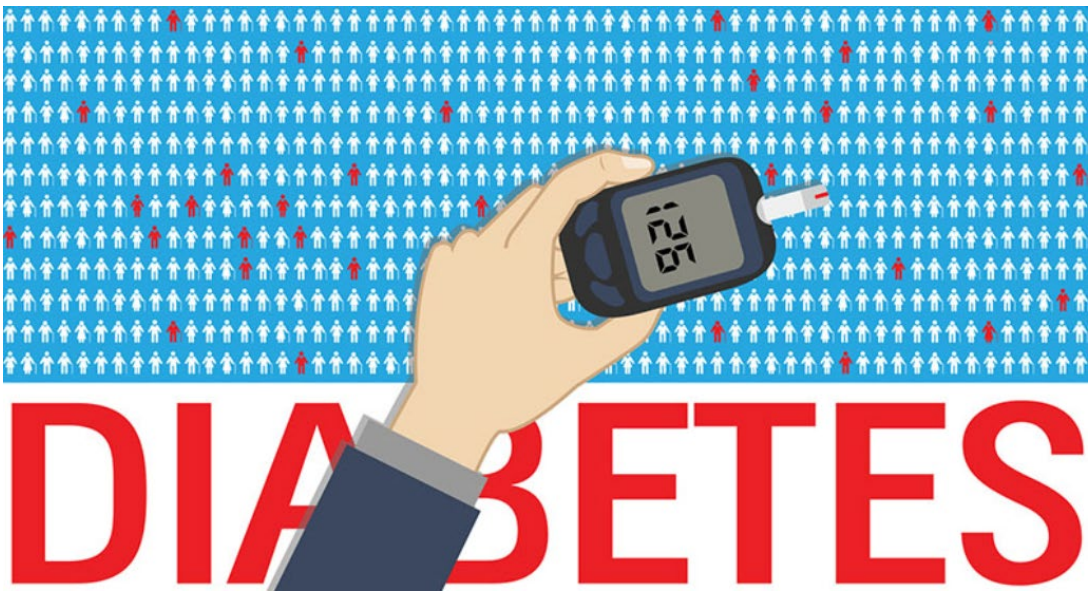

HOSPITAL DIABETIC CASES READMISSION

Data698 – 11/29/2020

Abdelmalek Hajjam, Monu Chacko, Md Forhad Akbar, Shovan Biswas

Contents

Team Members

- Abdelmalek Hajjam
- Monu Chacko
- Md Forhad Akbar
- Shovan Biswas

Abstract

Diabetic Disease is the seventh leading cause of death in the United States. It affects thousands of people all over the states. Many diabetic patients got readmitted to hospital within 30 days after discharge due to various reasons. Such readmission can usually be avoided if additional attention is paid to patients with high readmission risk and appropriate actions are taken. This makes early prediction of the hospital readmission risk an important problem. The goal of this research is to conduct a study on evaluating different types of classification machine learning models, for predicting the readmission risk of diabetic patients. We evaluate those algorithms on a dataset of 100,000 records and 52 features, obtained by UCI machine learning repository. Our analysis showed that the prediction performance in terms of Accuracy and AUC (Area Under the Curve) can be improved by using autoML on the cloud.

Introduction

Hospitalizations account for almost one-third of the total health care spending in the United States. A substantial portion of those hospitalizations are readmissions, which is why the Hospital Readmissions Reduction Program (HRRP) was created by CMS with the goal of improving the quality of care for patients and reducing healthcare spending.

Hospital readmissions within 30 days of discharge (30-d readmissions) have become a high-priority healthcare quality measure and target for cost reduction.

The Centers for Medicare and Medicaid Services (CMS) now penalizes hospitals for excess readmissions rates for certain conditions, with up to a three percent payment reduction. For many health systems, this can translate into millions of dollars. In fact, just one or two excess readmissions in populations such as diabetes, Heart Failure and Knee Replacements can tip a hospital over its allowable readmission rate.

The most common reasons for readmission according to primary discharge diagnoses were diabetes, heart failure, procedural complications, chest pain, shortness of breath, acute kidney failure, and urinary tract infection.

Diabetes is the seventh leading cause of death in the United States. Diabetes is the condition with the 3rd most all-cause, 30-day readmissions for Medicaid patients, and in 2011, American hospitals spent over $41 billion caring for diabetic patients who were readmitted within 30 days of discharge [1]. Readmission of diabetes patients can usually be avoided if additional attention is paid to these patients with high readmission risk and appropriate actions are taken. This makes early prediction of the hospital readmission risk an important problem. Being able to predict which patients will be readmitted could help save hospitals billions of dollars while also improving quality of care [1].

Goal and Modeling

The goal of this project is to see how well we can predict 30-day hospital readmission of diabetes patients, i.e. classify and identify patients that are at risk of being readmitted after being discharged. We should be able to predict whether the patient will be readmitted to the hospital or

not. For predicting it most accurately we will be using various prediction models for this purpose.

Because this is a classification problem, to predict whether a patient will be readmitted or not, we will be using six different classifiers, namely, Support Vector Machines, Generalized logistic regression, Artificial Neural Networks, Random Forest Classifier, Naïve Bayes Classifier and Decision Trees. The models will also be used to help determine what factors are the most important in predicting hospital readmission for diabetic patients.

After building, training, and testing the model, our models should be able to classify patients (yes/no) being readmitted within 30 days.

Data Source

The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks [1]. It consists of 100,000 records and 52 features representing patient and hospital outcomes. This dataset is made publicly available by the UCI Machine Learning Repository.

The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result [1], diagnosis, number of medications, diabetic medications, number of outpatients, inpatient, and emergency visits in the year before the hospitalization, etc.

This dataset will undergo intense cleaning procedures to be ready for modeling. Feature selection for these models will be done by conducting Correlation Analysis, and eliminating

features with class imbalance. Data will be converted and pre-processed by removing null values and changing categorical variables.

Literature Review

Before embarking upon our research, we reviewed the literature and got a sense of the history of what has been done in the past. Reducing hospital readmissions continues to be a high-priority task across the healthcare systems, because unnecessary hospitalizations not only expose the patients to potential infections (especially in times of COVID19 like this), but also increase costs.

Since we are doing this research for a Data Science project, we like to explore the applicability of machine learning and other data science techniques. So, we checked the literature and found significant work have been done in that regard, especially in the area of readmissions for diabetes. Extensive study of HbA1C tests, as predictor for readmissions within 30-day period has been performed. Almost 70,000 encounters were considered with about 50 features. Algorithms like Logistic Regression, Decision Tree, Random Forest, Adaboost and XGBoost were deployed in such studies [1]. While readmission rates remained the highest for circulatory diagnoses, readmission rates for patients with diabetes appeared to be associated with the decision to test for HbA1c, rather than values of its result.

More than $25 billion are spent each year on diabetic readmissions alone. Sarthak et al evaluated existing models and proposed new embedding based on deep and neural network (DNN), which can predict whether a hospitalized diabetic patient would be readmitted within 30-days or not, with an accuracy of 95.2% and AUROC of 97.4%, based on data collected from 130 US hospitals between 1999-2008 [2]. The results have been encouraging with patients having

changes in medications. Identifying prospective patients for readmission could reduce readmission costs. Every time a patient is admitted to a hospital, creatinine level is tested. Ben-Assuli et al used multivariate logistic regression model on this indicator (creatinine) from Electronic Health Records of 5,103 patients [3]. Their findings suggest that three significant components impacting readmission are age, gender and creatinine levels.

Different researchers approached with different combinations of machine learning. Joshi and Alehegn (2017) reviewed various data mining techniques, which are generally used to predict chronic diseases. Classification techniques of machine learning, e.g. Support Vector Machine, Naïve Bayes and Decision Tree work well in generating better diagnoses predictions for diabetes and other diseases [4]. In order to classify the risk of diabetes, Nongyao et al [5] applied four machine learning classification methods namely Decision Tree, Artificial Neural Networks, Logistic Regression and Naïve Bayes. They improved the model with Bagging and Boosting techniques. Although Random Forest algorithm gives optimum results among all the machine learning algorithms they used, Hammoudeh et al [6] used a balanced combination of Convolutional Neural Networks and data engineering against real life data, to predict readmissions in early stages and thereby reduce costs and preserve reputation of the hospitals.

Patient wellness score integrates many lifestyle components and a holistic patient prospective. Agarwal et al used a large comprehensive survey of over 5000 people, conducted by the CDC, to build machine learning models. 8 out of the 9 models were shown to have a statistically significant ($p = 0.05$) increase in area under the receiver operating characteristic when using the hybrid approach when compared to expert-only models [7]. We also reviewed some literature on use of big data to evaluate risk of readmissions for diabetes patients. A paper [8] by Salian recommends a prescient model that can discover the hazard indicators on patients

with perpetual diabetes. The data was analyzed on Hadoop cluster. Weight, plasma, glucose, age, pregnancy, family work was found to be some of the top indicators of readmission for diabetic patients.

As of 2012, according to American Diabetic Association, more than 29 million Americans were diabetic. S. Behara's paper compared multilayer perception (MPL) and Bayesian networks (BNs) in diabetic patient classification on a cohort of 5000 individuals, surveyed by the CDC. This paper [9] showed that MLP models predict with larger AUC, higher accuracy and lower RMSE.

Among the few other papers, we consulted were by Goudjerkan [10], where they used Random Forest for feature selection and SMOTE algorithm for data balancing. They also proposed MLP on data collected from 130 American hospitals. The proposed combination of data engineering and MLP was found to outperform existing practices. Two other papers by Kumar et al [11] used predictive analysis on in Hadoop and Baechle et al [12] used Naïve Bayes.

Methodology

Data:

The dataset we will be using was obtained from the UCI Machine Learning Repository. It represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks [1]. It consists of 100,000 observations and more than 50 features. The response variable is a binary outcome named "readmitted" (0 or 1), referring to whether or not the patient was readmitted within 30 days after his/her visit Fig 1. We have a classification problem in hand.

Many steps were taken to clean this dataset such as removing features that were irrelevant, redundant or containing too many missing values. We combined other variables under one variable as seen necessary. After our final features were settled, correlation matrix was built to verify variables dependency. After that a dimension reduction algorithm (PCA) was run and showed that our 2 readmission classes were not well separated fig 2. The **final dataset** consists of **70420 observations** and **42 variables** (41 independent, 1 response).

Our Data seems to be highly imbalanced; 641829 cases (91%) were **NOT** readmitted and 6291 cases (9%) were readmitted. Algorithms in general tend to be bias toward the majority class (i.e. when one class dominates the other) when dealing with classification problems. To solve this problem, we will be using SMOTE (Synthetic Minority Over-sampling Technique) to smartly create new instances of the minority class, so the number of minority instance will be almost similar of the number of instances in the majority class (these are not duplicate). Our Data will be spitted in 2 proportions, one set for training (80%) and another set for testing (20%). SMOTE will only be applied to the training set.

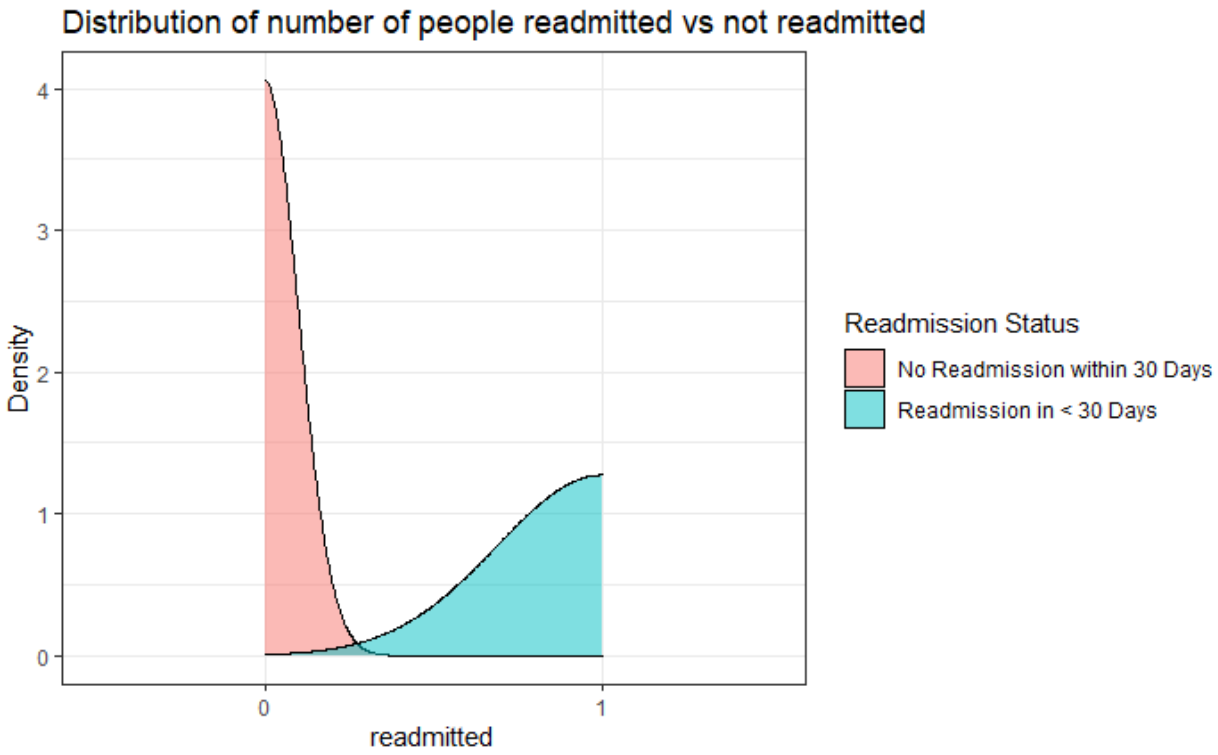
Distribution of number of people readmitted vs not readmitted
Density
4
3
2
1
0
0
1
readmitted
Readmission Status
No Readmission within 30 Days
Readmission in < 30 Days

Fig 1

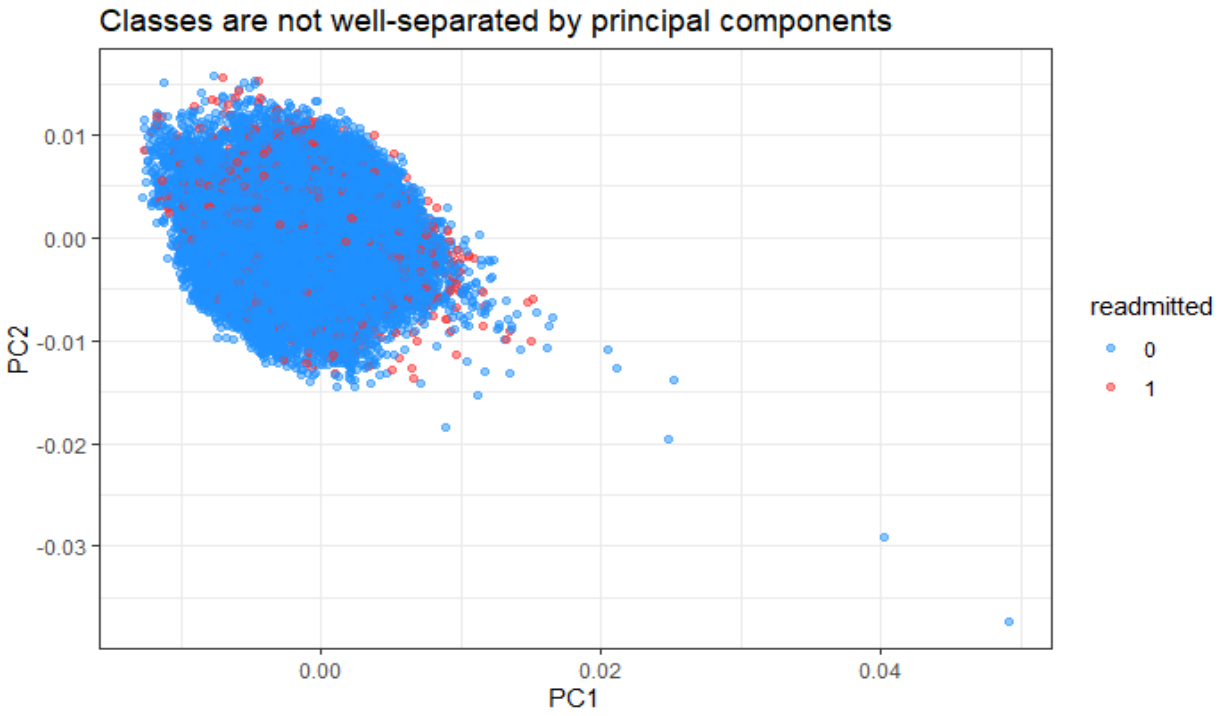
Classes are not well-separated by principal components
PC2
0.01
0.00
-0.01
-0.02
-0.03
0.00
0.02
0.04
PC1
readmitted
0
1

Fig 2

Distribution of few variables:

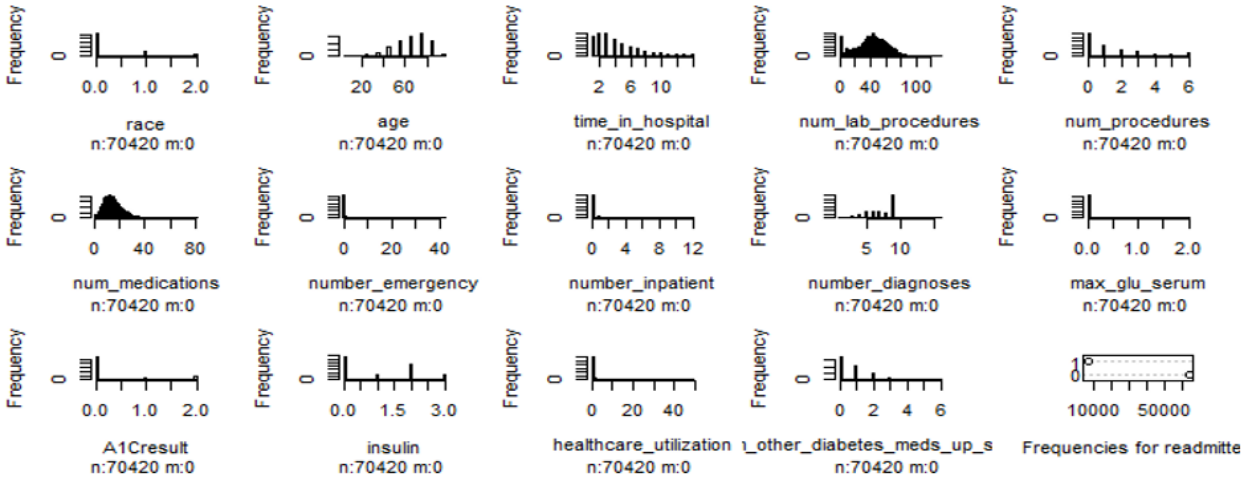

Fig 3

Algorithms:

This is a classification problem (1 or 0), to predict whether a patient will be readmitted or not, we will be using six different classifiers, namely, Support Vector Machines, Generalized logistic regression, Decision Trees, Random Forest Classifier, Naïve Bayes Classifier and Artificial Neural Networks.

Finally, we will use the latest algorithms in machine learning called **autoML**, an algorithm that runs many models in parallel using many trials while searching for the best hyper parameters for every algorithm that returns the best accuracy. This will be accomplished through **h2o** framework (locally), as well Azure ML (on the cloud).

We will compare all models to decide for the best that gives the best accuracy.

Machine Learning & Modeling

After our data cleanup, feature engineering, correlation analysis and dimensionality reduction, we settled on a clean dataset with 42 variables where 41 variables are predictors and 1 being the response (target) variable identified as “readmitted”.

Our principal component analysis and t-sne reduction algorithm showed us that the two readmission classes are not well separated. We also found that a small portion of the variance was explained by principal components.

Our proposed problem is a classification problem (whether the patient will be readmitted or not), therefore we need to consider some of the famous classifiers to work with. In this case we choose to work with SVM, Naïve Bayes, Logistic Regression, Neural Networks, Decision Trees and Random Forest. We will analyze each classifier accuracy and performance and compare all of them. In addition, we will use the brain new autoML with h2o framework to train the model and use an ensemble of algorithms, combine them and return the best one that gives the best performance.

As mentioned in the previous section, our data seems to be highly imbalanced; 64129 cases (91%) were **NOT** readmitted and 6291 cases (9%) were readmitted. When you have imbalanced data like in this example, any classification model will look at the data and cleverly decide that the best thing to do is to always predict the majority class and achieve high accuracy. Because 91% of patients are not readmitted, the classifier will have no problem deciding for us that any patient will indeed not be readmitted. This is totally wrong. Therefore accuracy is not a robust measure of classifier performance in presence of class imbalance. There are metrics that

have been designed to tell you a more truthful story when working with imbalanced classes, such as Precision, Recall and F Beta.

Another way to do it, is to use SMOTE (Synthetic Minority Over-sampling Technique) to intelligently create new instances of the minority class, so the number of minority instance will be almost similar of the number of instances in the majority class (these are not duplicate). In this case we can carry on with the Accuracy as a metric. Only the training data is SMOTed. We never apply SMOTing to test data.

When all algorithms are done running, a summary matrix is presented with all performance metrics for all classifiers.

Naive Bayes

Naive Bayes is a supervised learning classification algorithm based on Bayes Theorem. It is an extension of Exact Bayes classifier and assumes all variables are independent of each other. This method is suitable when dimensionality of inputs are high. It uses a simple method of conditional probability and can be computationally inexpensive and outperform more sophisticated models.

The parameters of the model of a Naive Bayes classifier i.e. a priori and conditional probabilities are determined using a deterministic set of steps. This involves two very trivial operations: counting and dividing. Naive Bayes is faster because all it needs are the prior probability values that do not change and can be stored ahead of time. The same probability values are reused in while calculating the posterior. It does not compute a giant distribution; therefore less computation.

- There is no iteration".
- There is no "epoch".
- There is no "optimization of a cost equation", Sometimes at least O(n^2).
- There is no "error back-propagation".
- There is no operations involving "solving a matrix equation".
- During inference phase, it uses Bayes equation to compute the a posteriori probability.
- This involves trivial arithmetic operations like addition and multiplication and further normalization is only a division by a scalar.
- So the inference is FAST.

This classifier (Table 1) achieved an accuracy of 55% in our case.

Table 1: Metrics for Naive Bayes

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
NB	0.56	0.55	0.55	0.43	0.57	0.45

Table 1

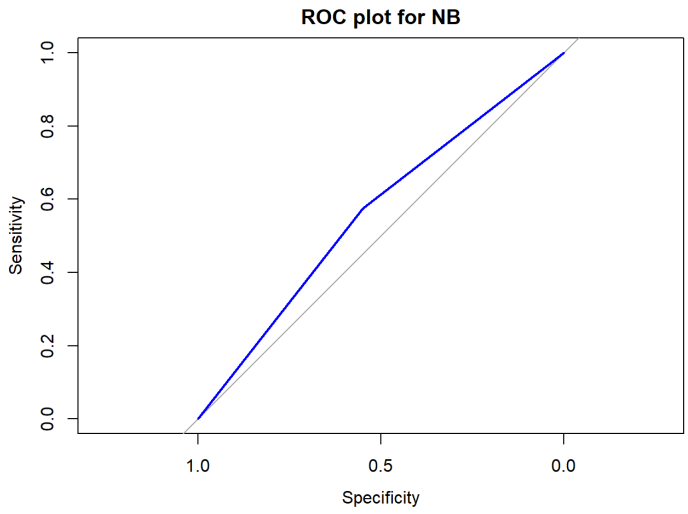

Fig 4

SVM

A support vector machine (SVM) SVM is a linear classifier. SVM is linear in separating the data by a linear separator. By default, it is a binary classifier and a linear separator.

This concept of separating data linearly into 2 different classes using a Linear Separator or a straight linear line is called Linear Separability. SVM requires lot of data for training. It is computationally expensive. SVM spends lot of time in the training phase to come up with a margin called support vectors, collection of observations in the training set that define that margin. If you have n points, then when a new observation comes in, you will be able to decide which side of the margin this new observation lies in and therefore define its class (label).

SVM is great, as you do not have to carry all the points; and you can explain your decision based on real data. In addition SVM is an easy classifier to explain and fast in scoring once the model is trained. Also, SVM is very robust when it comes to outliers.

This classifier (Table 2) achieved an accuracy of 76% in our case.

Table 2: Metrics for SVM

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
SVM	0.57	0.76	0.81	0.67	0.33	0.19

Table 2

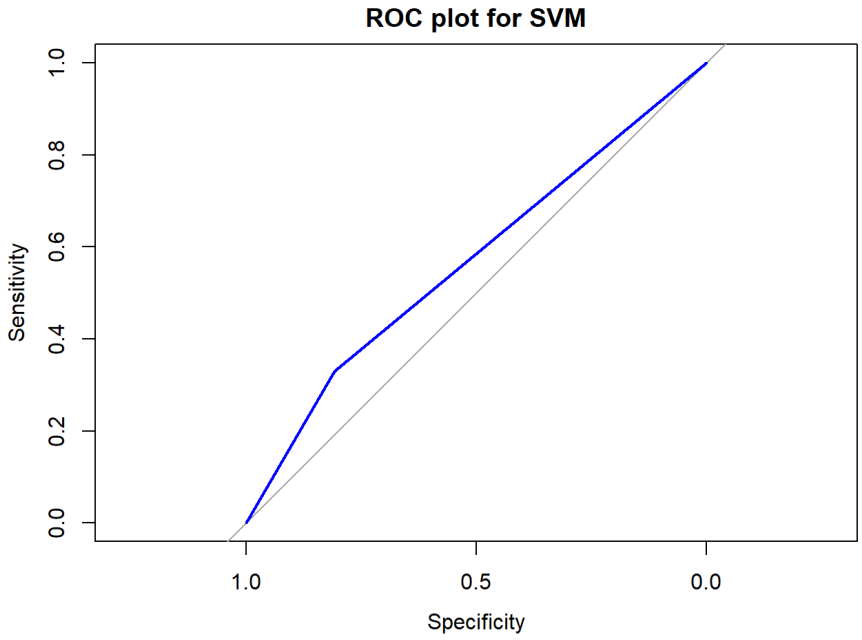

Fig 5

Logistic Regression

Logistic Regression is supervised learning classifier algorithm. Logistic Regression is used when the dependent variable (target) is categorical or binary. The goal of the this algorithm is to find the best fitting model that explain the response variable "readmitted" function of the independent variables. Logistic Regression is a discriminative classifier. Logistic regression directly models the posterior probability of P(y|x) by learning the input to output mapping by minimizing the error. Logistic regression tries to find the optimal decision boundary that best separates the classes.

This classifier (Table 3) achieved an accuracy of 76% in our case.

Table 3: Metrics for Logistic Regression

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
LR	0.62	0.76	0.8	0.66	0.34	0.2

Table 3

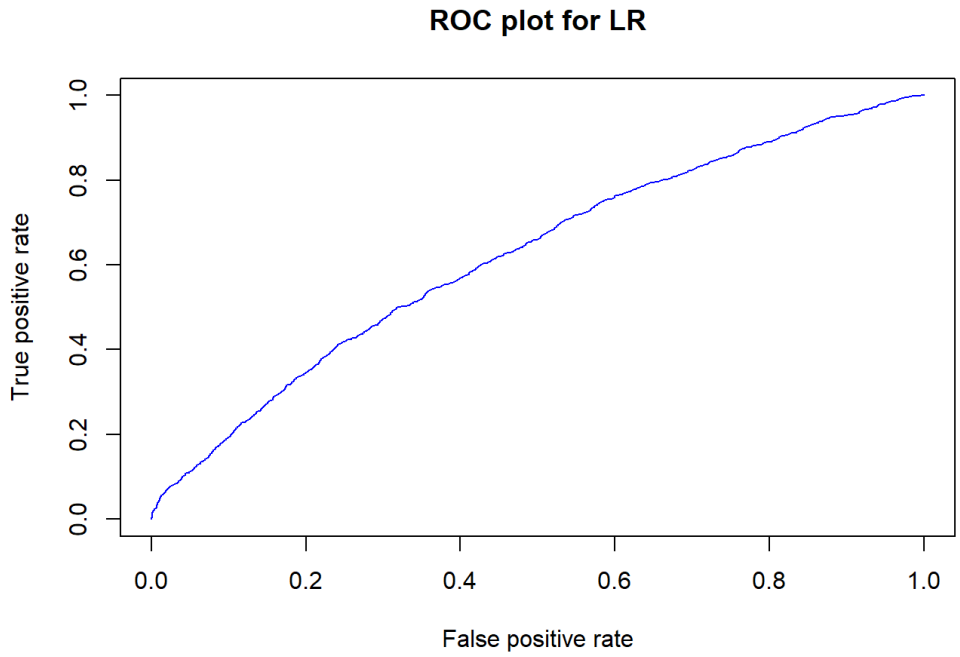

Fig 6

KNN

KNN is a classification algorithm that can solve classification problems by determining the majority class of nearest neighbors. If the majority of the nearest neighbors of the new data point belong to a certain class, the model classifies the new data point to that class.

The model representation for KNN is the entire training dataset. KNN has no model other than storing the entire dataset, so there is no learning required. KNN makes predictions using the training dataset directly. Predictions are made for a new instance (x) by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances. The k-nearest-neighbor is an example of a "lazy learner" algorithm, meaning that it does not build a model using the training set until a query of the data set is performed.

This classifier (Table 4) achieved an accuracy of 80% in our case.

Table 4: Metrics for KNN

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
KNN	0.52	0.81	0.87	0.82	0.18	0.13

Table 4

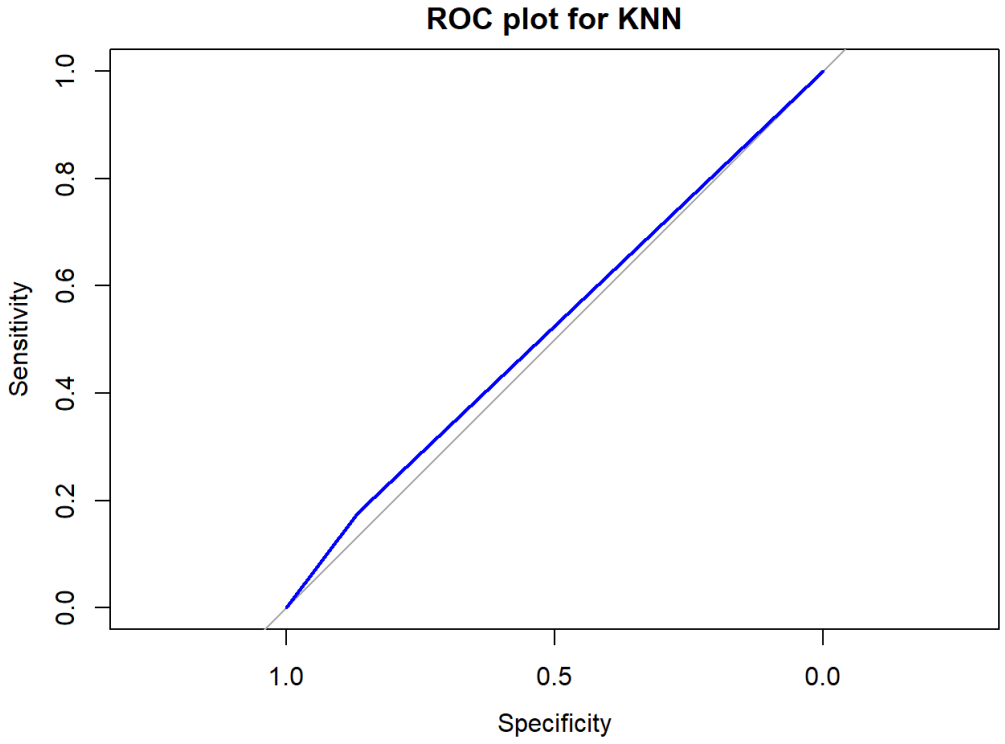

Fig 7

Decision Tree

A decision tree is yet another supervised classifier algorithm. This algorithm can be used for regression and classification problems – yet, is mostly used for classification problems. A decision tree follows a set of if-else conditions to visualize the data and classify it according to the conditions. Decision trees are non-linear, which means there's a lot more flexibility to

explore, plan and predict several possible outcomes to your decisions, regardless of when they actually occur. Decision Tree are a variation of non-parametric methods. Decision Trees partition the data into a directed acyclic inverted rooted tree. Each node is a test and based on the test results one of the branches is taken. The test are predicates on the covariates. This process is repeated until we reach a leaf node where all the observations belong to a class. The never seen before observation is assigned the class. Trees are prone to overfitting if trained excessively. The tree becomes deep too many levels of capturing noise in the training set. Therefore fails to generalize adequately.

This classifier (Table 5) achieved an accuracy of 86% in our case.

Table 5: Metrics for Decision Tree

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
DT	0.54	0.86	0.92	0.84	0.16	0.08

Table 5

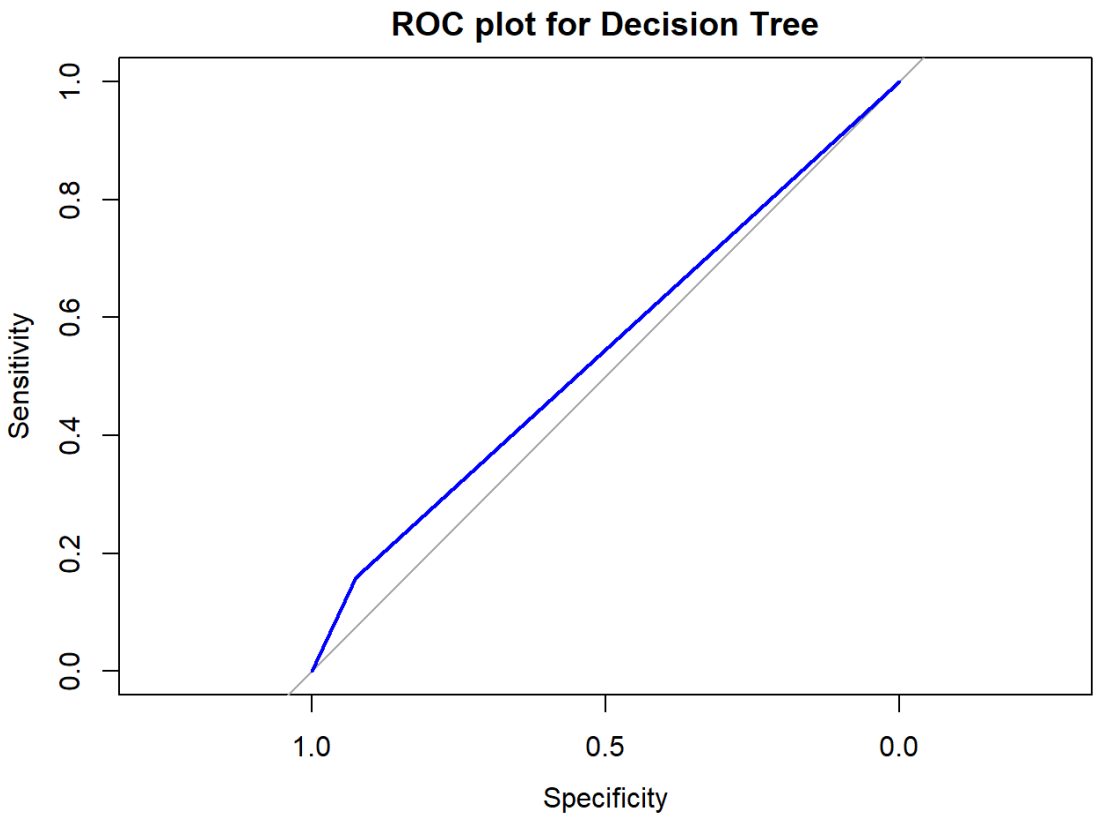

Fig 8

Random Forest

Random Forest is yet another classifier. Random forest, combine multiple trees, generate numerous models using random attribute sets and then take the average. Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. The reason the random forest model works so well is - a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

This classifier (Table 6) achieved an accuracy of 89% in our case.

Table 6: Metrics for Random Forest

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
RF	0.52	0.89	0.97	0.93	0.07	0.03

Table 6

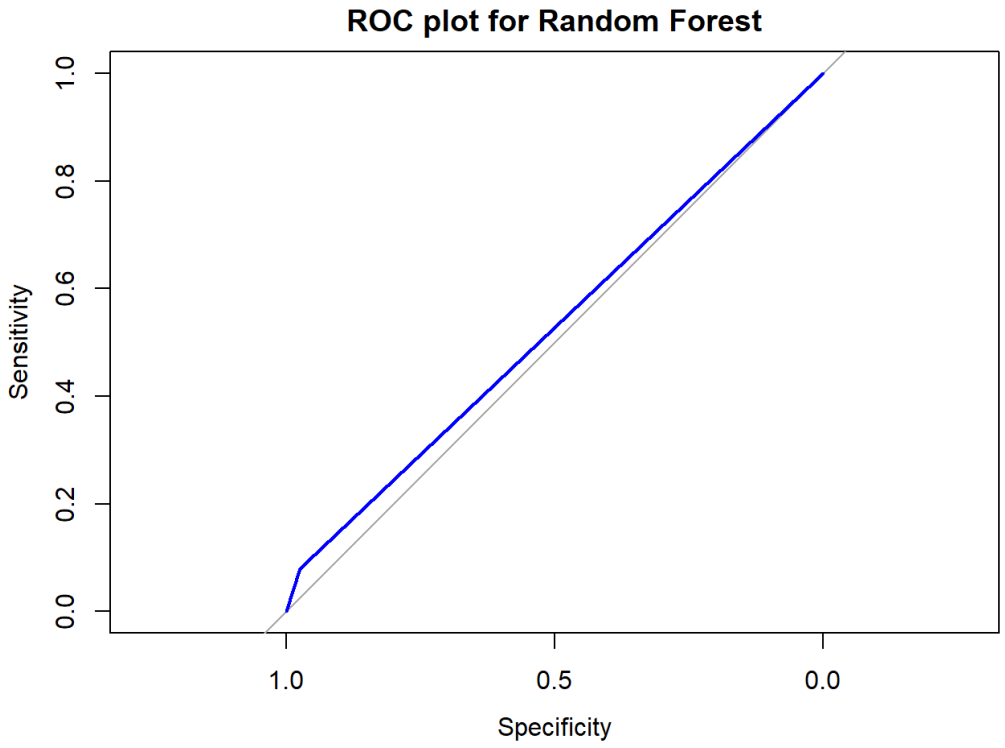

Fig 9

Neural Networks

Neural nets are a means of doing machine learning, in which a computer learns to perform some task by analyzing training examples. Usually, the examples have been hand-labeled in advance. An object recognition system, for instance, might be fed thousands of labeled images of cars, houses, coffee cups, and so on, and it would find visual patterns in the images

that consistently correlate with particular labels. Modeled loosely on the human brain, a neural net consists of thousands or even millions of simple processing nodes that are densely interconnected. Most of today's neural nets are organized into layers of nodes, and they're "feed-forward," meaning that data moves through them in only one direction. An individual node might be connected to several nodes in the layer beneath it, from which it receives data, and several nodes in the layer above it, to which it sends data.

This classifier (Table 7) achieved an accuracy of 78% in our case.

Table 7: Neural Networks

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
NN	0.56	0.78	0.83	0.71	0.29	0.17

Table 7

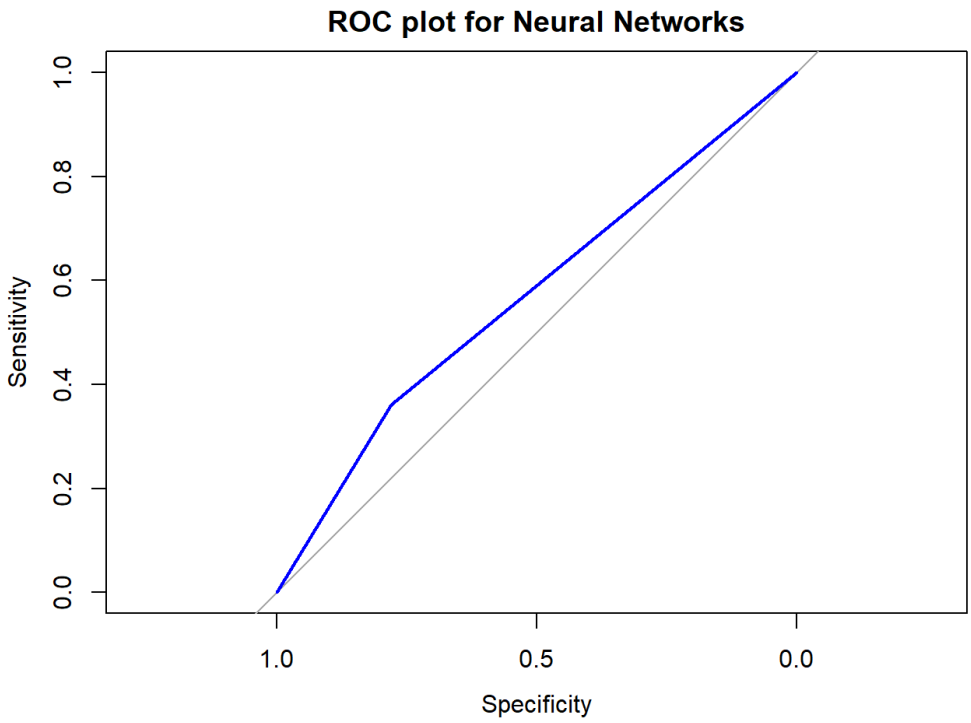

Fig 10

AutoML

Automated machine learning, also referred to as automated ML or AutoML, is the process of automating the time consuming, iterative tasks of machine learning model development.

Traditional machine learning model development is resource-intensive, requiring significant domain knowledge and time to produce and compare dozens of models. With automated machine learning, we will be able to accelerate the time it takes to get production-ready ML models with great ease and efficiency.

R does not have packages that do autoML. For this we used the help of H2O, a framework that integrate with R to give it the ability to do autoML.

Another one we used that support autoML is Azure autoML that runs on the cloud and gives us the ability to do that through the code in R with the help of AzureSDK for R. But for this, we choose to do it through the autoML Azure User interface.

H2O

AutoML with H2O in R, starts with initiating an h2o cluster. H2o has an easy autoML function that does the job. Running it against our data yields an AUC of 64%, returned by the most top algorithm used, StackedEnsemble in this case. The following (Output 1) shows the top models returned.

Output 1: AutoML with H2O

	model_id <chr>	auc <dbl>	logloss <dbl>	aucpr <dbl>	mean_per_class_error <dbl>
1	StackedEnsemble_AllModels_AutoML_20201130_123901	0.6451221	0.2897707	0.1591801	0.4020236
2	StackedEnsemble_BestOfFamily_AutoML_20201130_123901	0.6433244	0.2902958	0.1573896	0.4054043
3	GBM_1_AutoML_20201130_123901	0.6388830	0.2911247	0.1532442	0.4069578
4	GBM_2_AutoML_20201130_123901	0.6340468	0.2924717	0.1503562	0.4093414
5	GLM_1_AutoML_20201130_123901	0.6333360	0.2917587	0.1483458	0.4053267
6	GBM_3_AutoML_20201130_123901	0.6316216	0.2923413	0.1503605	0.4122483

Output 1

We also get the **ROC** curve as shown here (Fig 11).

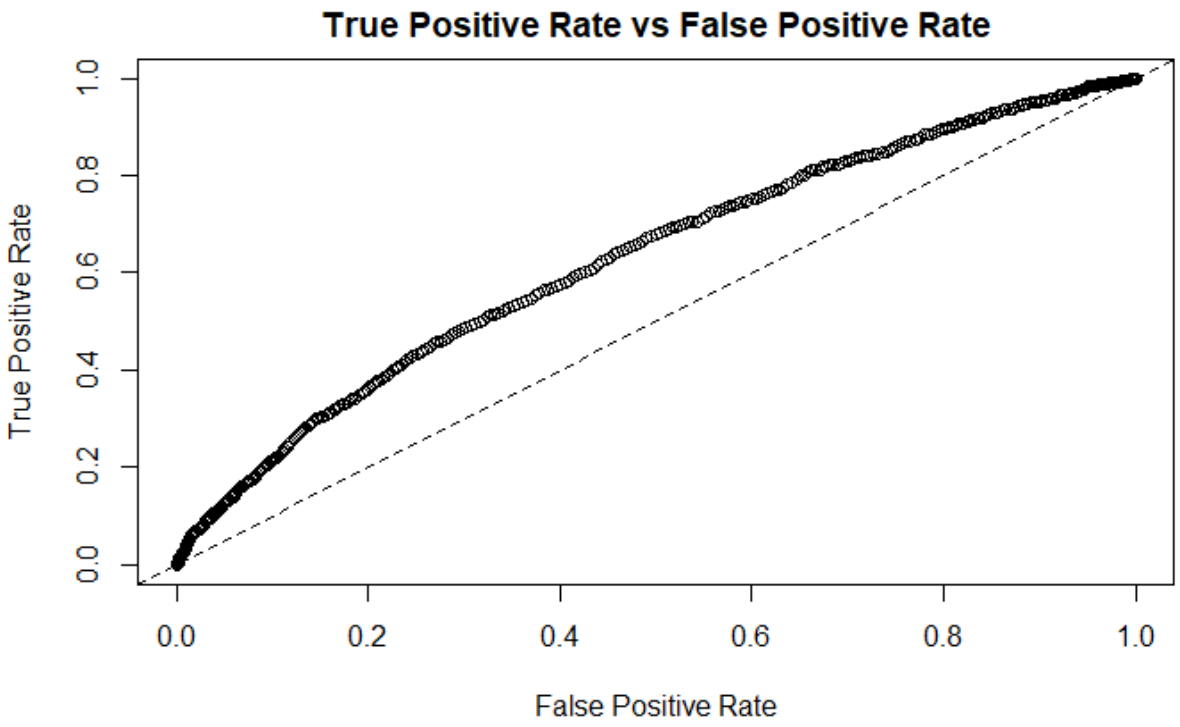

Fig 11

Because our top performing algorithm (StackedEnsemble) family does not currently have a meta-learner, we use the second top family top algorithm **GBM** to find out the features that return the best performance, as shown in the following plot (Fig 12).

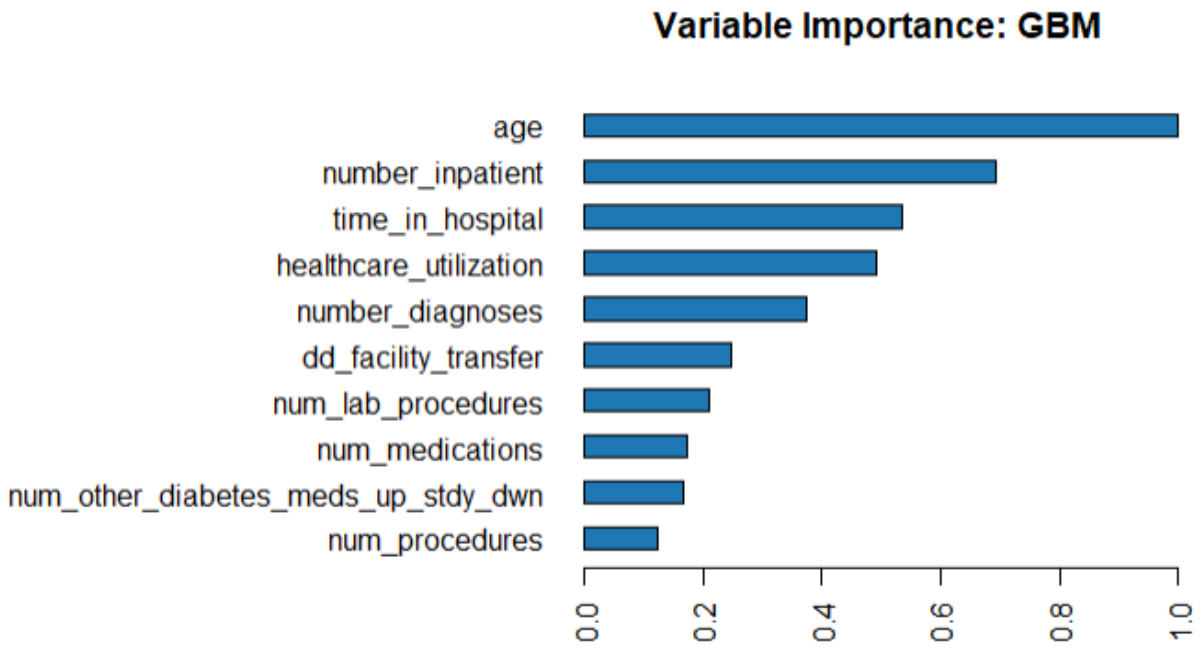

Fig 12

Azure AutoML

Microsoft Azure Machine Learning is a service on the cloud. Azure Machine Learning allows us to train and tune a model using the target metric we specify. In our study of modeling Hospital readmission with Azure AutoML, we were able to get an accuracy of 91% with an AUC of 62%. We found that this is the best model we could came up with (Fig 13).

Fig 13: Azure AutoML - Best Model

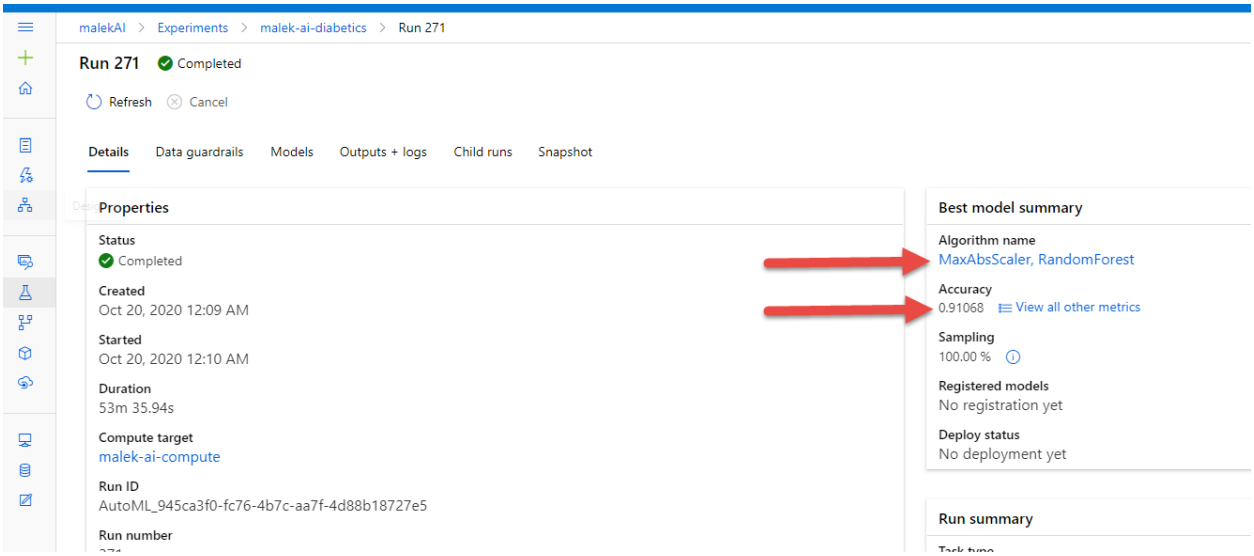

Fig 13

Azure AutoML, also offers us an explain-ability graphic (Fig 14), showing all the features that were involved in explaining this model.

Fig 14: Azure AutoML - Important Features

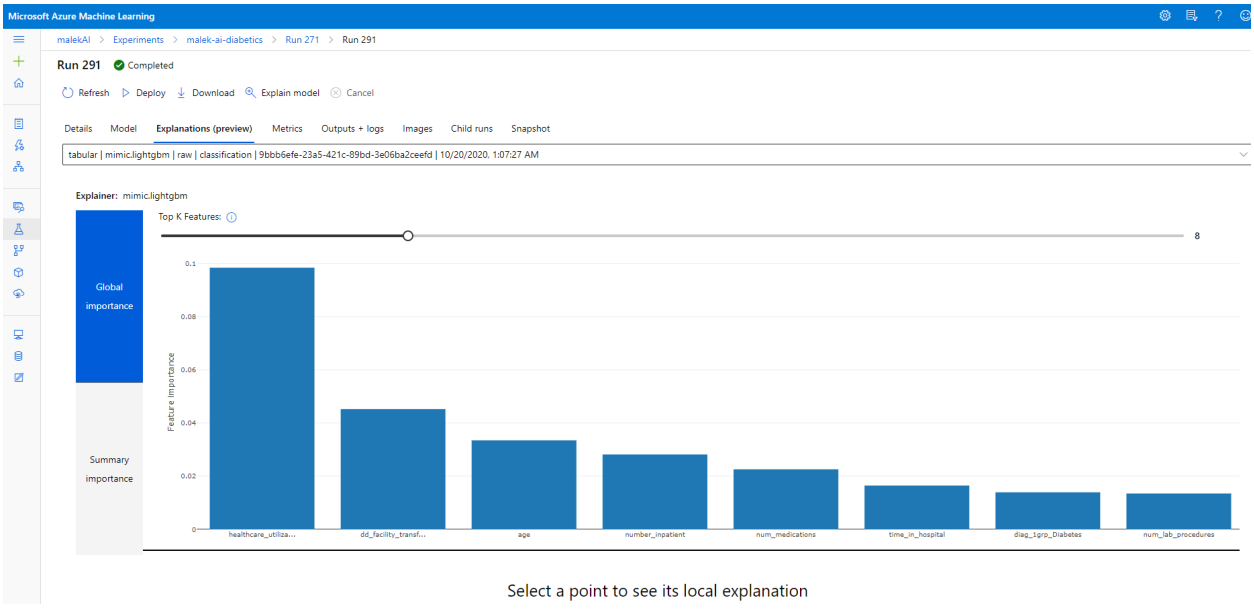

Fig 14

Azure autoML also offers the display of accuracy (Table 8) and ROC (Fig 15) in a friendly user interface.

Table 8: Accuracy & AUC for Azure autoML experience

Metrics | Outputs + logs | Images | Child runs | Snapshot

View as: (•) Chart ( ) Table

accuracy	weighted_accuracy	AUC_macro	AUC_weighted	recall_score_weighted
0.9106787844362397	0.9904715813536897	0.6259827947851356	0.6259827947851356	0.9106787844362397

Table 8

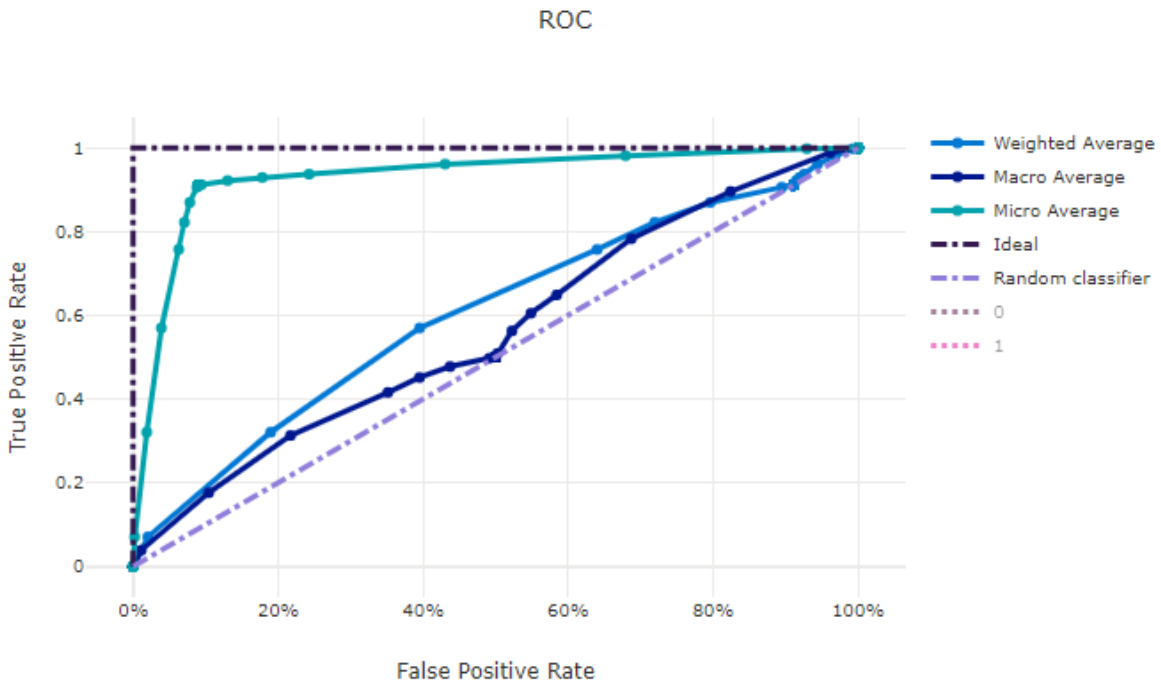

Fig 15

Conclusion

The following table (Table 9) shows the comparison across all 6-classification models and summarizes the results for all their metrics. These are the metrics of the weak learners (base algorithms).

Table 9: Summary of weak learners with their metrics

ALGORITHM	AUC	ACCURACY	TPR(Sensitivity)	FPR(Specificity)	TNR	FNR
LR	0.62	0.76	0.8	0.66	0.34	0.2
NB	0.56	0.55	0.55	0.43	0.57	0.45
KNN	0.52	0.81	0.87	0.82	0.18	0.13
SVM	0.57	0.76	0.81	0.67	0.33	0.19
DT	0.54	0.86	0.92	0.84	0.16	0.08
RF	0.53	0.89	0.97	0.92	0.08	0.03
NN	0.57	0.74	0.78	0.64	0.36	0.22

Table 9

Based on table 9, we found that Random Forest is the best model with highest accuracy of 89% and sensitivity of 97%, but low AUC of 53%, from weak learner's perspective.

We introduced 2 autoML frameworks, one with h2o and another one with Azure. AutoML takes advantage of running all the known algorithms in parallel while tuning the parameters to achieve high performance.

We found out that the best performance was returned by Azure with MaxAbsScaler-RandomForest being the best model that returns the best performance with 91% accuracy and a higher AUC of 62%. We also found that the top 8 variables found to be relevant for this

performance in order of importance are: Healthcare_Utilization, dd_facility_transfer, age, number_inpatient, num_medications, time_in_hospital, diag_1grp_Diabetes and num_lab_procedures. Consequently, the Classifier utilized the remaining set of variables.

Much of the readmission literatures have primarily used weak learners (default models) to estimate patients' readmission probability. As we report in this research, all weak learners returned rather an acceptable accuracy (around 70%), except Naïve Bayes. In addition Random Forest is the best model from the 7 that returned an accuracy of 89% and a sensitivity of 97%, because in each iteration, this intelligible model tries a new set of features, taking advantage on the variability of features.

This research extends previous researches by using autoML to achieve the highest accuracy there is, for predicting readmission probability.

References

[1] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," *BioMed Research International*, 03-Apr-2014. [Online]. Available: https://www.hindawi.com/journals/bmri/2014/781670/. [Accessed: 21-Oct-2020].

[2] Sarthak, S. Shukla, and S. P. Tripathi, "EmbPred30: Assessing 30-Days Readmission for Diabetic Patients Using Categorical Embeddings," *Smart Innovations in Communication and Computational Sciences Advances in Intelligent Systems and Computing*, pp. 81–90, 2020. [Online]. Available: https://arxiv.org/pdf/2002.11215.pdf. [Accessed: 21-Oct-2020].

[3] O. Ben-Assuli, R. Padman, M. Leshno, and I. Shabtai, “Analyzing Hospital Readmissions Using Creatinine Results for Patients with Many Visits,” *Procedia Computer Science*, 21-Sep-2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916321998?via=ihub. [Accessed: 21-Oct-2020].

[4] R. Joshi and M. Alehegn, “Analysis and prediction of diabetes diseases using machine learning algorithm: Ensemble approach,” *International Research Journal of Engineering and Technology (IRJET)*, Oct. 2017. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/54976577/IRJET-V4I1077.pdf?1510396014=&response-content-disposition=inline%3B+filename%3DAnalysis_and_prediction_of_diabetes_dise.pdf&Expires=1603304802&Signature=eA5hHU8cZgbcv1Lg43~IXVLeIkNTfTXtfbDrGeRM6x7pQ9PH3viAw5AE7sSaV2hepMzkJtt3uIGLnWyr8aqKEv9VQFu0oNExGmRCIelVVFOp36UYB~1Kp-czNjY1k25X6eses846ByctxiHrgwOdqBL4ZGDFPE6xjOWWqBc3lmR69vNBRNUQGeaHy3Jm~12yMzZnsebERsLqWRdIs-6Uc87w-iOUapLKNRcI0OuA65xgXKvyUDuHAV8yhCGRGhkNRB6iYgjRo-JJwZP0YQXize2Yt7XR7~fYxOtI6LjrIy-Zth4ZdYsxpeBasIbybcgosi32~WSQMbyZBjKdLNXLKQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA].

[5] N. Nai-Arun and R. Moungmai, “Comparison of Classifiers for the Risk of Diabetes Prediction,” *Procedia Computer Science*, 14-Nov-2015. [Online]. Available:

https://www.sciencedirect.com/science/article/pii/S1877050915031786. [Accessed: 21-Oct-2020].

[6] A. Hammoudeh, G. Al-Naymat, I. Ghannam, and N. Obied, "Predicting Hospital Readmission among Diabetics using Deep Learning," *Procedia Computer Science*, vol. 141, pp. 484–489, 2018. [Online]. Available: https://switchpointventures.com/wp-content/uploads/2018.11_predicting-hospital-readmission-among-diabetics-using-deep-learning.pdf. [Accessed: 21-Oct-2020].

[7] A. Agarwal, C. Baechle, R. S. Behara, and V. Rao, "Multi-method approach to wellness predictive modeling," *Journal of Big Data*, 01-Jan-1970. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0049-0. [Accessed: 21-Oct-2020].

[8] S. Salian and D. G. Harisekaran, "Big Data Analytics Predicting Risk of Readmissions of Diabetic Patients," *International Journal of Science and Research (IJSR)*, Apr. 2015. [Online]. Available: https://www.ijsr.net/archive/v4i4/SUB152923.pdf. [Accessed: 21-Oct-2020].

[9] R. S. Behara, A. Agarwal, V. Rao, and C. Baechle, "Predicting the occurrence of diabetes using analytics." [Online]. Available: http://faculty.eng.fau.edu/ankur/files/2017/08/Predicting-the-Occurence-of-Diabetes-Using-Analytics.pdf. [Accessed: 21-Oct-2020].

[10] T. Goudjerkan and M. Jayabalan, "Predicting 30-day hospital readmission for diabetes patients using multilayer perceptron." [Online]. Available: https://thesai.org/Downloads/Volume10No2/Paper_36-

Predicting_30_Day_Hospital_Readmission_for_Diabetes_Patients.pdf. [Accessed: 21-Oct-2020].

[11] N. M. S. kumar, T. Eswari, P. Sampath, and S. Lavanya, "Predictive Methodology for Diabetic Data Analysis in Big Data," *Procedia Computer Science*, 08-May-2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915005700?via=ihub. [Accessed: 21-Oct-2020].

[12] C. Baechle and A. Agarwal, "A framework for the estimation and reduction of hospital readmission penalties using predictive analytics," *Researchgate*, Dec-2017. [Online]. Available: https://www.researchgate.net/publication/320818442_A_framework_for_the_estimation_and_reduction_of_hospital_readmission_penalties_using_predictive_analytics. [Accessed: 21-Oct-2020].