An analysis of the Impact of Hemoglobin Measurements on Hospital Readmission Rates

Paul Kiyambu Mvula 300169684
School of Electrical Engineering And Computer Science
University of Ottawa
Ontario, Canada
pmvul089@uottaawa.ca

Abstract*—*The object of this project is to analyze health care data by looking at the impact of Haemoglobin (HbA1c) measurements on hospital re-admission rates. This project makes use of both supervised and semi-supervised learning algorithms in Machine Learning to classify whether the patient is readmitted within or after 30 days or not. The data set obtained from the University of California, Irvine, Machine Learning Repository.

I. Introduction

The term HbA1c refers to glycated haemoglobin, a form of haemoglobin that is chemically linked to sugar. It is made when the glucose (sugar) in a person's body sticks to their red blood cells [1]. An haemoglobin (A1c) test tells what the average level of blood sugar of a person has been over a period of weeks/months. This test may be used to help monitor people with diabetes' condition and glucose levels, or check for prediabetes in adults whose blood levels show that they are at risk of getting diabetes. This test is important, especially for people with diabetes, because the higher it is, the greater the danger of developing diabetes-related inconveniences. The World Health Organization defines diabetes as a metabolic disorder with various causes characterized by chronic hyperglycemia and disturbances of carbohydrate, fat and protein metabolism resulting from defects in insulin secretion, insulin action, or both [2]. People with diabetes are at increased risk of cardiac, peripheral arterial and cerebrovascular disease [3].

According to the Centers for Disease Control and Prevention (CDC) [4], a normal A1c level is below 5.7%, 5.7% to 6.4% indicates prediabetes and a level of 6.5 or above indicates diabetes. The different levels of HbA1c can be found in Table I. In addition, Fig. 1 shows the comparison between the blood glucose (in mmol/L) and the HbA1c levels (in %) test results according to the global diabetes community in the United Kingdom [1]. In this data set, as the records are old, the normal A1c level is less than 7%, prediabetes when the level is between 7% and 8% and diabetes when it is above 8%.

TABLE I. Levels of HbA1c (%)

HbA1c	%
Normal	Below 5.7%
Prediabetes	5.7% to 6.4%
Diabetes	6.5% or above

In this project, we use the data set representing 10 years of health care information about patients with diabetes from 130 hospitals in the United States and integrated delivery networks from the UCI Machine Learning Repository [5]. A description of the data set and the missing values will be given in the coming section. For the supervised learning approach, we have applied five different classification algorithms: Decision Tree (DT) to help visualize the structure and observe the features with high importance, the Extreme Gradient Boosting (XGBoost), the Random Forests (RF), the Logistic Regression (LR) and the Support Vector Machine (SVM). For the semi-supervised learning approach, we have used self-training with decision tree, then label-propagation, and label-spreading. As this project is medical domain related, we will care more about the sensitivity (True Positive Rate) and specificity (True Negative Rate), further details on these metrics and those used for evaluating the algorithms' performance are given in the section IV.

The rest of the paper is organized as follows: in the first following section, description of the data set used in the project, the methods for handling the missing values and data preprocessing steps are

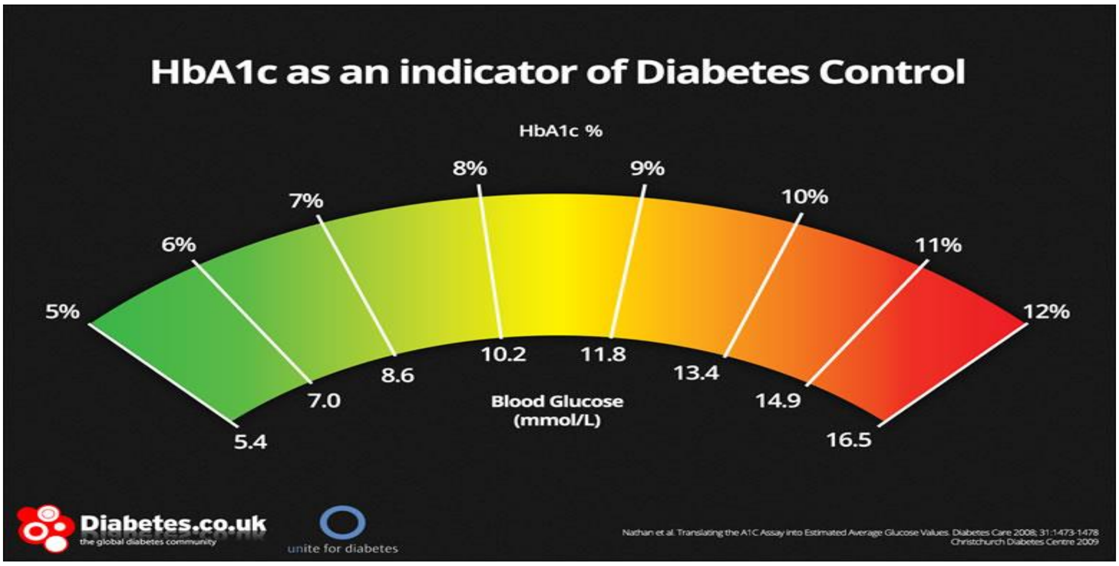

Fig. 1. HbA1c/Blood Glucose comparison

given, the experimental setups are described in Section III. After the comparative experiments, the results are depicted in Section IV. Finally, Section V provides conclusion and future works, VI, acknowledgements followed by references.

II. Data Set

The data set used in this project “Diabetes 130-US hospitals for years 1999-2008” [6], contains information about patients from 130 US hospitals in the USA for 10 years. This data set has been proposed by Beata et al. [7] to demonstrate that the decision to obtain HbA1c measurements for patients with diabetes is a useful predictor of readmission rates. It contains 101766 encounters of 71518 unique patients. For each patient, we have only kept one record, specifically the record resulting in a readmission (if available). Each record contains 50 features, and the percentage of missing values per feature is shown along with the description of each feature [7], namely:

1. **Encounter ID** *Numeric* Unique identifier of an encounter 0%
2. **Patient number** *Numeric* Unique identifier of a patient 0%
3. **Race** *Nominal* Values: Caucasian, Asian, African American, Hispanic, and other 2.63%
4. **Gender** *Nominal* Values: male, female, and unknown/invalid 0%
5. **Age** *Nominal* Grouped in 10-year intervals: [0, 10), [10, 20), . . ., [90, 100) 0%
6. **Weight** *Numeric* Weight in pounds. 96%
7. **Admission type** *Nominal* Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available 0%
8. **Discharge disposition** *Nominal* Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available 0%
9. **Admission source** *Nominal* Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital 0%
10. **Time in hospital** *Numeric* Integer number of days between admission and discharge 0%
11. **Payer code** *Nominal* Integer identifier corresponding to 23 distinct values, for example, Blue Cross\Blue Shield, Medicare, and self-pay 42.32%
12. **Medical specialty** *Nominal* Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family\general practice, and surgeon 48.34%
13. **Number of lab procedures** *Numeric* Number of lab tests performed during the encounter 0%
14. **Number of procedures** *Numeric* Number of procedures (other than lab tests) performed during the encounter 0%

15. **Number of medications** *Numeric* Number of distinct generic names administered during the encounter 0%
16. **Number of out-patient visits** *Numeric* Number of outpatient visits of the patient in the year preceding the encounter 0%
17. **Number of emergency visits** *Numeric* Number of emergency visits of the patient in the year preceding the encounter 0%
18. **Number of inpatient visits** *Numeric* Number of inpatient visits of the patient in the year preceding the encounter 0%
19. **Diagnosis 1** *Nominal* The primary diagnosis (coded as first three digits of ICD9); 848 distinct values 0.02%
20. **Diagnosis 2** *Nominal* Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values 0.40%
21. **Diagnosis 3** *Nominal* Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values 1.60%
22. **Number of diagnoses** *Numeric* Number of diagnoses entered to the system 0%
23. Glucose serum test result *Nominal* Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured 0%
24. **A1c test** result *Nominal* Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured. 0%
25. **Change of medications** *Nominal* Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change” 0%
26. **Diabetes medications** *Nominal* Indicates if there was any diabetic medication prescribed. Values: “yes” and “no” 0%
27. **24 features for medications** *Nominal* For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed 0%
28. **Readmitted** *Nominal* Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission.

A. *Handling missing values*

The data set exploration was carried out after removing the duplicate records of some patients and some columns have been found to contain missing values. Those columns are the race (2.63%), the weight (96%), the payer code (42.32%), the medical specialty (48.34%), primary diagnosis (0.02%), secondary diagnosis (0.40%) and third diagnosis (1.60%). They have been handled in different ways. They can be considered as Missing Completely At Random (MCAR) values.

1) Race: there were 1886 records, 2.63%, missing the race attribute in the original data set, we have imputed using the most frequent strategy, therefore the missing values have been set to “Caucasian” because that is the most frequent race in the data set.

2) Weight: more than 68666 records, 96% of the data set, had weight missing, as explained in [7], in 2009, hospitals and clinics were not required to capture the weight in a structure format. Although removal is not an optimal option to consider, because it leads to the loss of information, the weight feature has been removed because imputing would lead to overfitting as the amount of missing values was too large.

3) Payer code: similar to the weight, the payer code has too many missing values to consider imputing, furthermore, it is irrelevant to the outcome, just like the encounter ID and the patient ID, as explained in the next sub-section. This feature has therefore been removed.

4) Medical specialty: 34572 missing values for the specialty of the admitting phisician, the missing values have been set to “Other” to specify that the patient was admitted by a department that was not listed. In total, 73 departments are found in the data set and some are sub-branches of others. The sub-branches have been grouped with the parent branch and the number of departments has been reduced to 42, and converted to numeric values with the label encoder.

5) Primary, secondary and third diagnosis: similar to the race, the missing values in these columns have been imputed with the most recent imputer.

After these modifications, the data set did not contain any missing values and the number of columns has been reduced to 48.

B. Data Preprocessing

After handling the missing values, some feature transformation techniques have been applied to some columns in the data set, and some features have been dropped as they either contained too many missing values or were deemed irrelevant for this study.

1) Encouter ID: this feature has been dropped as it does not give any meaningful information to the study.

2) Patient Number: the data set contains multiple encounters for some patients at either the same or different hospitals and most of them have different outcomes with regards to the readmission feature. We decided to keep those records of which the outcomes are the readmission of a patient or if both records have a readmission outcome, we only kept those readmitted in less than 30 days.

3) Gender: as decribed above, the gender is a nominal feature containing three values, respectively: female, male and unknown/invalid. In the original data set, there are 54708 female, 47055 male, and 3 unknown/invalid patients. The unknown/invalid gender has been removed because it contained too few samples. After that it has been converted to numeric with the label encoder.

4) Number of lab procedures & Number of medication, …: the box and whisker plot has been used to visualize the distribution of the outliers. The outliers were kept in the data set to avoid loss of information.

5) Discharge disposition: we have removed the disposition that resulted in death of a patient or those that were invalid or were not mapped.

The remaining nominal features have all been encoded with the label encoder to numeric values as some of the algorithms used in this study do not take nomial features.

C. Conclusion

After the steps mentioned above, we remained with 66854 records, 50441 that resulted in no readmission and 16414 that resulted in a readmission of a patient within less than or more than 30 days as the two outcomes have been merged to one. 35604 female and 31251 male patient records. It can also be noticed that more than 55000 patients did not have their HbA1c measured and was set to “None”, that category has been left untouched as imputing could lead to adding unrealistic values to those patients. As this data set is imbalanced, we have applied a sampling method called NearMiss [15] to under-sample the larger class (No readmission) and have a balanced distribution. After NearMiss, we remained with 16414 records per class.

III. Experimental setup

For the supervised learning approach, in total, 5 algorithms have been applied: the Decision Tree Classifier, the Logistic Regression, the Random Forest Classifier, the Support Vector Machine and the Extreme Gradient Boosting [8]. Except from the Extreme Gradient Boosting algorithm which has been developed in C++ and downloaded separately to use in the Python Jupyter NoteBook, all the algorithms have been implemented the scikit-learn library [9] in python. As for the unsupervised learning approach, we have used self-training with a decision tree as the base learner, label propagation and label spreading.

The Decision Tree Classifier is among the popular and powerful algorithms. It helps select appropriate features for splitting the tree into subparts and getting the target item. At the bottom of the tree, each leaf is assigned to a class. Decision trees can be used for classification (classification trees) as well as regression (regression trees). For this project, the parameter “max depth” which sets the maximum depth of the tree has been tuned to 13.

The Random Forests are ensemble algorithms consisting of several Decision Trees which are used as voters. The algorithm was developed by Breiman and Cutler [10]. We have chosen Random forests because of their ability to attain high accuracies and handle outliers and noise in the data. In RF, each tree is trained with a random sub-sample drawn from the main training set. The term “ensemble” refers to the grouping of two or more weak learners or weak learning algorithms to improve the performance of the performance of a single classifier [11].

The Support Vector Classifier is also an important algorithm used in Machine Learning. It has been proposed by [16]. It is used for both classification and regression. It is a geometric model that finds hyperplanes to create boundaries between classes. By using kernel functions and the kernel trick, which calculates the high dimensional relationships without actually transforming the data in a higher dimension, it reduces the amount of computation and makes the infinite dimensions used by kernel functions such as the Radial Kernel, which states that the similarity between two points in the transformed feature space is an exponentially decaying function of the distance between the vectors and the original input space as shown in (1), possible.

$$K(x,x') = exp\ (-\gamma||x-x'||^2) \tag{1}$$

The Extreme Gradient Boosting is also an ensemble method but differs from the RF in the way that the decision trees are built. They are built sequentially; previous weights are fed to the next decision tree until a termination criterium is met. It can be used for both classification and regression. Among the algorithm's features, the regularized boosting which prevents overfitting, the ability to handle missing values automatically and the tree pruning which results in optimized trees are what make the algorithm very powerful.

The Logistic Regression is a supervised classification algorithm. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression) [12]. It models the data using the sigmoid function (2) shown in Fig. 2.

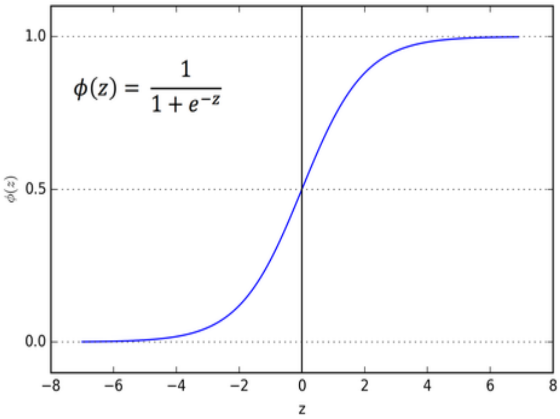

Fig. 2. Logistic Regression Sigmoid function

$$g(z) = \frac{1}{1+e^{-z}} \tag{2}$$

Self-training is a form of semi-supervised learning approach where the classifier is trained on a small amount of labeled data and then it is used to make predictions of the unlabeled data, those predictions, called pseudo-labels, are then adopted in subsequent iterations of the classifier. The workflow of the self-training algorithm is shown in Fig. 3. In this project, we have used the decision tree as the base classifier; the pseudo-labels are used in an incremental method; the most informative examples are added to the train set only if they have a prediction probability greater than 99%, this constitutes our selection criterion. We stop iterating when there are no more informative examples in the unlabeled data set or when all the instances in the unlabeled set have been labeled, this make up our stopping criterion.

Label propagation is a family of semi-supervised algorithms that rely on the graph representation of a dataset to exploit the existing relations among nodes to propagate the labels to unlabeled points [14], it uses the raw similarity matrix constructed from the data with no modifications. Label spreading minimizes the loss function that has regularization properties which makes it more robust to noise; it iterates on a modified version of the original graph and normalizes the edge weights with the use of Laplacian Matrix, Fig. 4 shows an illustration of the label spreading algorithm. They have both been developed in sci-kit learn [13]. For both algorithms, we have used the two kernel functions: Radial Basis Function and K-nearest neighbor kernel.

IV. EXPERIMENTAL RESULTS

For the supervised learning approach, the train and test have been split with scikit-learn built-in function. the train set consists of 66.66% of the total number of records under-sampled with NearMiss, and the test set consists of 33.33%. For the semi-supervised approach on the other hand, different levels of unlabeled data have been tested, 10%, 20%, 50%, 90% and 95%.

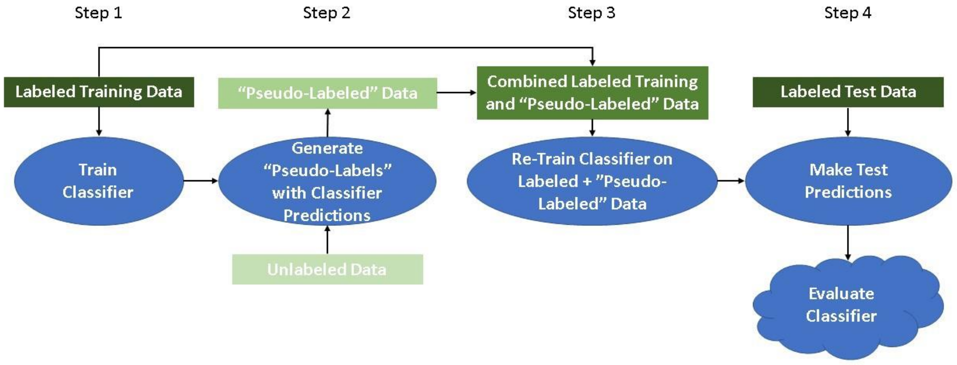
Step 1
Step 2
Step 3
Step 4
Labeled Training Data
"Pseudo-Labeled" Data
Combined Labeled Training and "Pseudo-Labeled" Data
Labeled Test Data
Train Classifier
Generate "Pseudo-Labels" with Classifier Predictions
Re-Train Classifier on Labeled + "Pseudo-Labeled" Data
Make Test Predictions
Unlabeled Data
Evaluate Classifier

Fig. 3. Self-training workflow

Experiments were carried out on a Hewlett Packard device with 2.6 GHz Intel Core i7 processor and 8 GB of 1600MHz DDR3 RAM. Each test is executed with the different algorithms trained with the corresponding training data set.

Frequently, to evaluate the performance of a model, a confusion matrix is constructed and from the values of the confusion matrix, metrics such as the accuracy, the f1 score, the specificity (True Negative rate, recall of the negative class), the sensitivity (True Positive Rate, recall of the positive class), …, can be calculated to evaluate the efficiency of an algorithms. Equations to calculate

$$Sensisivity\ (TPR) = \frac{TP}{TP\ +\ FN} \tag{3}$$

$$Specificity\ (TNR) = \frac{TN}{TN\ +\ FP} \tag{4}$$

$$F - Measure = 2 \times \frac{Precision\ \times\ Sensitivity}{Precision\ + Sensitivity} \tag{5}$$

$$Accuracy = \frac{TP\ +\ TN}{TP\ +\ TN\ +\ FP\ +\ FN} \tag{6}$$

the metrics are shown in (2-5). Where TP consists of the true positives, the patients predicted as readmission that are actually readmission, TN the true negatives, the records predicted not readmission that were actually not readmission, FP the false positives, the records predicted as readmission but are actually not readmission and FN the false negatives, the records predicted as not readmission but are actually readmission, produced by the algorithms. In addition to these metrics, we have plotted the Receiver Operating Characteristic (ROC) curve and calculated the area under the curve for each classifier.

A. *Task 1*

The first task consists in predicting outcomes, whether the patient is readmitted or not. Each classifier's performance is shown in Tables (II-IX)

1) *Decision Tree*

TABLE. II. Decision Tree performance

Metric	Score
Sensitivity	0.531659
Specificity	0.930437
F-1	0.623781
Accuracy	0.682536

TABLE. III. 10-fold cross-validation

Fold	Time	F-1	ACC.	TPR	TNR
1	0.3498	0.6813	0.7450	0.5453	0.9445
2	0.3288	0.6824	0.7340	0.5716	0.8964
3	0.4077	0.6463	0.7100	0.5301	0.8897
4	0.3208	0.6466	0.7060	0.5380	0.8739
5	0.4137	0.6291	0.6804	0.5420	0.8190
6	0.3238	0.5770	0.6454	0.4835	0.8074
7	0.3168	0.5976	0.6414	0.5322	0.7507
8	0.3198	0.5622	0.6177	0.4908	0.7446
9	0.3068	0.5625	0.5947	0.5210	0.6684
10	0.3148	0.6004	0.6176	0.5746	0.6605
AVG	0.3403	0.6185	0.6692	0.5329	0.8055
STD	0.0368	0.0430	0.0502	0.0280	0.0927

Evaluation: Apart for the high specificity, the decision tree does not have great results with other metrics. One way to improve the performance is to fine tune it with different parameter settings and compare the importance of each feature compare the results obtained from each method.

2) *Logistic Regresion*

TABLE. IV. Logistic Regression performance

Metric	Score
Sensitivity	0.611593
Specificity	0.669562
F-1	0.627699
Accuracy	0.640866

TABLE. V. 10-fold cross-validation

Fold	Time	F-1.	ACC	TPR	TNR
1	0.2838	0.6908	0.7191	0.6276	0.8105
2	0.2508	0.7101	0.7267	0.6697	0.7838
3	0.2398	0.6908	0.6966	0.6782	0.7149
4	0.2468	0.6198	0.6436	0.5813	0.7058
5	0.2218	0.5884	0.6122	0.5542	0.6703
6	0.2428	0.5749	0.5887	0.5560	0.6215
7	0.2498	0.5515	0.5839	0.5115	0.6563
8	0.2548	0.5896	0.5921	0.5858	0.5984
9	0.2154	0.5934	0.5895	0.5990	0.5801
10	0.2138	0.5822	0.5737	0.5941	0.5533
AVG	0.2420	0.6191	0.6326	0.5957	0.6695
STD	0.0199	0.0537	0.0567	0.0489	0.0808

Evaluation: The logistic regression constitutes an improvement as compared to the decision tree in term of sensitivity, and F-1, but the specificity and accuracy are lower. There is a trade-off between the different results. Looking at the fit time, we see that it is the smallest compared to the other algorithms.

3) *Extreme Gradient Boosting*

TABLE. VI. XGBoost performance

Metric	Score
Sensitivity	0.707956
Specificity	0.795331
F1	0.738707
Accuracy	0.752078

TABLE. VII. 10-fold cross-validation

Fold	Time	F-1.	ACC	TPR	TNR
1	2.5845	0.8001	0.8215	0.7148	0.9281
2	2.1647	0.8015	0.8227	0.7160	0.9293
3	2.7624	0.7807	0.8026	0.7032	0.9019
4	2.2936	0.7523	0.7791	0.6709	0.8873
5	2.3536	0.7145	0.7407	0.6485	0.8330
6	2.4286	0.7169	0.7368	0.6662	0.8074
7	2.5055	0.6851	0.7024	0.6473	0.7574
8	2.2976	0.6374	0.6442	0.6254	0.6630
9	2.3626	0.6434	0.6349	0.6587	0.6112
10	2.3366	0.6495	0.6301	0.6855	0.5746
AVG	2.4090	0.7181	0.7315	0.6737	0.7893
STD	0.1616	0.0605	0.0720	0.0290	0.1257

Evaluation: The overall performance of the XGBoost is great, this may be thanks to the ability to focus on the difficult to learn examples; the average fit time is larger than that of the logistic regression (almost ten times) and the decision tree because of number of trees that are being built sequentially.

4) *Random Forest Classifier*

TABLE. VIII. RFC performance

Metric	Score
Sensitivity	0.683034
Specificity	0.763119
F1	0.709764
Accuracy	0.723476

TABLE. IX. 10-fold cross-validation

Fold	Time	F-1	ACC.	TPR	TNR
1	5.5363	0.7626	0.7855	0.6892	0.8818
2	5.5927	0.7858	0.8044	0.7178	0.8909
3	5.4908	0.7652	0.7825	0.7093	0.8556
4	5.4268	0.7132	0.7426	0.6404	0.8447
5	5.5008	0.6888	0.7158	0.6291	0.8025
6	5.4878	0.6785	0.6926	0.6485	0.7367
7	5.4398	0.6568	0.6731	0.6254	0.7209
8	5.8026	0.6456	0.6466	0.6437	0.6496
9	5.4238	0.6353	0.6191	0.6636	0.5746
10	5.5098	0.6393	0.6166	0.6794	0.5539
AVG	5.5211	0.6971	0.7079	0.6646	0.7511
STD	0.1058	0.0537	0.0660	0.0311	0.1185

Evaluation: The overall performance of the Random Forest is also great, but it is not an improvement over the XGBoost as its average fit time is larger than that of all the previous algorithms.

5) *Support Vector Classifier*

TABLE. VIII. SVM performance

Metric	Score
Sensitivity	0.609193
Specificity	0.747918
F1	0.652818
Accuracy	0.679247

TABLE. IX. 10-fold cross-validation

Fold	Time	F-1	ACC.	TPR	TNR
1	196.84	0.6947	0.7282	0.6185	0.8380
2	201.34	0.7001	0.7267	0.6386	0.8148
3	198.13	0.6729	0.6911	0.6355	0.7466
4	194.32	0.6188	0.6484	0.5709	0.7259
5	187.87	0.5824	0.6165	0.5347	0.6983
6	188.97	0.5723	0.5967	0.5395	0.6538
7	189.86	0.5358	0.5836	0.4805	0.6867
8	188.13	0.5716	0.5927	0.5432	0.6422
9	185.48	0.5635	0.5786	0.5441	0.6130
10	191.43	0.5511	0.5667	0.5319	0.6014
AVG	192.25	0.6063	0.6329	0.5637	0.7021
STD	4.93	0.0583	0.0587	0.0489	0.0761

Evaluation: Although the support vector machine is the slowest of all the algorithms used in this project, its sensitivity and f-score are higher than that of the decision tree and its specificity, accuracy and f-score are higher than that of the Logistic Regression. It is still not an improvement over the XGB.

6) *ROC AND AUC*

Figure 5 shows the ROC curves of each algorithm along with the AUC. Again, the XGBoost has the highest Area Under the Curve (0.825). It is therefore selected as our best algorithm for this task.

B. *Task 2*

For the second task, we have chosen to predict the admission source. Before training the models, we applied the NearMiss under-sampling strategy to rebalance the data set. We have reduced the maximum number of samples per class to 2857. The classes with a lower number of records were left untouched. Table X shows the different accuracies score obtained with each classifier. It is

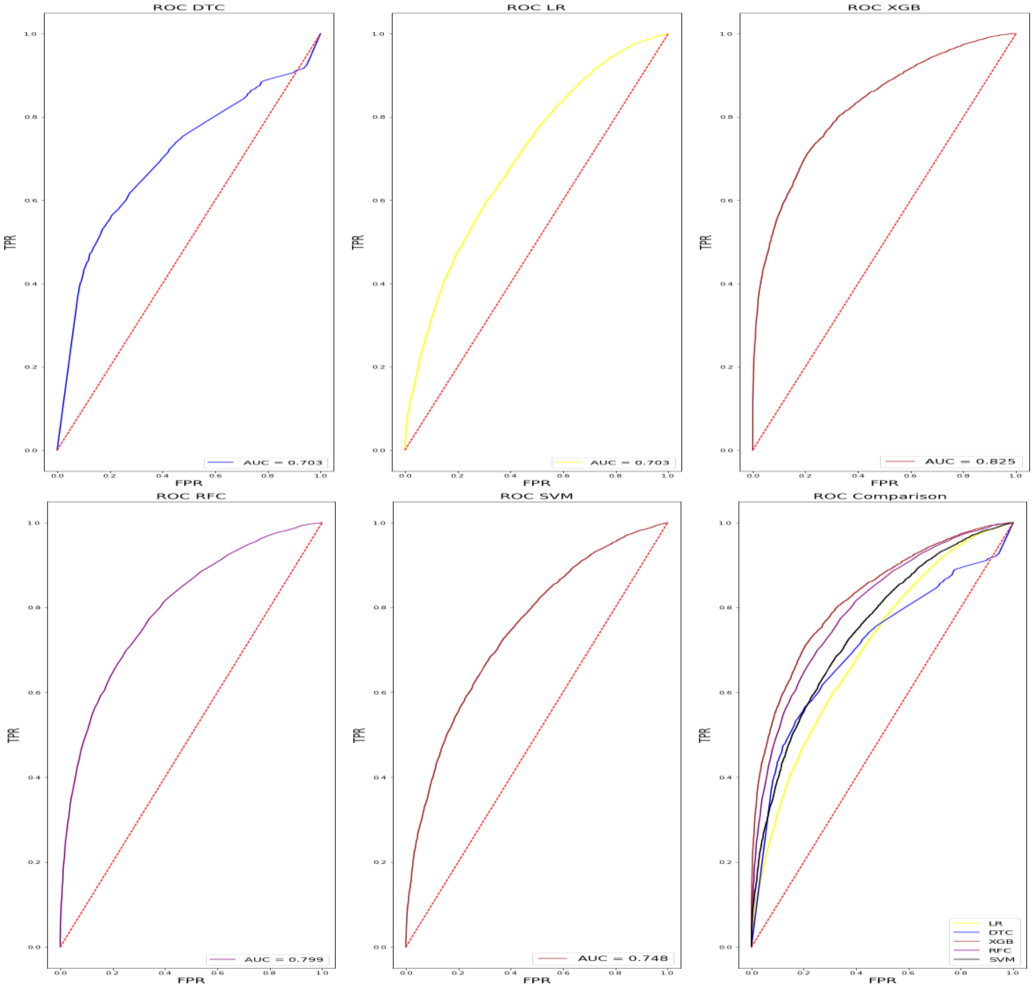

Fig. 5. Different ROC Curves and AUC

noticed that the XGBoost has the highest scores (accuracy, weighted- and macro-averaged F-1) compared to the other algorithms. We further checked the macro-averaged and weighted-average AUC score for these classifiers, shown in Table XI, XGBoost is still the best model in this scenario.

TABLE. X. Classifiers' accuracies

Model	Accuracy	Wgt F-1	Macro F-1
DT	0.7591	0.76	0.63
LR	0.6400	0.63	0.49
XGBoost	0.8140	0.81	0.67
RF	0.8129	0.81	0.62
SVM	0.7415	0.73	0.49

TABLE. XI. Macro- and weighted-average AUC scores

Model	Macro-avg	Weighted-avg
DT	0.780	0.814
Logit	0.876	0.891
XGBoost	0.959	0.971
RF	0.891	0.927
SVM	0.815	0.866

C. *Task 3*

In the third task, we implement three different semi-supervised learning models using three different approaches: self-training, label propagation and label spreading to predict the outcomes, that is whether the patient ends up being readmitted (in less/more than 30 days) or not, thus the primary data set is the same as the one used in Task 1.

1) *Self-training:* we have used the decision tree as the base learner of our self-training implementation. We have tested different levels of unlabeled data. The tables for the different levels of unlabeled data are showns below.

TABLE. XII. 0-10% unlabeled data performance

%	Iteration	F-1.	ACC	TPR	TNR
0	**0**	**0.62**	**0.68**	**0.52**	**0.83**
10	**0**	**0.61**	**0.67**	**0.52**	**0.81**
	1	0.61	0.67	0.53	0.81
	2	0.61	0.67	0.52	0.81
	3	0.61	0.67	0.53	0.81
	4	0.61	0.67	0.53	0.80
	5	0.61	0.67	0.53	0.81
	6	0.61	0.67	0.53	0.81
	7	**0.61**	**0.67**	**0.53**	**0.80**

TABLE. XIII. 20-95% unlabeled data performance

%	Iteration	F-1.	ACC	TPR	TNR
20	**0**	**0.62**	**0.67**	**0.55**	**0.79**
	1	0.61	0.67	0.53	0.80
	2	0.61	0.67	0.52	0.82
	3	0.61	0.67	0.51	0.82
	4	0.61	0.67	0.53	0.82
	5	0.62	0.67	0.53	0.81
	6	0.61	0.67	0.53	0.82
	7	**0.62**	**0.67**	**0.53**	**0.81**
50	**0**	**0.63**	**0.67**	**0.55**	**0.80**
	1	0.62	0.67	0.53	0.82
	2	0.63	0.67	0.56	0.79
	3	0.64	0.68	0.56	0.80
	4	0.64	0.67	0.57	0.78
	5	0.65	0.68	0.59	0.76
	6	0.64	0.68	0.56	0.79
	7	0.64	0.68	0.56	0.79
	8	0.64	0.67	0.56	0.79
	9	0.64	0.68	0.57	0.78
	10	0.63	0.67	0.56	0.78
	11	0.64	0.67	0.57	0.78
	12	0.64	0.68	0.58	0.78
	13	0.64	0.68	0.57	0.78
	14	**0.64**	**0.68**	**0.58**	**0.78**
90	**0**	**0.60**	**0.64**	**0.57**	**0.70**
	1	0.60	0.6	0.65	0.56
	2	0.57	0.58	0.61	0.54
	3	0.61	0.59	0.69	0.51
	4	0.60	0.59	0.66	0.52
	5	0.61	0.59	0.67	0.52
	6	0.61	0.59	0.68	0.51
	7	0.60	0.59	0.67	0.51
	8	**0.60**	**0.59**	**0.67**	**0.51**
95	0	**0.56**	**0.62**	**0.52**	**0.70**
	1	0.56	0.52	0.67	0.40
	2	0.57	0.52	0.70	0.38
	3	0.56	0.56	0.61	0.51
	4	0.56	0.56	0.62	0.51
	5	**0.56**	**0.56**	**0.61**	**0.51**

Evaluation: In bold are the first metrics: those obtained with without any pseudo-labels in the training set, and final metrics that were obtained after the pseudo-labels were added to the train set and the model was retrained then predicted on the test set. We can notice an improve in every metric when there is 50% unlabeled data. In conclusion, although some metrics values dropped below 50% while training, we can see that at the end, every metric is above the 50% threshold with the decision tree as base learner.

2) *Label-propagation:* we have used the default implementation of label propagation with the RBF kernel and KNN kernel; results are shown in Table XIV. The percentage column indicates the amount of labeled data. In addition, ROC curves are AUC

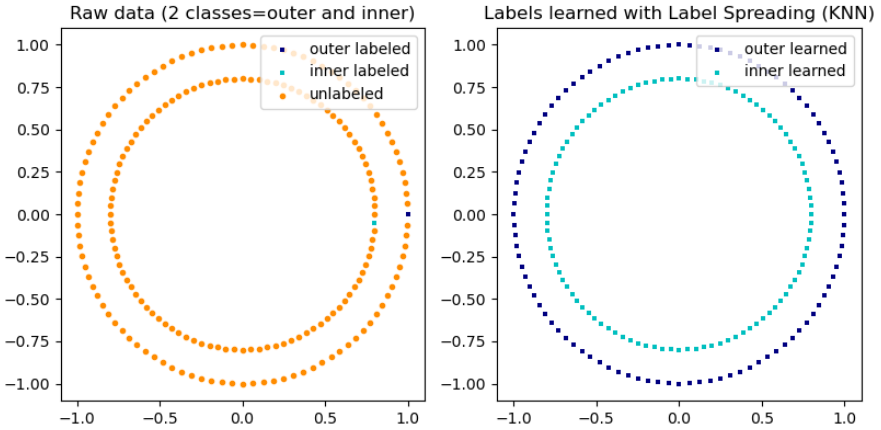

Fig. 4. An illustration of label spreading algorithm

for RBF and KNN kernels are shown in the with 100% labeled data are in the upper half of Fig. 6.

TABLE. XIV. Label propagation results

%	Kernel	F-1.	ACC	TPR	TNR	AUC
100	RBF	**0.131**	**0.535**	**0.071**	**0.986**	**0.689**
	KNN	**0.565**	**0.62**	**0.503**	**0.703**	**0.657**
90	RBF	0.100	0.528	0.053	0.989	0.689
	KNN	0.553	0.613	0.486	0.735	0.657
80	RBF	0.069	0.521	0.036	0.993	0.688
	KNN	0.551	0.608	0.487	0.726	0.654
50	RBF	0.015	0.509	0.008	0.995	0.689
	KNN	0.079	0.517	0.042	0.979	-
10	RBF	0.008	0.507	0.004	0.996	0.680
	KNN	0	0.507	0	1	-
5	RBF	0.004	0.507	0.002	0.998	0.676
	KNN	0	0.507	0	1	-

Evaluation: Poor results are obtained with the RBF kernel except from the increase in specificity with any amount of unlabeled data, on the other hand, the KNN kernel performed a little better than the RBF while the scores were still dropping in the upper half of the table but after the amount of unlabeled data reached more than 50%, all the scores dropped except from the specificity.

3) *Label-spreading:* similar to the label propagation, we have used the default implementation with RBF and KNN kernels. Results are shown in Table XV and ROC curves and AUC for RBF and KNN kernels with 100% labeled data are in the lower half of Fig. 6.

TABLE. XV. Label spreading results

%	Kernel	F-1.	ACC	TPR	TNR	AUC
100	RBF	**0.127**	**0.535**	**0.069**	**0.988**	**0.688**
	KNN	**0.564**	**0.618**	**0.501**	**0.732**	**0.663**
90	RBF	0.111	0.531	0.060	0.989	0.689
	KNN	0.555	0.613	0.490	0.733	0.660
80	RBF	0.089	0.525	0.047	0.990	0.688
	KNN	0.552	0.610	0.487	0.729	0.656
50	RBF	0.024	0.510	0.012	0.995	0.688
	KNN	0.535	0.602	0.464	0.736	0.647
10	RBF	0.016	0.508	0.008	0.995	0.673
	KNN	0.492	0.560	0.432	0.684	-
5	RBF	0.012	0.509	0.006	0.997	0.668
	KNN	0.468	0.554	0.398	0.706	-

Evaluation: Like label propagation, poor results are obtained with the RBF as kernel in label spreading, except from the increase in specificity at every decrease of labeled data in the train set. With KNN kernel, although the results keep dropping, they are not as low as the label propagation.

V. CONCLUSION AND FUTURE WORK

In this project, we have used set representing health care information about patients with diabetes to mainly predict whether the patient ends up being readmitted or not, thus making it a binary task. We applied supervised and semi-supervised learning algorithms. In the medical domain, we are interested in the sensitivity and specificity, therefore, in the supervised learning approach, the Extreme Gradient Boosting algorithm achieved the best results compared to the DT, LR and RFC with an average sensitivity of 67.37% and specificity of 78.93%. In the Decision Tree, we noticed that the number of in-patient visits played a key role in the classification. In the semi-supervised learning approach, results were not as conclusive as compared to the supervised approach.

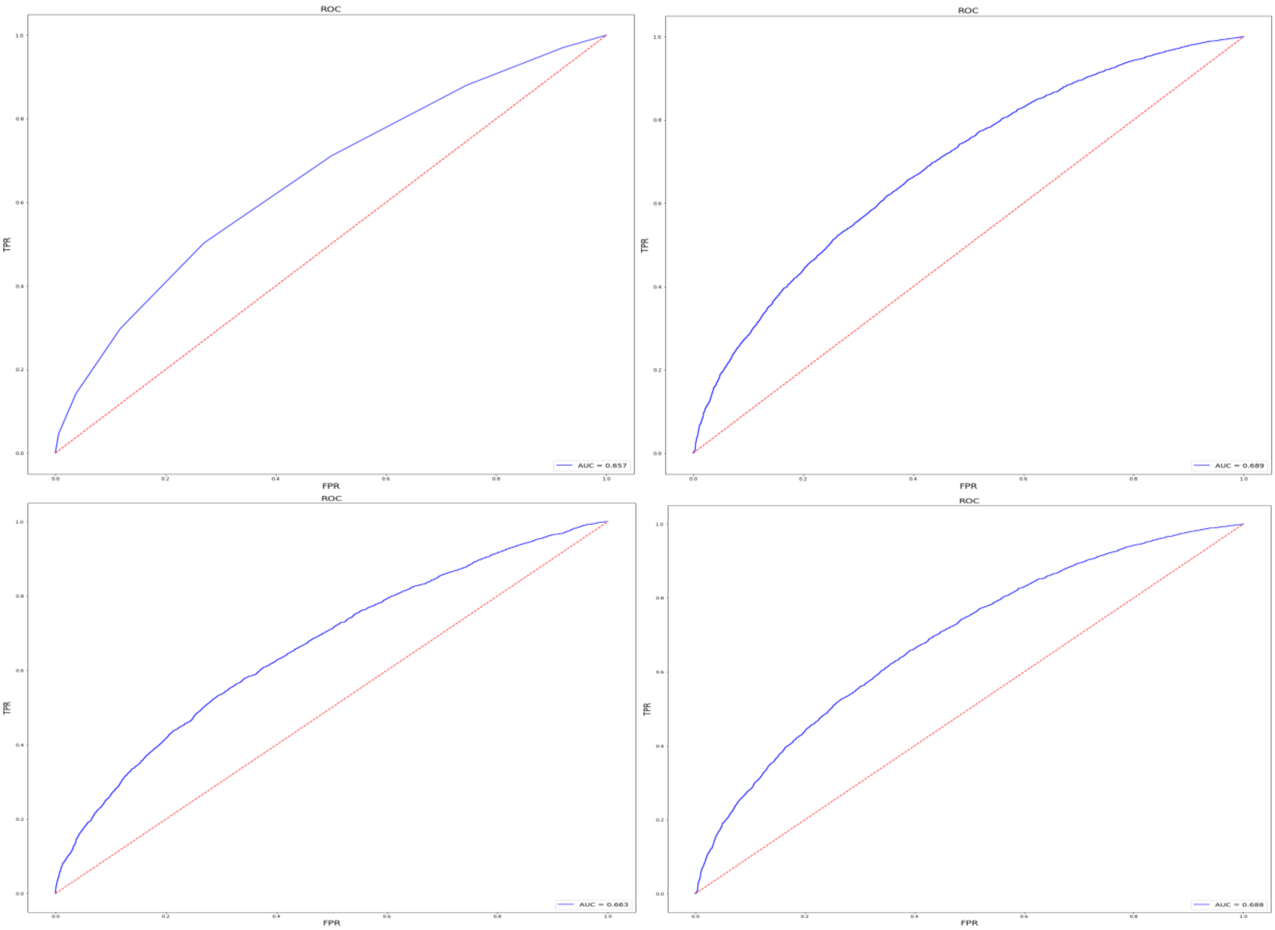

Fig. 6. Label spreading (upper-half) and Label propagation (lower-half) ROC curves

Future work will first be looking at the data set from a different angle to predict different outcomes and observing the impact of each feature in the predictions. Second, investigate more semi-supervised learning approaches to get better results as in the real world, we get are expected to have more unlabeled than labeled data and it is of the upmost importance not to misclassify the samples and keep high scores.

VI. ACKNOWLEDGMENTS

Through this course project, I obtained more experience with Machine Learning concepts and algorithms. It also gave me a deeper understanding of the course applications in the real world, especially in the medical domain.

Special thanks go to Dr. Herna L. Viktor for being so creative and passionate about her teaching, subject, and for communicating ideas and unique lessons with us.

REFERENCES

[1] What is haemoglobin ? The global diabetes community (UK) www.diabetes.org.uk

[2] World Health Organization. *Definition, Diagnosis and Classification of Diabetes Mellitus and its Complications. Part 1: Diagnosis and Classification of Diabetes Mellitus*. WHO/NCD/NCS/99.2 ed. Geneva, World Health Organization, 1999.

[3] Fox CS, Coady S, Sorlie PD et al. Increasing cardiovascular disease burden due to diabetes mellitus: The Framingham Heart Study. *Circulation, 2007, 115:1544-1550.*

[4] Breiman L, Friedman JH, Olshen RA and Stone CJ (1984). Classification and regression Trees. Wadsworth, Belmont

[5] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, 2010.

[6] Diabetes 130-US hospitals for years 1999-2008 Data Set, Available on: https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008

[7] Beata S., Jonathan PD, Chris G. et al. *Impact of HbA1c Measurement On Hospital Readmission Rates: Analysis Of 70000 Clinical Database Patient Records*, 2014

[8] Extreme Gradient Boosting (XGBoost) https://xgboost.readthedocs.io/en/latest/

[9] Scikit-learn https://scikit-learn.org/stable/

[10] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001

[11] Kittler, J., Hatef, M., Duin, R.P. W., & Matas, J.(1998). On Combining classifiers. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, 20(3), 226-239

[12] Tolles, Juliana; Meurer, William J (2016). "Logistic Regression Relating Patient Characteristics to Outcomes". *JAMA*. 316 (5): 533–4

[13] Semi-supervised https://scikit-learn.org/stable/modules/label_propagation.html

[14] Giuseppe B. "Mastering Machine Learning Algorithms - Second Edition". Jan.2020.

[15] NearMiss https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.under_sampling.NearMiss.html

[16] Vapnik, V. (1998). *Statistical learning theory*. New York: John Wiley.