Payer code with 40% missing values
- Medical specialty with 39% missing values.

For the other columns that had less than 40%, we had the following:

- Race: had less than 3% missing values, so we decided to drop those rows
- Gender: had less than 1% missing values, so we decided to drop those rows.

For all our categorical variables, we had to map them to numerical variables to run models such as SVM and Logistic Regression. This data set has a lot of numerical fields that have meaningful medical ranges. We referred to a medical research paper on how to interpret these fields, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records."

Diag1, Diag2, and Diag3 contain numerical values. Each range corresponds to a specific type of diagnosis. We used rules we found on "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records." Research paper to map Diag numerical ranges to their right category. For example: range (460,520) => Category 2. A similar strategy was applied to all the medicaments and medical tests: Max Glucose Serum, A1Cresult, 23 fields of medicaments.

We also drop unmeaningful columns such as: 'encounter_id' and 'patient_nbr'.

Evaluation Metrics

The result is evaluated by a confusion matrix as illustrated in the table as follow:

Confusion Matrix

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

According to the confusion matrix, TN is the number of negative samples correctly classified. FP is the number of negative samples incorrectly classified as positive. FN is the number of positive samples incorrectly classified as negative. TP is the number of positive samples correctly classified. In our project we need to

maximize the TP.

The overall accuracy:

$$\textbf{Accuracy} = \frac{(TP+TN)}{(TP+FP+TN+FN)}$$

True Negative Rate (TNR):

$$\textbf{TNR} = \frac{TN}{N} = \frac{TN}{(TN+FP)}$$

True Positive Rate (TPR) or Recall:

$$\textbf{TPR} = \textbf{Recall} = \frac{TP}{P} = \frac{TP}{(TP+FN)}$$

False Negative Rate (FNR):

$$\textbf{FNR} = \frac{FN}{P} = \frac{FN}{(TP+FN)}$$

False Positive Rate (FPR):

$$\textbf{FPR} = \frac{FP}{N} = \frac{FP}{(FP+TN)}$$

Precision:

$$\textbf{Precision} = \frac{TP}{(TP+FP)}$$

F_1 is the harmonic mean of Precision and Recall, and F_1 is obtained by:

$$\textbf{F}_1 = 2 \times \frac{precision \times recall}{precision + recall} = \frac{TP}{TP+0.5(FP+FN)}$$

The ROC curve can be plotted using TPR and FPR at different thresholds. AUC measures the area underneath the ROC curve. AUC and F_1 Score are used as our main measurements to evaluate the models for this project. We want to focus on increasing Recall as it is the most important metric for our model evaluation, as the cost of FN is much higher than the cost of FP in our case.

Imbalanced Dataset Solution

Two methods are proposed to solve this problem. One method is to use an imbalanced training set to train the model and select the optimal threshold. Another method is to use down-sampling and Random Over sampler algorithms for sub-sampling in each fold of the 10-fold cross-validation of the imbalanced training set. Since the calculation limits, without the use of the sampling in this case. After trying both solutions, Over-sampling gave us the best results.

Hyperparameters Tuning

Each model is trained by cross-validation to optimize the model's parameters and prevent overfitting. We also use grid-search to loop through different parameters for each model.

Model Choices:

- Naive Bayes - Baseline Model
- Logistic Regression * Best performance

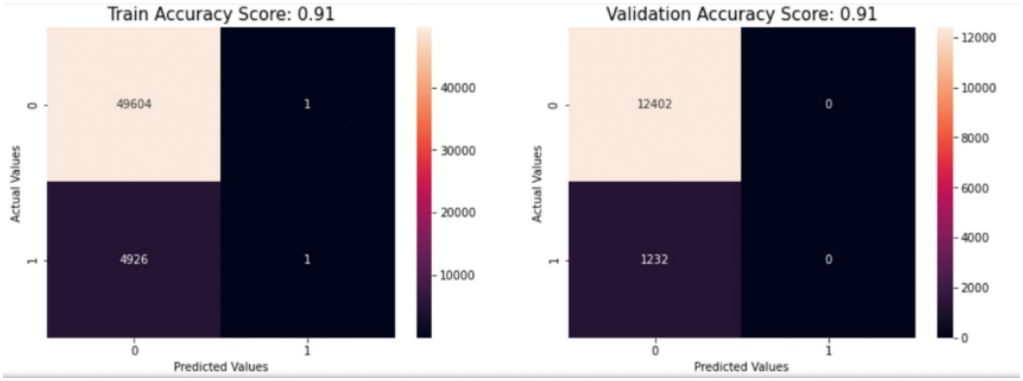
- SVM
- Knn Classifier

The following section covers the detailed results of each model.

Naïve Bayes
We chose Naïve Bayes as our baseline model. We want to get better results than our best Naïve Bayes model.
The initial Naïve Bayes model gave us the following results:

Accuracy	0.91
Precision	0.00
Recall	0.00

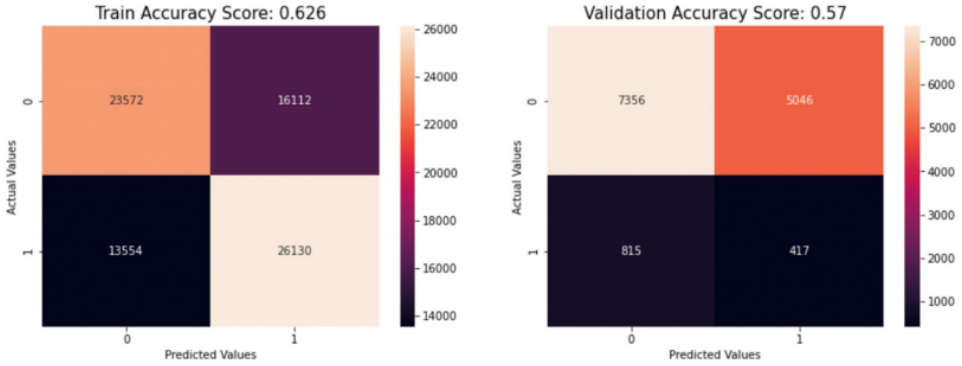
AUC = 0.59



After running the initial Naïve Bayes model on the cleaned dataset and no parameter tuning, we get very high accuracy and low Precision and Recall. From the confusion matrix, we could also see that the false positive and true positive rates are all 0. This means that the model is trying to predict everything as not readmitted to reach high accuracy. We will now try to tune our model to improve the recall score by balancing the data. Below are the results:

Accuracy	0.57
Precision	0.08
Recall	0.34

AUC = 0.45



Recall slightly improved, but overall performance is still very low. AUC is at 0.45, and overall accuracy also decreased a lot. We will try to perform better than this baseline model in the following sections.

Logistic Regression:

The binomial logistic regression algorithm works by finding the theta value that maximizes the following log likelihood function:

$$l(\theta) = \log L(\theta)$$

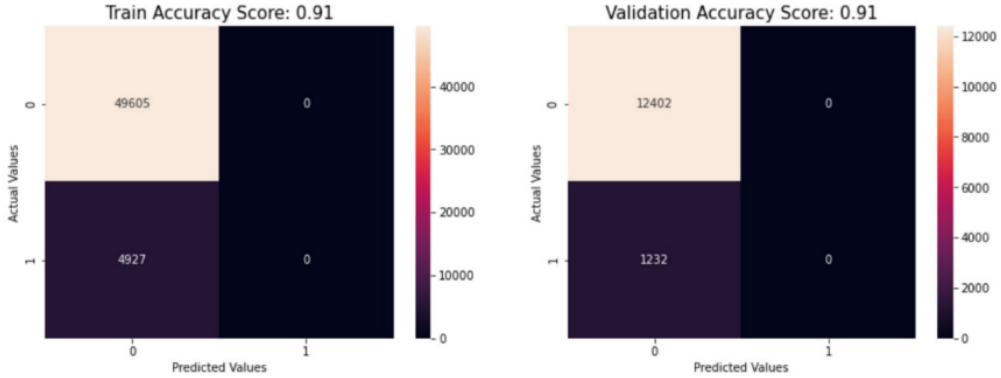
$$\log L(\theta; x, y) = {}^{m}\Sigma_{i=1}\, y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))$$

In the above equation, $x^{(i)}$ denotes the features matrix, $y^{(i)}$ represents the labels vector, and $h_\theta(x)$ is $1/(1 + e^{-\theta T(x)})$.

We are getting the same issue again with Logistic Regression. Let us tune the model.

Accuracy	0.91
Precision	0.27
Recall	0.00

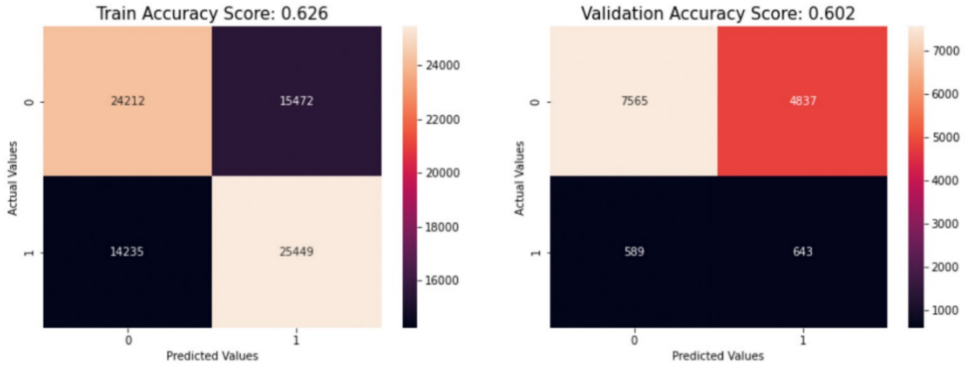
AUC 0.59



In order to improve our model, we used oversampling technique, grid search, and 10-fold cross-validation. Our best model yielded the following parameters: penalty='l2', C=1, max_iter=500. After running the tuned model, we got the following results:

Accuracy	0.60
Precision	0.12
Recall	0.52

AUC 0.58



We could see that Recall improved a lot after running our best LR model; also this lowered the overall accuracy. Since we are focusing on Recall, this is our best-performing model so far.

K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a supervised machine learning algorithm that classifies each new observation based on how similar to its neighboring data points. The similarity is measured using distance metrics, and Euclidean distance is the most common way to measure it. The equation for Euclidean distance is as below

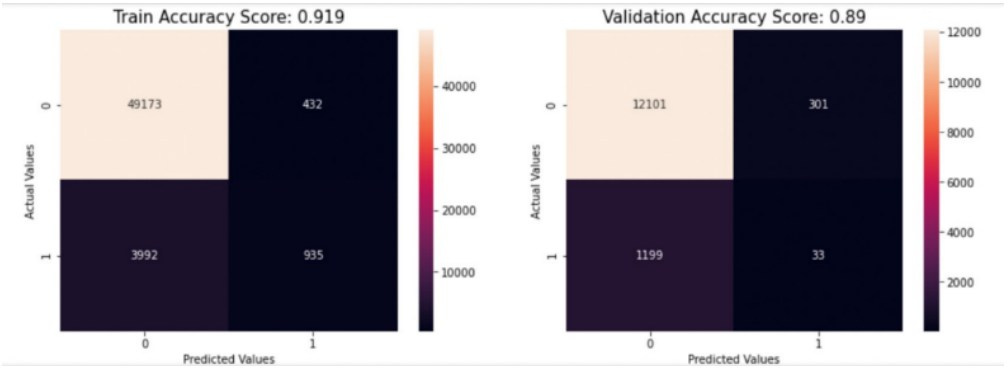
$$h(x') = \arg\max_{y \in Y} \{\Sigma_{i \in kNN(x')}\, 1_{[y(i) = y]}\}$$

In the above equation, x′ denotes a new example, y′ denotes the best prediction.
It is a non-parametric and lazy learning algorithm because it does not assume data distribution, and it must be run at prediction time. It uses Euclidean distance: therefore, normalization is strongly recommended. Also,

KNN is extremely sensitive to the dimensionality of data and outliers. It is not suitable for many features, and outliers should be treated.
We used kfold cross-validation to find the best K for this model; according to that, we got K = 5.
Below are the results we got after running the initial Knn with K=3:

Accuracy	0.88
Precision	0.10
Recall	0.03

AUC = 0.50

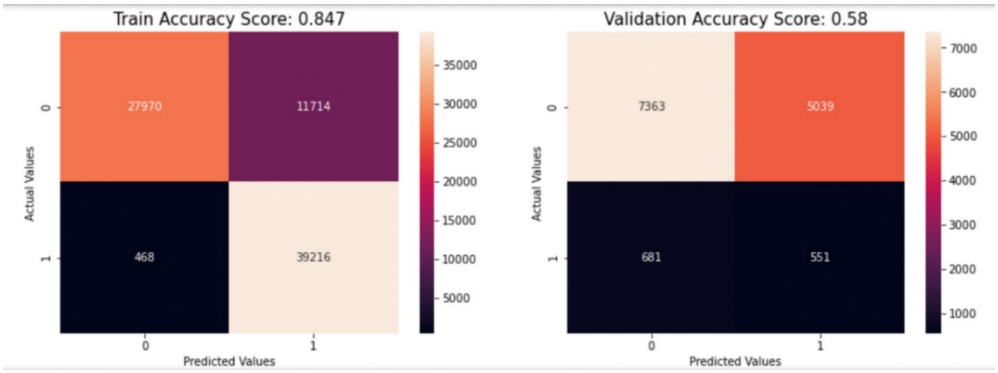


Again, we are getting the same issue with Logistic Regression on our initial model. In the next section, we try to improve the performance.

In order to tune our Knn model, we ran ten-fold cross-validation with different K values ranging from 1 to 50. The metric we chose to evaluate against is AUC.
The best K came back as K=5. Below are the results of our best Knn.

Accuracy	0.58
Precision	0.10
Recall	0.45

AUC = 0.53



We could see that our tuned model performs better, especially in terms of recall value, which means it is doing a better job at predicting the True Positives. However, Precision is still very low.

Support Vector Machine

Support Vector machines (SVMs) are a type of powerful classifier that can draw a hyperplane through multidimensional data. The distance between a point and the hyperplane determines the data prediction/classification accuracy. The goal of using an SVM for early readmission prediction on the development set was to see if the data had any geometric order.
Given linearly separable training data, we can learn SVM by

$$y^i(w^T x^i + b) \geq \gamma \text{ (for all i) and } ll\ w\ ll = 1.$$

Given the primal problem:

$$\min_{w,b} f_o(w) = \frac{1}{2} ll\ w\ ll^2$$

subject to $f_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0 \; \forall \; i$

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} ll\, w\, ll^2 - \Sigma^m_{i=1} \alpha_{(i)}\{ y^{(i)}(w^T x^{(i)} + b) - 1 \}$$

$g(\alpha) = inf_{w,b}\, \mathcal{L}(w, b, \alpha)$ so we need to know w,b that minimizes $\mathcal{L}$

$$w = \Sigma^m_{i=1} \alpha_{(i)} y^{(i)} x^{(i)} \;\; \& \;\; \Sigma^m_{i=1} \alpha_{(i)} y^{(i)} = 0$$

Therefore, the dual function will be

$$g(\alpha) = \Sigma^m_{i=1} \alpha_{(i)} - \frac{1}{2} \Sigma^m_{i,j=1} y^{(i)} y^{(j)} \alpha_{(i)} \alpha_{(j)} x^{(i)T} x^{(j)} - b \Sigma^m_{i=1} \alpha_{(i)} y^{(i)}$$

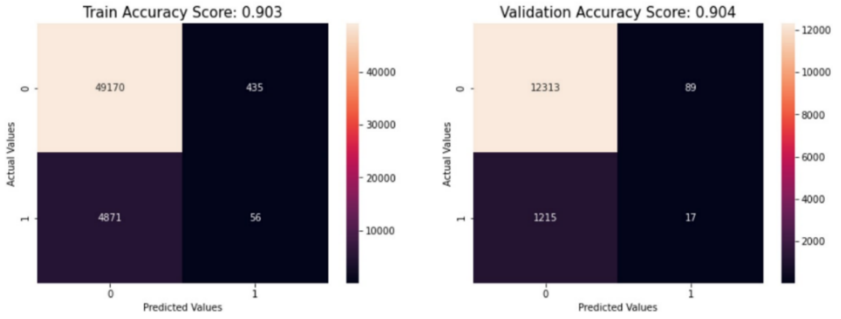
Different Kernels can also be used to help for computation which otherwise would involve computations in higher dimensional space. The dual then now is:

$$max_\alpha\, g(\alpha) = \Sigma^m_{i=1} \alpha_{(i)} - \frac{1}{2} \Sigma^m_{i,j=1} y^{(i)} y^{(j)} \alpha_{(i)} \alpha_{(j)} \langle x^{(i)} x^{(j)} \rangle$$

Subject to $0 \leq \alpha_{(i)} \leq C \forall\, i$ *and* $\Sigma^m_{i=1} \alpha_{(i)} y^{(i)} = 0$

Accuracy	0.90
Precision	0.16
Recall	0.01

AUC- 50.33



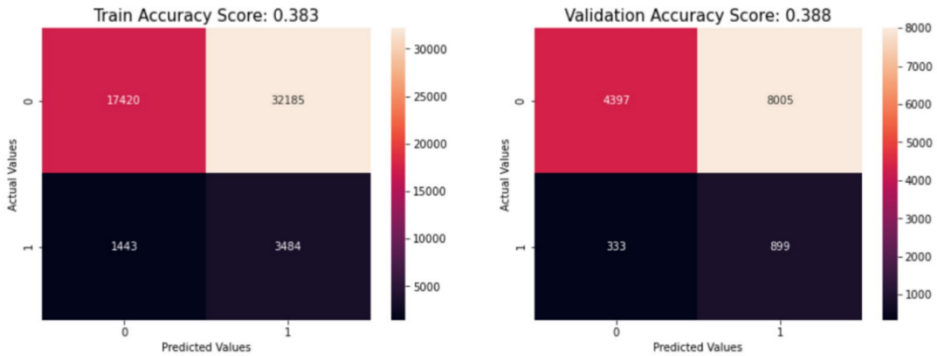
Again, High accuracy and low Precision/Recall. Let us try to improve this.

In order to tune our SVM model, we ran a 10-fold cross-validation and grid search to find the best C, Gamma, and Kernel. The resulting best model has {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}.

After running the best SVM model, we got the following results:

Accuracy	0.38
Precision	0.10
Recall	0.72

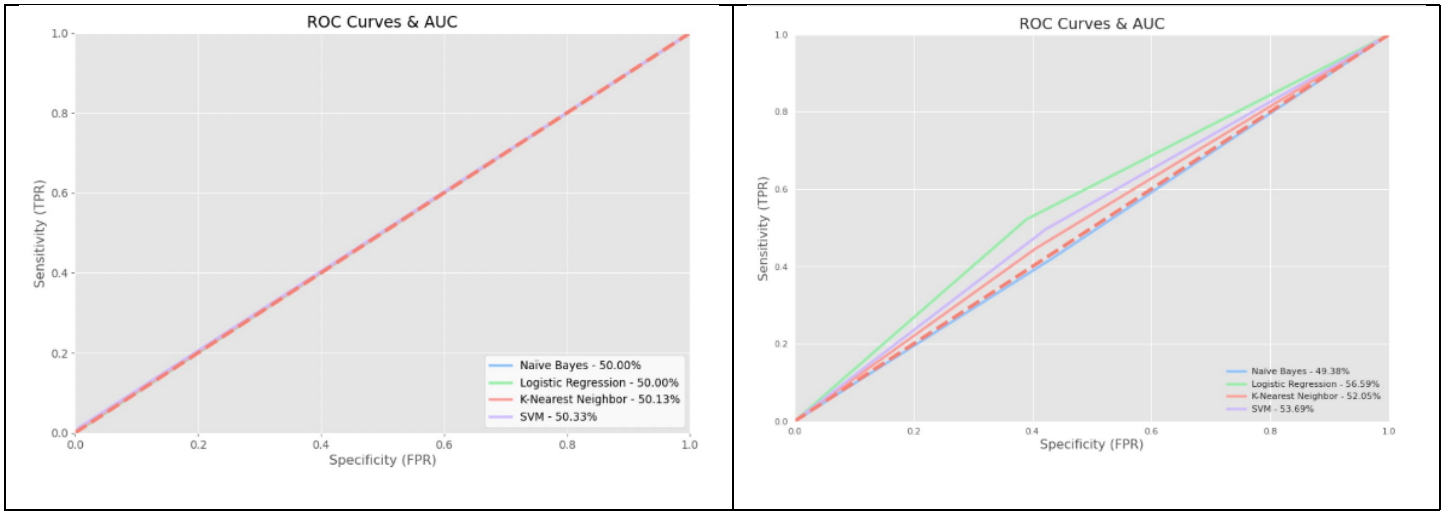
AUC- 54.21



Our Tuned SVM gives us the best Recall values at 0.72, which means it is doing a good job predicting the true positives. However, overall accuracy is at 0.38, which is the lowest compared to other models. At this point, the tuned LR is still our best-performing model.
This method uses the classic Naive Bayes algorithm, which assumes that all features are independent of one another.

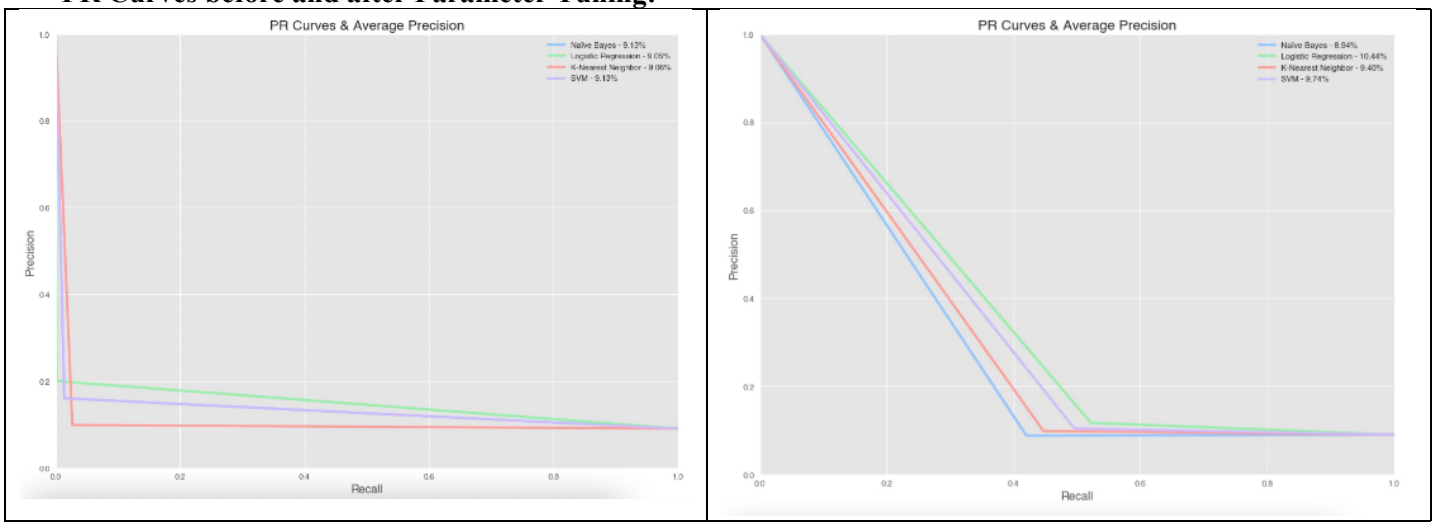
Final Results

ROC Curves before and after Parameter tuning:



As we can see from the ROC curve comparisons above, our initial models had very low AUC scores. The overall accuracy was high, but this just meant our model was predicting every patient as not readmitted to boost accuracy. On the right side, we see the AUC scores after tuning our models, and we could see that all improved in terms of AUC except for Naïve Bayes. The best performing models by AUC score in logistic regression (green line).

PR Curves before and after Parameter Tuning:



In the figure above, we see the comparisons of the PR curves before and after tuning the model. We could see a significant improvement, especially with the recall scores, which is precisely what we needed to improve in our model to maximize the TP predictions. (cancer-like prediction) Logistic regression is our best performing

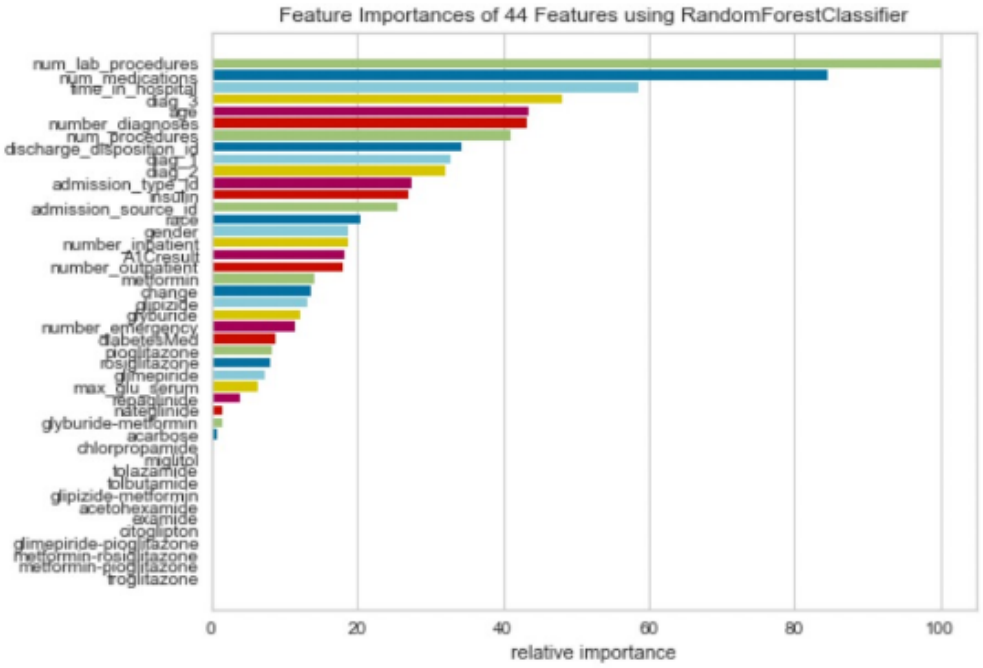
model based on these results as it was able to boost the recall score while maintaining a relatively good overall accuracy and precision score.

The table below summarizes all the scores before and after tuning the models.

Model	Accuracy Before	Accuracy After	Precision Before	Precision After	Recall Before	Recall After	AUC Before	AUC After
Naive Bayes	0.91	0.55	0.00	0.09	0.00	0.42	0.59	0.45
Logistic Regression	0.91	0.61	0.12	0.11	0.00	0.52	0.63	0.58
SVM	0.90	0.57	0.16	0.10	0.01	0.49	0.50	0.54
KNN	0.88	0.58	0.10	0.10	0.03	0.45	0.50	0.53

Feature Selection

Feature importance could help us improve our models by knowing which fields are more important in predicting the outcome. It could also help us give suggestions to the business on what fields to focus on. We will look at the most important features and do some more data analysis on them to understand their impact on whether a patient will be readmitted or not. We chose Random Forest model to perform feature selection as it is the model that gives us the highest scores (not covered in this paper).



The figure above shows the most important features in descending order.

Conclusion or Discussion

In conclusion, this Diabetic patients dataset requires a lot of feature engineering as it contains many features and a lot are medical technicalities that require medical expertise. We used researcher papers published in the medical field to properly understand and handle these fields. We developed four models: Naïve Bayes, Logistic Regression, Knn and SVM. After many attempts to improve them, we compared all four models and performed models selection. In this Diabetic readmission classification problem, the most important measure we relied on is recall which measures the percentage of actual positives our model predicted to be positive. What percentage of patients readmitted in less than 30 days do our models predict to be readmitted. Our focus is not on accurately predicting the positive cases but making sure we have captured all the positive cases. This is because the consequences of predicting a patient not to be readmitted when they do are much severe. After comparing our models, we found that Logistic regression is our best model based on the ROC and PR curves. It boosted recall without severely penalizing overall accuracy and precision score.

The top 5 most important features are the Number of lab procedures, number of medications, time spent in the hospital, diag3, and age. These features make sense medically, as the more procedures/medications, a patient has the graver their case is. Time in the hospital also hints on the condition of the patient and whether they will be readmitted. Based on these results, we advise the business to focus on patients having long stay times in the hospital, a higher number of procedures and medications.
In future work, we will also use the results from feature importance to get new feature ideas. This will help us compile a new list of features that we can use for feature reduction. PCA for dimensionality reduction is another area we want to experiment with, as capturing our data in fewer dimensions could impact our predictions. Finally, we want to try other ensemble models such as Random forest, which can boost our results.

Citations

[1] Ostling, S., Wyckoff, J., Ciarkowski, S.L. *et al.* The relationship between diabetes mellitus and 30-day readmission rates. *Clin Diabetes Endocrinol* 3, 3 (2017). https://doi.org/10.1186/s40842-016-0040-x
[2] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.