



Independent project, 15 credits, for Bachelor of Science with a
major in computer science
Spring Semester 2019
Faculty of natural sciences

Artificial intelligence and Machine learning – a diabetic readmission study

Robin Forsman Jimmy Jönsson

Författare

Jimmy Jönsson & Robin Forsman

Titel

Artificiell intelligens och Maskininlärning - en undersökning om diabetisk återtagande

Engelsk titel

Artificial intelligence and Machine learning – a diabetic readmission study

Handledare

Niklas Gador

Examinator

Kamilla Klonowska

Sammanfattning

Utvecklingen av Artificiell intelligens ger stora möjligheter för sjukvården och skapar också nya utmaningar. För att Artificiell intelligens ska fungera så krävs det en noggrann analysering av den data man samlat in samt utförliga tester i olika algoritmer för att fastställa vilken algoritm som fungerar bäst med den data man samlat. I denna studie har det samlats data som innehåller patienter med diabetes som återinträtt och inte återinträtt till sjukhuset inom 30 dagar efter besök. Denna data har sedan undergått en noggrann analysering och jämförelse i olika algoritmer för att fastställa vilken algoritm som fungerar bäst och hur Artificiell intelligens kan förändra sjukvården idag baserat på resultaten.

Ämnesord

Artificiell intelligens, Maskininlärning, Logistisk regression, K-närmaste grannar, Artificiella neurala nätverk, Förstärkt beslutsträd.

Author

Jimmy Jönsson & Robin Forsman

Title

Artificial intelligence and Machine learning in health care

Supervisor

Niklas Gador

Examiner

Kamilla Klonowska

Abstract

The maturing of Artificial intelligence provides great opportunities for healthcare, but also comes with new challenges. For Artificial intelligence to be adequate a comprehensive analysis of the data is necessary along with testing the data in multiple algorithms to determine which algorithm is appropriate to use. In this study collection of data has been gathered that consists of patients who have either been readmitted or not readmitted to hospital within 30-days after being admitted. The data has then been analyzed and compared in different algorithms to determine the most appropriate algorithm to use.

Keywords

Artificial intelligence, Machine learning, Logistic regression, K-nearest neighbor, Artificial neural network, Boosted decision tree.

Table of Contents

LIST OF ABBREVIATIONS	6
1 Introduction	7
1.1 Background.....	7
1.2 Research questions	7
1.3 Problem.....	7
1.4 Purpose	8
1.5 Delimitations	8
2 Related Work.....	9
2.1 Choice of algorithms	9
2.2 Choice of data.....	9
3 Theoretical Background	11
3.1 Introduction to AI	11
3.2 Machine learning	11
3.3 Algorithms	12
3.3.1 K-nearest neighbor.....	12
3.3.2 Logistic regression	13
3.3.3 Artificial neural networks	14
3.3.4 Boosted Decision Tree	16
4 Method	18
4.1 Pearson’s correlation method	18
4.2 Chi-Squared method	18
4.3 Dataset	18
4.4 Azure machine learning platform	21
5 Result & analysis.....	22
5.1 Result.....	22
5.2 Result analysis	24
5.3 Algorithms	24
6 Discussion	26
6.1 Result discussion	26

6.2 Sustainability & ethics perspective.....	26
6.3 Choice of technique	26
6.4 Future work.....	27
7 Conclusion.....	28
Works Cited.....	29
Appendix	32

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial intelligence
ML	Machine learning
ANN	Artificial neural network
BDT	Boosted decision tree
K-nn	K-nearest neighbor
AML	Azure machine learning
LR	Logistic regression
Correlation test	Relationship between two variables.

1 Introduction

1.1 Background

Within healthcare recent focus has been on 30-day readmission which is when a patient is being readmitted to the hospital within 30-days after being discharged from the hospital. If a hospital has an excessive number of readmissions within thirty days, they can obtain a punishment of lowered payment. To avoid getting lowered payment hospitals in the US follows the Hospital readmission reduction program (HRRP) which was implemented to increase the quality of healthcare, reduce reimbursement to hospitals, reduce healthcare costs, maintain quality and decrease the high costs of patient readmission. [1]

A subject that can be combined with readmission is AI which is a technique that has created great potential across multiple sectors and is today especially reverberating across the healthcare sector. The growth of AI has led to new innovative projects such as the Babylon service provider that was launched in 2016 and was used to predict the best action to take based on several symptoms given by a patient. IBM Watson-Oncology is another project which has been used to pick up drugs to patients with cancer and has had a better efficiency than a human being. [2,3]

An efficient way to reduce high readmission costs or emergency room visits using AI today is to collect data from patients and use this data to make new predictions. One way of collecting important data from patients is through mobile monitoring which collects vital signs and other health data and uploads it to the cloud for further analyzing, this can then be used to alert doctors if patterns or unusual signs are found within the data. [4]. Collection of data and pattern analysis in ML could potentially lead to lower readmission costs and avoid symptoms from going undetected.

1.2 Research questions

Is machine learning a reliable option to predict hospital readmittance?

How do different machine learning algorithms differ from each other in performance?

1.3 Problem

In the United States during 2012 the medical cost for patients with diabetes mellitus (DM) was estimated to be \$178 billion and it is approximated that United States spends \$218 billion per year in health care for patients with diabetes. Patients diagnosed with DM has exorbitant rates between 14.4% and 22.7% of the total readmissions. In the United States alone \$41 billion was spent on 30-day all-cause readmissions during 2011. [5, 6]

Though AI is becoming more regularized within the healthcare sector it can still be challenging to determine which algorithm to use for the data that has been collected. Several factors need to be considered when choosing algorithm, the size of the data, quality and type of data. Even the most experienced data scientists cannot determine which algorithm is the right one to use without trying them, it is therefore crucial to test several algorithms to determine which has the best result. [7]

1.4 Purpose

The intent of this study is to collect and use data of diabetic patients who have either been readmitted or not readmitted and feed the data into four ML algorithms to determine which algorithm performs the best. Comparisons about true/false positives, training scale and accuracy are performed on Boosted decision tree, Logistic regression, K-nearest neighbor and Artificial neural networks. An introduction to AI and ML is given in the theoretical background.

1.5 Delimitations

Due to the time limitations most of all tests are done on Azure platform, this means that the experiment depends on the function ability of the algorithms provided by the cloud service. The data collected also contains several missing values and it can be hard to determine the affect these missing values have on the accuracy. The dataset used for the experiment is also imbalanced, only 11% of the total dataset are patients who have been readmitted within 30 days which represents 11353 patients in the dataset. Non-readmittance patients in the dataset represents 89% of the total dataset (88 647 patients) which means that to avoid biased results for some tests 77 294 patients are removed from non-readmittance patients to get a balanced dataset.

2 Related Work

2.1 Choice of algorithms

Logistic regression (LR) is a classification algorithm that can be used to predict two possible outcomes from the data, in our study; readmitted within 30 days or not readmitted. LR assumes that there is a linear separability between the points (Refer to p.13-14). Previous studies such as [8] has shown that LR is a successful algorithm to use with our dataset as it had an accuracy of 88.8% for the same dataset, and other similar studies [9] with similar dataset resulted with a 79% accuracy.

K-nearest neighbor: (K-nn) The reason for using K-nn is to be able to measure the speed and performance of a model that requires no explicit training with a model that requires training (Read more on p.12). K-nn was tested in previous studies of the same dataset [10] with Euclidean distance. However, the dataset tested with K-nn contained dataset with columns that had one observation which might have affected the prediction accuracy of the test. Accuracy was 67% while recall and specificity were between 46% and 65% in the test.

Artificial neural network: (ANN) Because of the vast amount of input data and only two possible outputs the ANN architecture is ideal to the dataset used in this study. The dataset already has the answer to the readmission question; therefore, the supervised training model will be used. [11]

In a recent study [12] the ANN model was used. The result showed that the ANN model performed at a very high speed and generated an accuracy of 95%. The similarity to our data makes the ANN model relevant to include in this research.

Boosted decision tree: (BDT) was chosen because of the large number of data that is used, according to a recent study [13] the boosting part in the algorithm is important because it will create new trees and correct the error in the earlier tree during the training process. It also is an algorithm which is fast during the training phase, has a high precision to give a desired output which is crucial to the dataset used in our study. The dataset used in previous studies [13] on big datasets can grow just like the dataset used in our study.

2.2 Choice of data

Several studies have been done focusing on diabetic readmission data such as [8]. In the study the authors ran diabetic dataset in 11 different algorithms: Naïve classifier, Naïve bayes, Binomial log reg (kitchen sink), Binomial log reg (selected features based on significance in kitchen sink model), Multinomial log reg (kitchen sink), Elastic net, Random forests, Support vector machine (SVM) (no kernel), SVM (3rd order polynomial kernel), SVM (gaussian kernel) and ANN. The study has little effort in data mining techniques such as presenting correlation tests between independent values and dependent value. 7 irrelevant columns were removed that were considered “too sparse” but no proof of the data being too sparse is presented in either graphs or tables. In the study [8] the authors proceeded to remove additional 8 columns without showing any proof of why. By not providing proof the research article can become less reliable for the reader. In the research [8] a total of 15 columns of the 50 were removed that were considered bad. This can have negative effects on the algorithms if the columns have not been analyzed correctly such as running them in correlation tests to

see if there is a relationship between the columns and the dependent value or analyzing the columns in graphs to observe their outputs.

The accuracies presented in [8] can be considered as biased as the data used for the study has an imbalanced dataset. 11% of the dataset are patients who have been readmitted and 89% of the dataset are patients who have not been readmitted. The imbalanced dataset can lead to inaccurate results because if the algorithm only predicts that all patients will not be readmitted it can still generate a high accuracy of 89% without predicting a single readmittance patient. This is something that can be seen in the study. The study's [8] best performing algorithm Random forest generated an accuracy of 88.8% but when examining the confusion matrix (Figure 1) of Random forest it shows poor accuracies when predicting readmittance patients. The algorithm predicted 19 940 (17869+2071) patients as non-readmittance patients (Pred 0) whereas 17869 were correct and 2071 incorrect which is an accuracy of almost 90%. The algorithm only predicted 411 (194+217) as readmittance patients whereas 217 were correct and 194 were incorrect which is an accuracy of approximately 50% (Pred 1). This result is an example of biased result because it has very low accuracy when predicting readmittance but high when predicting non-readmittance and because non-readmittance represent higher volume of data the accuracy will be higher. If the data is balanced the algorithm instead needs to be able to predict high from both classes to achieve high accuracy.

RF	Actual 0	Actual 1
Pred 0	17869	2071
Pred 1	194	217

Figure 1: Confusion matrix [8]

A similar technique was used in a different study [9] where the categorical data in the dataset was transformed into numerical. The test was then performed in 7 different algorithms, K-nn, LR, Stochastic gradient descent, Naïve bayes, Decision tree, Random forest and Gradient boosting classifier. Several columns in the dataset such as diag_1, diag_2 and diag_3 that represents patient diagnosis were excluded from the study without taking into consideration the affect these can have on the accuracy. The study also lacks data mining techniques as it includes irrelevant columns with 0 correlation to dependent value such as tolbutamide and glipizide-metformin. The best model Gradient boosting classifier in the study was only able to get an accuracy of 58% when predicting readmissions.

3 Theoretical Background

3.1 Introduction to AI

AI can be found in homes, cars, the office or the bank. The point of AI is to have a computer do similar tasks to what a human being can do through different approaches:

Learning is done by applying learning functions to a machine consisting of errors and trials. The machine starts off with a random process and continues until it uncovers the correct process and stores it into memory.

Reasoning signifies drawing of inferences which can be divided into two classes deductive and inductive reasoning. Having a machine draw inferences depending on the situation is until today one of the hardest parts about AI.

Problem solving is an approach for the machine to go through multiple stages and look for the stage leading to a specific goal.

Perception is when a machine can scan the environment to identify real objects. This is one of the advanced technologies that is widely used today. By applying sensors to the machine, it can identify individuals or having self-driven cars keeping track of the current speed limit.

Language is when the machine can express itself in a way such as mini language, e.g. when a self-driven car can detect hazards ahead and warn the driver. [14]

3.2 Machine learning

Supervised & Unsupervised learning

ML can be divided into two varieties, supervised and unsupervised learning. Supervised learning is when the machine is given a set of input values that are related to an output value. The goal is then for the machine to discover a relation between the input and the output value based on the given values. The computer is provided with feedback whether it has achieved the correct output value with help of error functions that measure how close the machine was. The models used to find a relationship vary depending on the data it is given; these models can be divided into classification tasks and regression tasks. [15,16]

Classification tasks are used to create models out of data that are labelled as discrete values such as predicting if a person will pay their loan or not.

Regression tasks are for continuous values and creates the models out of data consisting of quantities as labels, for example making a prediction on a specific house price.

Examples of algorithms used in supervised learning are K-nn, ANN and LR.

In unsupervised learning the machine is not provided with input-output values and there are not error functions telling the computer how successful it was, instead the computer learns to recognize patterns itself from just input values. Examples of unsupervised learning are: estimating what the typical person looks like in a specific neighborhood based on features such as hair-color or eye-color.

Clustering (unsupervised) is a model that separates data into groups without labels by comparing features within the data. Each of the data then belongs to a specific group created by the model.

Examples of algorithms used in unsupervised learning are K-means, Mean-shift clustering, and Density based spatial clustering of applications with noise. [16]

3.3 Algorithms

3.3.1 K-nearest neighbor

According to [17] K-nn is a rudimentary algorithm used in ML in scenarios such as classification and regression. K-nn is a non-parametric method that belongs to the family instance-based learning. It learns from training data without an explicit training phase. During the classification stage K-nn assigns new data to a class based on its nearest neighbors (Figure 2). Its nearest neighbors are calculated by measuring the straight-line distance between the new data and its nearest neighbors (which are datapoints already known to the model). The K value determines how many neighbors to compare to. The data is then assigned to the class among its nearest neighbors. [17,18]

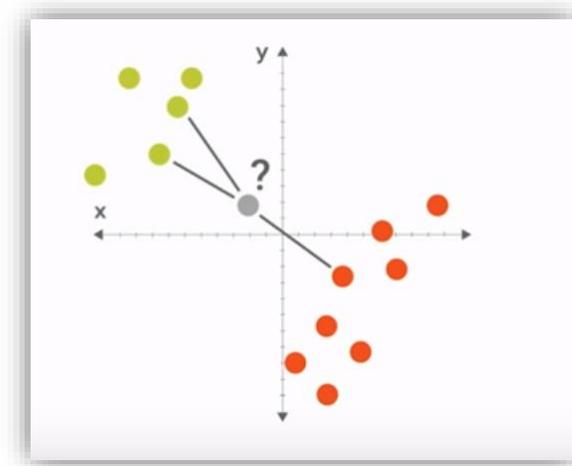


Figure 2: K-nearest neighbors (Google Developers, 2016. 4:26) [19]

Distance functions can have different accuracies depending on the data that is being used, for numerical values distance functions such as Euclidian, Manhattan and Makowski functions can be used and for categorical values Hamming distance function can be used. [18]

The **Euclidian distance** is a function used for measuring the distance between two data points. Simple 2-dimensional data can be calculated by using the function shown in Formula 1 where (x_1, y_1) and (x_2, y_2) represents the dimensions in the data. The distance function is also scalable to compile with data with high dimensions by adding additional features to the formula. [20]

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Formula 1: Euclidian Distance (Rosalind, n.d.) [20]

3.3.2 Logistic regression

LR is a technique that is based on the linear regression concept. The linear regression is used to respond in continuous numerical output values such as predicting house sales or stocks. It assumes a constant value of a feature results in a constant change in the value of the response and it takes no consideration into responding with qualitative variables.

The standard linear regression uses the basic linear equation shown in Formula 2 where P represents the dependent value, α represents the slope, x represents independent value and β represents the intercept. The algorithm trains by adjusting the coefficient values (β and α) to find the best fitting line through the points shown in Figure 3. Linear regression can however exceed a classification output with values between $(-\infty)$ and $(+\infty)$ which means it cannot give responses such as true/false. [21,22]

$$P(x) = \alpha * x + \beta$$

Formula 2: Linear equation (Giuseppe Ciaburro, 2018) [21]

Linear regression does not work when binary values are expected. When qualitative outputs are expected LR is an option. LR feeds the basic linear equation into a sigmoid function to achieve classification output with a probability range between $(0; 1)$ shown in Formula 3 where $\frac{1}{1+e^{-x}}$ represents the sigmoid function, β_0 represents the intercept, β_1 represents the regression coefficient, x independent value and P represents the predicted output. [21,23]

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Formula 3: Logistic regression (Dr. Saed Sayad, n.d) [23]

LR learns by attempting to achieve the best fitting ‘S’ line shown in Figure 3. After feeding the training data into logit function LR seeks to optimize the curve through the points by using function called the maximum likelihood (a way of observing the likelihood for all parameters given their observed status). The likelihoods are then multiplied together (the closer this sum is to 0 the better the line is fit). The error is then minimized by changing the coefficients values (β_0 and β_1 , Refer to Formula 3).[22]

Adjusting the coefficients is often done through a model called gradient descent. Gradient descent takes the sum of the error function and tries to minimize it as close to 0 as possible by adjusting coefficients values. Once the coefficients have been set and the lowest error sum has been found the model is ready to predict real unknown values.[21]



Figure 3: Linear regression & Logistic regression (Josh Starmer, 2018. 2:51-3:56) [24]

3.3.3 Artificial neural networks

ANN is a model within AI which primary task is to behave like a biological network. ANN is built to replicate the human brain by being able to recognize patterns and make assessments based on the information it receives.

A computer is not by definition smart, but when an ANN is built on the fundamental parts of a biological neuron network a computer can solve tasks like a human brain. Though, an ANN needs training just like a human brain. By behaving like a human brain, the ANN can see patterns, generalizations and make its own assumptions on the dataset. ANN of this kind needs huge amount of training to learn a new task. Often ANN gets thousands or even millions of input data to learn and recognize patterns. [25]

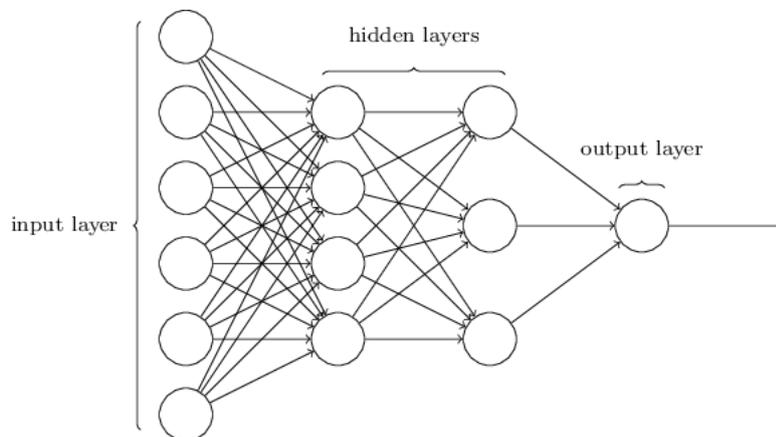


Figure 4. ANN visualized.[26]

ANN consist of neurons and synaptic strengths called weights shown in Figure 4. ANN contains different number of neurons depending on the input and the desired output. Most often a network consists of one or many hidden layers. Hidden layers compute a certain task and the number of hidden layers and their number of neurons can differ depending on the task.

Sigmoid neuron is most commonly used to implement ANN. The sigmoid function realizes the possibility to train a network and have it to behave in a manner that is desirable.

This also makes it more convenient to tune the network to behave as accurately as possible. Unlike a perceptron which is one of the first methods used to implement ANN and to produce 0 or 1 as an output, the sigmoid neurons can take inputs between 0 to 1. In a sigmoid network a valid number can be 0.145 or 0.99 for example.

A sigmoid function has inputs, $x_1, x_2 \dots$ and weights called $w_1, w_2 \dots$ and bias for each layer in the network, called b .

When a sigmoid function's input has a large and positive value the functions output will be close to 1, when the functions input has a low and negative number the output of the neuron is close to 0. With these numbers the behavior is close to a perceptron network, the big difference is when the input is midsize.

When the ANN needs to be changed to the desired output only a small change is needed to the bias and the weights, this is unique for the sigmoid function. Every neuron has a threshold, and if the input to the neuron is higher than the threshold the function is triggered. The sigmoid function is applicable on all the neurons in the different layers.

The ANN does not learn from the first time an input enters the network, an ANN needs training to preform and see patterns to operate as intended. Backpropagation is a method used to determine which weights between neurons aren't correctly configured. The ANN is trained to comprehend the output known by the developers. If the given output by the ANN variates from the known output the deviation can be calculated. When an ANN consists of hundreds, or even thousands of weights the question is how to find the weights which needs to be configured based on the deviation. [25, 26]

Calculating the different values throughout the network can be done with a method called Pass-Forward. This method takes the input to the layer which going to be calculated. The following formula is used to calculate the value of the specific neuron seen in Formula 4

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1$$

Formula 4. Pass-Forward formula for a unique neuron.[27]

Formula 4 calculates the total net input to the neuron, in this case a neuron which is the first neuron in the hidden layer (i.e. h_1). It is applicable on every neuron in the ANN. The formula also includes i_1 and i_2 which are the neurons from the previous layer, w_1 and w_2 are the weights which is connecting neurons from the previous layer and b_1 is the overall bias for the neuron.

The obtained answer from Formula 4 is then squashed in the logistic function in Formula 5 to get the total output of a single neuron.

$$out_{h1} = \frac{1}{1 + e^{-net_{h1}}}$$

Formula 5. Total output of a neuron.[27]

When the final value from the output layer is obtained, Formula 6 where *target* stands for the desired output that the ANN is supposed to produce, and *output* stands for the actual output which the ANN produce is then used to calculate the error of the network.

$$E_{total} = \sum \frac{1}{2} (target - output)^2$$

Formula 6. Formula to calculate the total error of the network.[27]

Backpropagation always starts at the output layer, to see how much a weight contributes to the error, in other words “*the gradient with respect to the weight*”. An illustration of a neuron and how the different input and output looks like can be seen in the Appendix.

Backpropagation on the hidden layer is a bit more complex because the output of the hidden layer contributes to the error, therefore a calculation of the deviation needs to be done both on the weights but also on the output of the neuron to see how much a misclassified weight is contributing to the total error. [27]

3.3.4 Boosted Decision Tree

Decision tree algorithm is a tree-structure algorithm where data is represented as nodes and leaves. The depth of a tree is based on the different parameters which occurs in a dataset and their possible values. Simplifying the idea one can think of a question like “*Will it rain tomorrow?*”, this question in a decision tree is represented as a node and the different answers to this question can be the leaves, like *yes* or *no*. This is called classification when the network uses a binary architecture, on the other hand, if a network shall predict the temperature tomorrow and there are several possibilities as an answer, this is called regression. Decision trees are powerful but a small change to the data can result in a different answer, also, the tree can be very large based on the input to the root node and therefore it can be hard to maintain such a big network. It is also very important to represent the most important nodes at the top of the tree, otherwise the tree can be unnecessarily big. As seen in the Figure 5 the most important node is placed at the root position.[28]

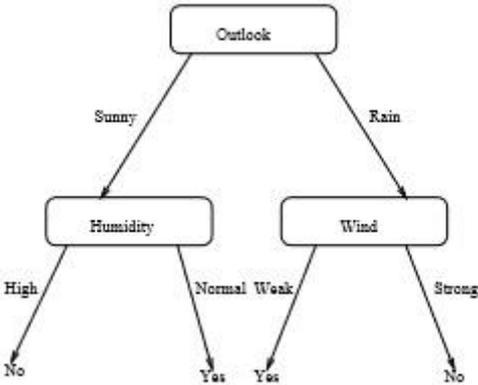


Figure 5. Decision tree where nodes are the questions and the leaves are the answer.[29]

One solution to get a more accurate decision tree is to use boosting. When a tree is trained under supervised learning the outcome from the tree can differ from the correct answer. To handle the error in the tree a boosting algorithm is performed on the training dataset multiple times to set up correct weights for some known errors in the tree. These errors get new weights and in the next iteration of the tree these misclassified events are prioritized by the algorithm. In this manner, many different trees can be created under the training session. In the end the tree is correcting the errors in earlier trees for every new iteration. A final prediction on the result is the total number of tree's created by the boosting algorithm [30].

4 Method

4.1 Pearson's correlation method

Pearson's correlation is a method that measures the strength and direction of a linear relationship between two variables. Pearson's correlation statistically evaluates whether there is a linear relationship and responds with a value between -1 and 1 where 1 is the highest relationship. Requirement for Pearson's correlation is that the data is continuous.[31].

4.2 Chi-Squared method

Chi-Squared test is a way of measuring the relationship between two categorical values, this can be done by comparing the observed values to the expected values shown in Formula 7 where x^2 is the Chi-Squared value, O is observed value and E is the expected value [32]

$$x^2 = \sum \frac{(O - E)^2}{E}$$

Formula 7 Chi-Squared formula.[32]

4.3 Dataset

The dataset used for the study is taken from UCI Machine Learning Repository. The dataset contains 100.000 patients during the time 1999–2008 with diabetic symptoms. The supervised data is a multiclass labelled dataset consisting of patients who has been readmitted within 30 days, not readmitted at all and not readmitted within 30 days.

The data contains 50 columns with different features whereas 2 of them are directly irrelevant when predicting. The irrelevant features are encounter_id and patient_nbr. These features represent unique id's that has no relationship to the dependent value, therefore they are not used for this analysis. There are also columns consisting of missing data where the percentage stands for the missing data. Examples are race: 2%, weight: 97%, payer code: 52%, medical specialty: 53% and diagnosis 3: 1%. The columns weight and payer code are removed from the analysis. Medical specialty is also excluded from some measurements to avoid biased results because replacing missing values with different values can be interpreted as real values and give false correlation points.

The multiclass dataset was first transformed into a two-class dataset shown in Figure 6 where NO represents all non-readmittance patients and <30 represents all patients readmitted within 30 days. This was done by using R-script.

Due to the asymmetrical dataset shown in Figure 6 the measurement of the relationship between independent values and the dependent values can be a complex task to perform without getting biased results because due to the imbalanced dataset some features can get low correlation points even if they are high because there is less data of them, therefore it is important to balance out the dataset.

The purpose of balancing out the data is also to punish the algorithms if they cannot make correct predictions from both classes (readmitted and not readmitted). This is something seen in previous articles such as [8] where algorithms could make wrong predictions on

readmittance patients but still achieve high accuracy. When balancing out the dataset the algorithm cannot achieve high accuracies unless it has high accuracies for both classes.

Balancing out the data also avoids class overlapping which is when a class with majority of data has similar data to the minority class and causes class data to overlap each other. K-nn is especially sensitive to overlapping, K-nn will measure distance to closest neighbors, if one label has much data similar to the other label the noise can increase, and the model will be most likely to have more neighbors to the label with the most data. [33]

Instead of using 100.000 rows for correlation test 11353 rows were taken out from readmitted and 11033 rows from non-readmitted before performing the correlation test (Figure 7).

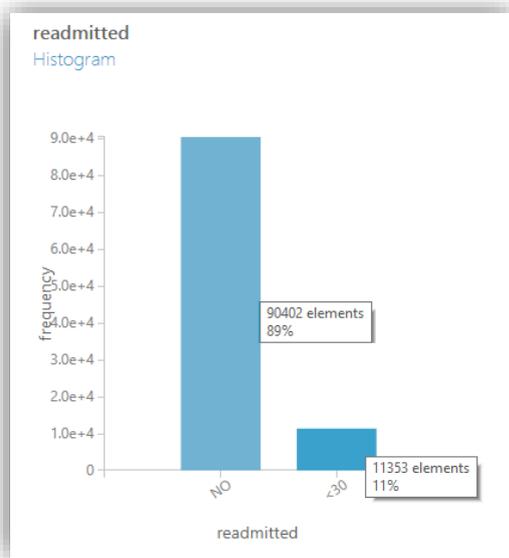


Figure 6: Showing original dataset

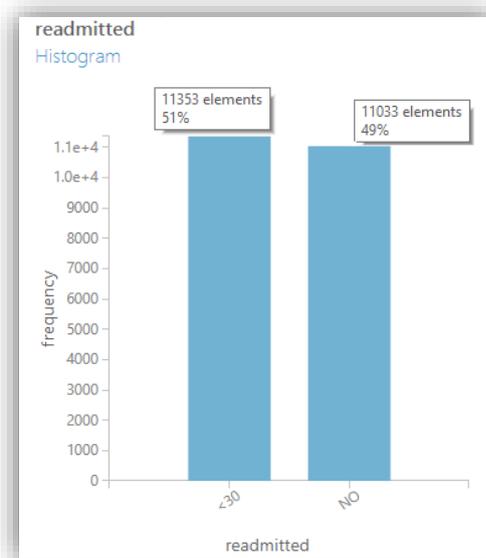


Figure 7: Showing a balanced dataset

To run the tests the dataset was separated into two: numerical and categorical values as Pearson's correlation method only works with numerical and Chi-Squared works for categorical.

Pearson's correlation method was used to measure the relationship for all numerical values within the dataset. The top 5 variables with highest correlation to dependent values are shown in Table 1 (Refer to Appendix for full result). Chi-Squared method was used to compare all categorical features to the dependent value. A visualization of the Chi-Squared with top 5 variables with highest correlation are shown in Table 2 and all variables with 0 correlation to the dependent value are shown in Table 3 (For full result refer to Appendix).

Features within the dataset: Medical specialty which contains 52% missing values had to be excluded from the test as it resulted with a biased result. Diag_1, diag_2 and diag_3 which represents patients' diagnosis was also excluded from the test as they contain a mixture of categorical and numerical values that can affect the tests negatively because neither of the methods works with mixtures.

Pearson Correlation - Numerical correlation test (closer to 1 means a higher correlation):

Readmitted	Number_diagnoses	Number_inpatient	Number_outpatient	Number_emergency	Admission_type_id
1	0.23833	0.242717	0.179141	0.150791	0.135292

Table 1 – Top 5 highest correlation to dependent value (numerical values)

Chi-Squared correlation results – categorical values (higher points mean higher correlation)

Readmitted	Insulin	Change	Max_Glu_Serum	diabetesMed	glypuride
1	1090.38497	405.73521	253.293392	234.894917	181.072474

Table 2 – Top 5 highest correlation to dependent value (categorical values)

Readmitted	acetoexamide	Insulin	Change	Max_Glu_Serum	diabetesMed	glypuride
1	0	0	0	0	0	0

Table 3 – variables with 0 correlation to dependent value (categorical values)

Six features with zero correlation to the dependent value was found from the tests. Visualizing this data in graphs (Figure 8 and 9) shows that the data only has one possible observation and since these values never change, they have no relationship to the dependent. These values are removed from the analysis.

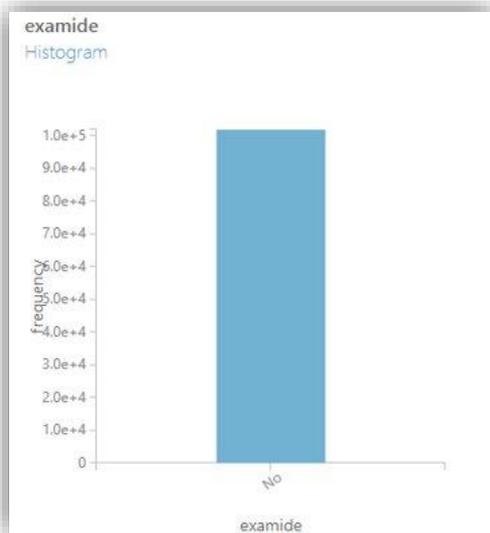


Figure 8: showing feature observation

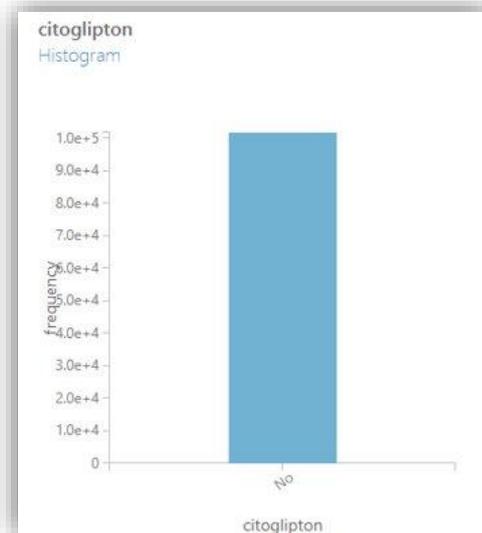


Figure 9: showing feature observation

The dataset contained many features with same observations as Figure 8 and 9. Six columns had 0 correlation to the dependent but additional 3 columns were found with same observation as Figure 8 and 9 when analyzing the columns in graphs. Following features were removed from the dataset: acetoexamide, tolbutamide, troglitazone, examide, citoglipton, glipiride-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, metformin-

pioglitazone. By removing all columns with 0 correlation the dataset becomes more reliable as all columns now contributes information to the algorithm.

4.4 Azure machine learning platform

All tests and algorithm evaluation except K-nn were done on AML. AML is a platform that allows users to interact with components through drag-and-drop method. Some examples of components offered are cleaning data, selecting columns and choose training dataset and predictive dataset.

The platform gives free access to multiclass and standard two-class algorithms such as LR, SVM, BDT and ANN. The studio is also suitable to perform various data mining tasks as it supports R-script and python script, which can be used to both transform multiclass labelled data into two-class and prove linear separate relation in the dataset.

AML offers evaluation model that presents information such as analyzing performance of algorithm, accuracy – how good hit rate the model had and confusion matrix that presents how accurate the model was (More on AML can be found in Appendix). [34]

5 Result & analysis

5.1 Result

The tables below show results of the accuracy of the different algorithms (Table 4 and 5) and confusion matrices that describes true and negative positives (Figure 10 and 11). Results for both balanced data and imbalanced data is presented. The results also present how well each algorithm performed (Figure 12-15) and timetable (Table 6).

Test 1 unbalanced	K-nn	LR	ANN	BDT
Accuracy	88.7%	89.1%	88.7%	89%

Table 4: Accuracy for each algorithm (unbalanced data)

BDT	Actual NO	Actual <30
Pred NO	18084	44
Pred <30	2186	39

ANN	Actual NO	Actual <30
Pred NO	18005	24
Pred <30	2286	38

K-nn	Actual NO	Actual <30
Pred NO	17597	2269
Pred <30	-	-

LR	Actual NO	Actual <30
Pred NO	18094	34
Pred <30	2183	42

Figure 10: Confusion matrices for all algorithms (unbalanced data)

Test 2 balanced	K-nn	LR	ANN	BDT
Accuracy	67.6%	77.3%	78.5%	79.2%

Table 5: Accuracy for each algorithm (balanced data)

BDT	Actual NO	Actual <30
Pred NO	1769	429
Pred <30	502	1778

ANN	Actual NO	Actual <30
Pred NO	1794	404
Pred <30	560	1720

K-nn	Actual NO	Actual <30
Pred NO	11754	5843
Pred <30	1244	1025

LR	Actual NO	Actual <30
Pred NO	1733	465
Pred <30	552	1728

Figure 11: Confusion matrices for all algorithms (balanced data)

From the confusion matrices it is possible to read out what the accuracy each algorithm had when predicting if someone should be readmitted or not readmitted within thirty days. Pred NO in Figure 11 and 12 shows how many values were predicted as not readmitted and the columns Actual NO and Actual <30 shows how many predictions were correct and incorrect.

Pred <30 shows how many patients were predicted as patients who will be readmitted within thirty days and Actual NO and Actual <30 shows how many correct ones it predicted as well as incorrect ones.

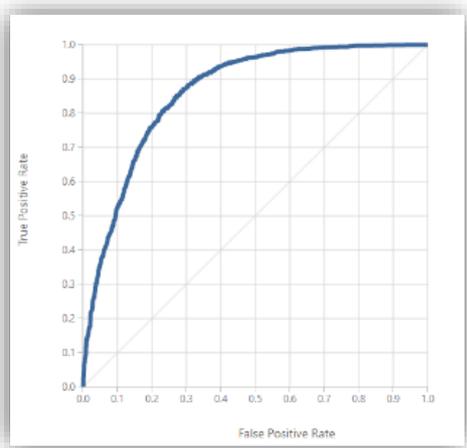


Figure 12: ANN training scale (balanced).

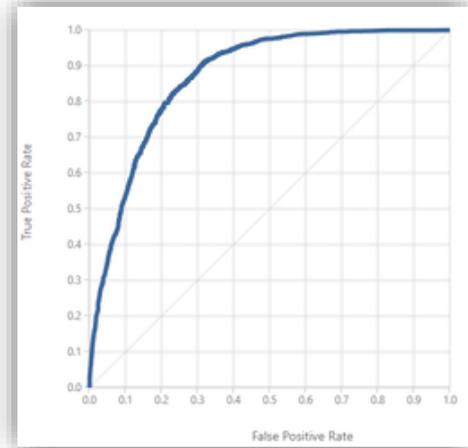


Figure 13: BDT training scale (balanced)

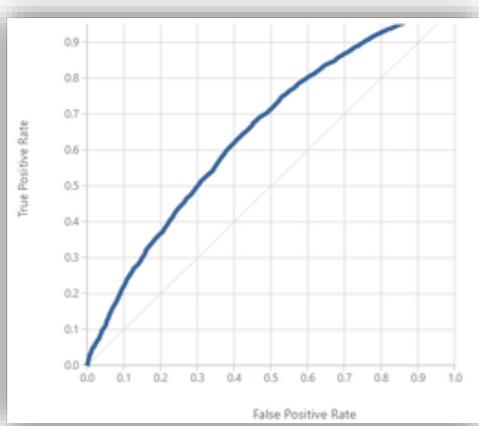


Figure 14: ANN training scale (unbalanced).

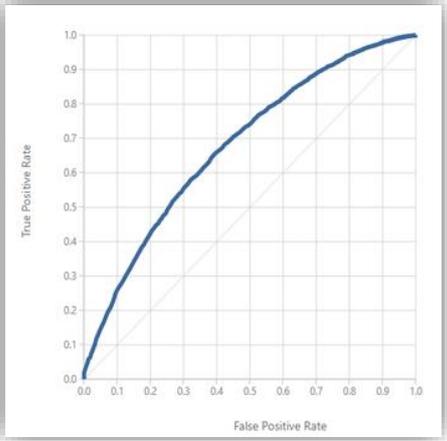


Figure 15: BDT training scale (unbalanced)

Figure 12-15 shows how well the algorithms with the highest accuracy performed by plotting true positives (how many correct predictions of non-readmittance patients) against false positives (how many correct predictions of readmittance patients), this curve is also known as Receiver Operating Characteristic (ROC). The closer the line is to the edge the better the algorithm is performing. [34]

Time table	K-nn	ANN	BDT	LR
Elapsed time	0:02:01	0:12:12	0:02:25	00:00:33

Table 6: Elapsed time for all algorithms (balanced data)

The timetable in Table 6 presents the average time it took for each algorithm to complete the task of training the model in format hours, minutes and seconds. (Full time results can be found in the Appendix)

5.2 Result analysis

Results from unbalanced dataset shows high accuracy for each algorithm with a high percentage above 80% for each algorithm. The confusion matrices (Figure 10 and 11) show what the algorithm predicted and what the correct answer was. Each algorithm had a low rate in predicting readmittance within thirty days when using the unbalanced dataset, each algorithm was only able to predict less than 40 readmittances while failing on more than 2000. This proves that accuracy has little significance when using imbalanced dataset. This is mainly because the algorithm only needs good prediction rates when predicting non-readmittance patients as they represent 89% of the dataset.

When the dataset was balanced by taking 11353 patients who has been readmitted and 11033 patients who has not been readmitted the general accuracy lowered significantly (Compare Table 4 and 5). Each algorithm had an accuracy below 80% but confusion matrices show that the algorithms perform much better in predicting readmittance and the ROC curve improved drastically, the balanced dataset results in a curve that is closer to the edge (See figure 12-15). BDT, LR and ANN almost had 80% accuracy in predicting both readmittance and non-readmittance patients. This ~80% accuracy also shows how important the data mining is when working with a dataset and how important a balanced dataset can be as it almost had over 30% better accuracy when predicting diabetic readmissions compared to the previous studies [8] and [9] with accuracies between 50 and 57%.

Balancing out the data do however result in lower accuracy when predicting non-readmittance patients because due to the balancing of the data a significant portion of non-readmittance data must be taken out. This can lead to information such as how frequent a data is repeated from being excluded.

5.3 Algorithms

Using the unbalanced dataset, the algorithms had an equal performance above 80%, however all the algorithms performed poorly when predicting readmittances. K-nn performed the poorest when using imbalanced data, the model predicted only one possible outcome which was not readmitted, this can be because of the data overlapping issue with K-nn [33].

BDT, ANN and LR performed equally when using the balanced dataset, the best algorithm BDT had an accuracy of 79% and an accuracy of 80% when predicting readmittances. K-nn which had the poorest performance was only able to get an accuracy of 67% and less than 50% when predicting readmittances. The algorithms training phases were also measured in

time. LR was the clear winner with an average time of 33 seconds in training phase. The algorithm with the poorest time result was ANN which had an average time of 12 minutes.

Due to the limited timeframe tune modeling the hyperparameters on K-nn had to be excluded, by using hyperparameters the model goes through different configurations until it finds the best one [35]. Using hyperparameter tuning on LR, BDT and ANN increased the accuracy with 1 to 3%.

6 Discussion

6.1 Result discussion

The best performing model showed an 80% accuracy on the balanced dataset, which is a high result, although there is a 20% error margin. If this should be applicable in a hospital environment it is not satisfying enough to trust just on the AI. Some predictions will be wrong and therefore this solution should just be used as a guidance to the nurses and doctors. If a prediction produces a not readmitted answer and the doctor trust this fully a patient can be sent home without any medications or advices to be aware of symptoms. If the AI is used as a guidance parallel with traditional methods to identify symptoms it could be a helpful tool to reduce work from the doctors and speed up the process to give the patient a reliable answer.

6.2 Sustainability & ethics perspective

The results show how much potential machine learning has within the healthcare sector, for the study that resulted in almost 80% accuracy only 22386 patients were included to achieve a balanced data. The balanced data includes 11353 non-readmitted patients and 11033 patients readmitted within 30 days. Gathering of more readmittance data can potentially increase accuracy drastically. A service monitoring and providing predictions about patients could prevent symptoms from going undetected and save money and time that could be used for other important scenarios. The service would however result in adding more servers and services for collection and storing of the gathered data which could have a negative effect on the environment. This could however be prevented by using successful cloud services with sustainable cloud computing.

From an ethical perspective the collection of the data can result in privacy issues, storing data about patients on servers will require a sustainable security system that cannot be accessed by unauthorized users. However, using services provided by an external part can lead to new ethical issues. One problem which can occur when an external part is responsible for the data is the possibility that providers have direct access to the data. Information on how data is collected and stored would have to be clearly defined in the patient act for respective country.

6.3 Choice of technique

The processing of the data through data mining was the most important task in the project. By using Pearson's correlation and Chi-Squared test it was possible to easily inspect the data and point out the columns with the strongest correlation to dependent value. By performing the correlation tests irrelevant data with low correlation could be removed from the dataset. Due to the AML's compatibility with components such as R-script and edit metadata it was possible to sort out data and convert string values to categorical values (See more in Appendix). AML also allowed for scrutiny of the data through graphs presenting outcomes of all columns within the dataset. Through this it was possible to remove data such as glipizide-metformin and tolbutamide containing only one outcome.

6.4 Future work

All the algorithms were implemented in AML, because of this choice the algorithms belong to Microsoft and therefore the possibility to analyze and make changes to the algorithm was impossible. The group took the decision to use the cloud-based service mostly because of flexibility but also because of the time-constraints. We trust the service Microsoft provides but it could be even more interesting to write the algorithms by ourselves and see if there can be some difference to the final output, also the performance of the algorithm may differ if they are implemented in another manner. If the algorithm were implemented by us, we could have been analyzing how each algorithm behave during runtime. This is impossible to see in AML because of the closed-source code. All the algorithms can be implemented in python with the help of some tools to validate the output and get the necessary data as accuracy and confusion matrices. In the beginning we contemplated if the implementation of BDT and ANN were possible, but the main goal was to see if ML is trustworthy enough to be used in healthcare, so therefore this idea was not realized.

7 Conclusion

The most important aspect of using AI is how the data is collected. Collecting the data and analyzing the data is a crucial part of AI if it should be used within healthcare. Using ML algorithms on readmittance data showed that the accuracy can have low significant meaning when using unbalanced dataset. The tests on the readmittance data using both imbalanced and balanced data showed that more imbalanced data lead to more noise and the data with the majority of the data will most likely have more features similar to the other label due to the frequency of the data which can lead to more false positives (when incorrect prediction of readmittance patient). Using balanced data for the second test resulted in a lower general accuracy but a boosted accuracy when it comes to predicting both non-readmitted patients and readmitted patients, this shows that balancing out the data can lead to the model being able to boost the accuracy when predicting non-readmitted and readmitted patients.

The tests in this study showed that BDT was the best option to use for this data with an accuracy of 79.2% when using balanced dataset. However, ANN had the highest accuracy when predicting actual non-readmitted patients with almost 82% accuracy. Just because BDT was the best performing algorithm on the balanced dataset that does not necessarily have to mean it is the best to use for larger quantity of data. If the data increases in quantity the models perform differently, when using the imbalanced dataset, the best performing algorithm was LR with 89.1%. Also, BDT has a heavy memory load which makes it inappropriate to use on big data sets.

Performance wise the algorithms varied, BDT had the best ROC curve performance which means that its classifier's performance is very good for this dataset, but timewise LR performed the best with less than one-minute training time.

Based on the tests performed in this study it is easy to conclude that AI has a lot of potential within healthcare, the balanced dataset consisting of 11033 non-readmittance patients and 11353 readmittance patients produced almost 80% accuracy for the best model. The dataset also excluded several columns with only one possible observation. Collecting more data with more variations where these columns are included with more observations might lead to higher accuracy. Balancing out the data and performing tests on models also shows that collecting a more balanced dataset can lead to higher results, in this case there was only 11353 readmitted patients but if there had been more readmitted patients the models would most likely improve in accuracy.

Works Cited

- [1] Centers for Medicaid Services,” Department of Health and Human Services,” 12 April 2018. [Online]. Available: <https://www.cms.gov/medicare/quality-initiatives-patient-assessment-instruments/value-based-programs/hrrp/hospital-readmission-reduction-program.html>. [Accessed 22 March 2019].
- [2] Bali J, Garg R, Bali RT. Artificial intelligence (AI) in healthcare and biomedical research: Why a strong computational/AI bioethics framework is required? *Indian J Ophthalmol* 2019; 67:3-6.
<http://web.b.ebscohost.com.ezproxy.hkr.se/ehost/pdfviewer/pdfviewer?vid=10&sid=27bd5337-e895-4833-b976-f1c48ab1e04c%40pdc-v-sessmgr03>
- [3] S. Houlton,” How artificial intelligence is transforming healthcare,” Wiley Interface Ltd, 2018. Available at: <https://onlinelibrary-wiley-com.ezproxy.hkr.se/doi/epdf/10.1002/psb.1708>. (Accessed 2 February 2019)
- [4] L. Qing, W. Chaunxue, “Mobile remote medical monitoring system” IEEE, 2016. Available at: <https://ieeexplore.ieee.org/document/7849727/metrics#metrics> (Accessed 24 February 2019)
- [5] Center for health information and analysis. Performance of the Massachusetts health care system series: A focus on provider quality. Boston, MA 02116: Center for health information and analysis,2015. Available at: <http://www.chiamass.gov/assets/Uploads/A-Focus-on-Provider-Quality-Jan-2015.pdf> (Accessed 17 February 2019)
- [6] Ostling, Stephanie. Wyckoff, Jennifer. Ciarkowski, Scott. Pai, Chih-Wen. Bahl, Vinita. Gianchandani, Roma. (2017) The relationship between diabetes mellitus and 30-day readmission rates. Available at: <https://clindiabetesendo.biomedcentral.com/articles/10.1186/s40842-016-0040-x> (Accessed: 3 March 2019).
- [7] Microsoft Azure,” How to choose algorithms for Azure Machine Learning Studio”, 2019. Available at: <https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice> (Accessed 18 March 2019)
- [8] Stanford,” Beating Diabetes: Predicting Early Diabetes Patient Hospital Readmittance to Help Optimize Patient Care”, 2017. Available at <http://cs229.stanford.edu/proj2017/final-reports/5244347.pdf> (Accessed 18 March 2019)
- [9] Towards Data Science,” End-to-End Data Science Example: Predicting Diabetes with Logistic Regression”, 2018. Available at <https://towardsdatascience.com/end-to-end-data-science-example-predicting-diabetes-with-logistic-regression-db9bc88b4d16> (Accessed 18 March 2019)
- [10] Towards Data Science, “Predicting Hospital Readmission for Patients with Diabetes Using Scikit-Learn”, 2018. Available at: <https://towardsdatascience.com/predicting-hospital-readmission-for-patients-with-diabetes-using-scikit-learn-a2e359b15f0> (Accessed 6 March 2019)

- [11] Microsoft Azure, “Microsoft Neural Network Algorithm” 2019, Available at: <https://docs.microsoft.com/en-us/sql/analysis-services/data-mining/microsoft-neural-network-algorithm?view=sql-server-2017> (Accessed 28 March 2019)
- [12] Hammoudeh A, Al Naymat G, Ghannan I, Obeid N. Predicting Hospital Readmission among Diabetics using Deep Learning. Amman: Elsevier Ltd; 2018. [cited 2019 March 20]. Available at: https://www.researchgate.net/publication/328887677_Predicting_Hospital_Readmission_among_Diabetics_using_Deep_Learning
- [13] Johansson R. Machine learning på tidsseriedataset [bachelor thesis on the Internet]. Luleå; Luleå tekniska universitet; 2018 [cited 2 April 2019]. Available from: <http://www.diva-portal.org/smash/get/diva2:1256329/FULLTEXT01.pdf>
- [14] B. Copel and,” Encyclopædia Britannica,” Encyclopædia Britannica, Inc., 6 February 2019. [Online]. Available: <https://www.britannica.com/technology/artificial-intelligence>. [Accessed 24 February 2019].
- [15] M. A. Boden, Artificial Intelligence A Very Short Introduction, Oxford: Oxford University Press, 2018. (Accessed 2 April 2019)
- [16] H. Saleh, MACHINE LEARNING FUNDAMENTALS, Packt Publishing, 2018. (Accessed 19 April 2019)
- [17] Bonaccorso, Giuseppe. Machine Learning Algorithms: Popular Algorithms for Data Science and Machine Learning, 2nd Edition, Packt Publishing Ltd, 2018. ProQuest Ebook Central, <https://ebookcentral.proquest.com/lib/kristianstad-ebooks/detail.action?docID=5504925>. (Accessed 22 April 2019)
- [18] Sayad, Dr. Saed. "K Nearest Neighbors - Classification”, N.D. Available at: https://www.saedsayad.com/k_nearest_neighbors.htm (Accessed 16 February 2019)
- [19] Writing Our First Classifier - Machine Learning Recipes #5 2016, Google Developers [Online]. Available at: <https://www.youtube.com/watch?v=AoeEHqVSNOw&t=314s> (Accessed 4 January 2019).
- [20] ROSALIND, “Glossary Euclidean distance”, N.D, Available at: <http://rosalind.info/glossary/euclidean-distance/> (Accessed 06 June 2019)
- [21] G. Ciaburro, Regression Analysis with R, Packt Publishing, 2018. (Accessed 2 February 2019)
- [22] Brownlee, Jason.” Logistic Regression for Machine Learning ”1 April 2016. [Online]. Available: <https://machinelearningmastery.com/logistic-regression-for-machine-learning/> [Accessed 22 March 2019].
- [23] Sayad, Dr. Saed. "Logistic Regression”, N.D. Available at: https://www.saedsayad.com/logistic_regression.htm(Accessed 2 June 2019)
- [24] StatQuest: Logistic Regression 2018, Josh Starmer [Online]. Available at: <https://www.youtube.com/watch?v=yIYKR4sgzI8&list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe&index=2&t=0s> (Accessed 18 January 2019).

- [25] Begg, Rezaul Dr. Kamruzzaman, Joarder. Sarker, Ruhul. Neural Networks in Healthcare: Potential Challenges. Idea Group Inc; 2016. [cited 2019 March 03]. Available at: https://books.google.se/books?hl=sv&lr=&id=rA_wTEgY1H8C&oi=fnd&pg=PR1&dq=neural+network+healthcare&ots=1qIrEqiRrN&sig=YGtvl5h5eW8VubZ1kkeHcQWgV0o&redir_esc=y#v=onepage&q=neural%20network%20healthcare&f=false
- [26] Nielsen A. Michael. Using Neural nets to recognize handwritten digits. Neural Networks and Deep Learning. Determination Press; 2015. [cited 2019 Feb 28]. Available at: <http://neuralnetworksanddeeplearning.com/chap1.html> (Accessed 2 May 2019)
- [27] Mazur, Matt. A Step by Step Backpropagation Example. 2015 [cited at 20 Feb 2019]. Available at: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>
- [28] Moberg, Anders. Pettersson, Anders. AI Lab2 Boosting [Internet]. Umeå universitet; 2004. [cited 20 March 2019]. Available at: <https://www8.cs.umu.se/kurser/KOGB05/HT04/reports/AI-lab2-Boosting.pdf>
- [29] Zhan, Zhijun. Du, Wenliang. Proceeding CRPIT '14 Proceedings of the IEEE international conference on Privacy, security and data mining. 2002;14: 1-8. Figure 3. An Example Decision Tree, s.7 (Accessed 1 February 2019)
- [30] Roe P, Byron. Yang, Hai-Jun. Zhu, Ji. Liu, Yong. Stancu, Ion. McGregor, Gordon. Boosted decision trees as an alternative to artificial neural networks for particle identification [Internet]. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2005;543 (2-3) (Accessed 16 February 2019)
- [31] Kent State University, "SPSS Tutorials: Pearson Correlation", 2019. Available at: <https://libguides.library.kent.edu/SPSS/PearsonCorr> (Accessed 07 February 2019)
- [32] P. Nikolaos," The chi-square test", vol. 150, pp. 898-899, 2016. Available at: <https://www-sciencedirect-com.ezproxy.hkr.se/science/article/pii/S0889540616304498> (Accessed 12 February 2019)
- [33] H. Lee and S. B. Kim," An overlap-sensitive margin classifier for imbalanced and overlapping data," *Expert Systems with Applications*, vol. 98, pp. 72-83, 2018. Available at: <https://www-sciencedirect-com/science/article/pii/S0957417418300071> (Accessed 12 February 2019)
- [34] Microsoft Azure, "How to evaluate model performance in Azure Machine Learning Studio" 2019, Available at: <https://docs.microsoft.com/en-us/azure/machine-learning/studio/evaluate-model-performance> (Accessed 28 March 2019)
- [35] Microsoft Azure, "Tune Model Hyperparameters" 2019, Available at: <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/tune-model-hyperparameters> (Accessed 28 March 2019)

Appendix

Diabetes test > Evaluate Model > Evaluation results



Figure 1 shows how results are presented in Azure machine learning studio. Figure describes the accuracy achieved by neural network. True positives/False negative present how many correct non-readmittance patients were correctly predicted and incorrectly predicted. True negative/False positive presents how many readmittance patients were correctly predicted and incorrectly predicted.

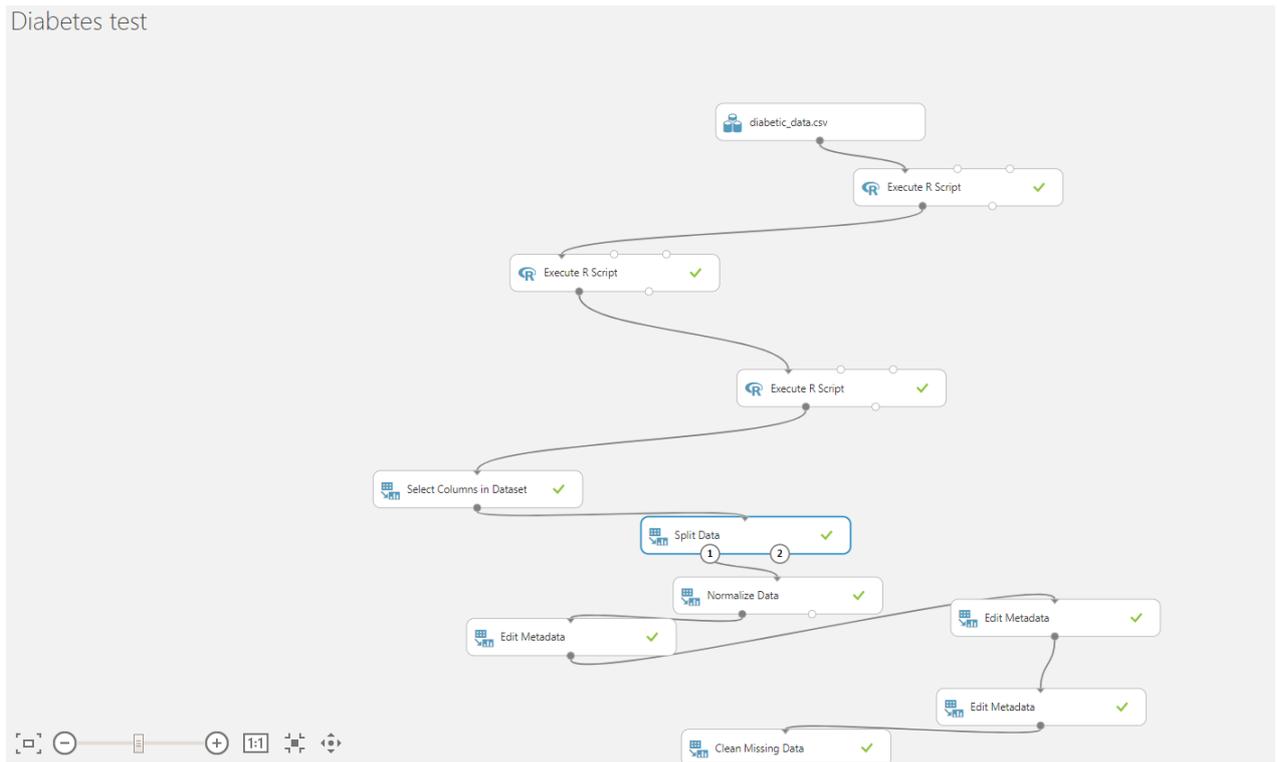


Figure 2 displays how Azure machine learning studio works. Azure machine learning studio provides drag-and-drop function that allows the user to manipulate the dataset through different components such as “split data” which was used to exclude unwanted data, “Select Columns in Dataset” to exclude unwanted columns within the data, “Execute R Script” which was used to transform the dataset into a two-class dataset instead of multiclass.

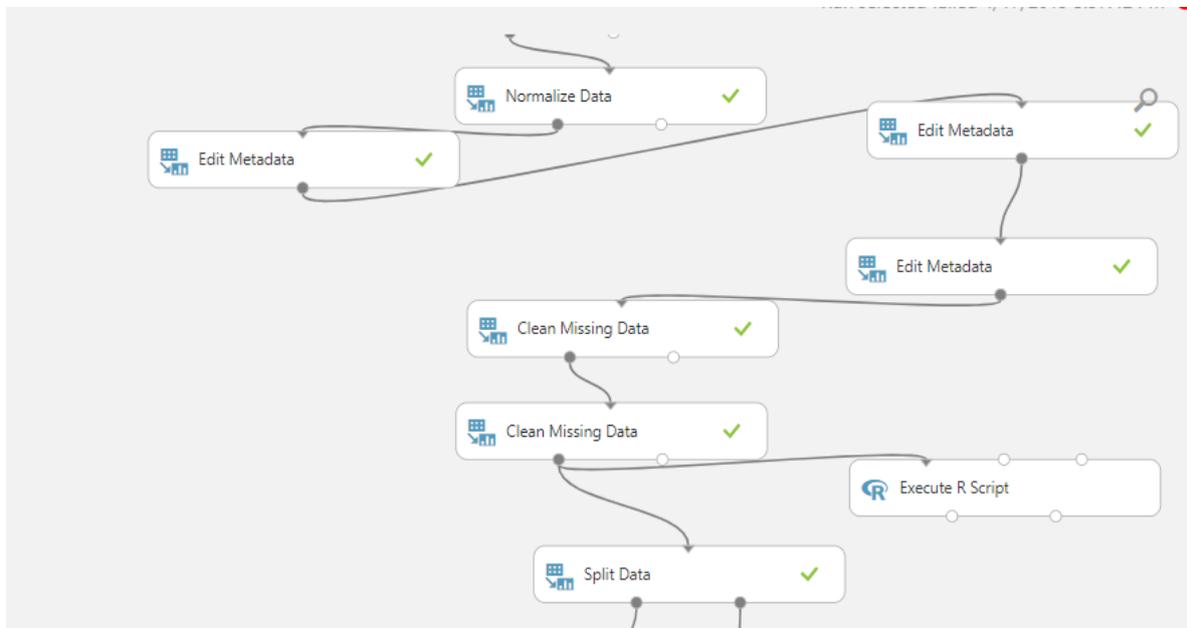


Figure 3 Overview of drag-and-drop components such as Edit Metadata that was used to convert all string values into categorical values and Clean Missing Data that was used to clean all numerical and categorical values.

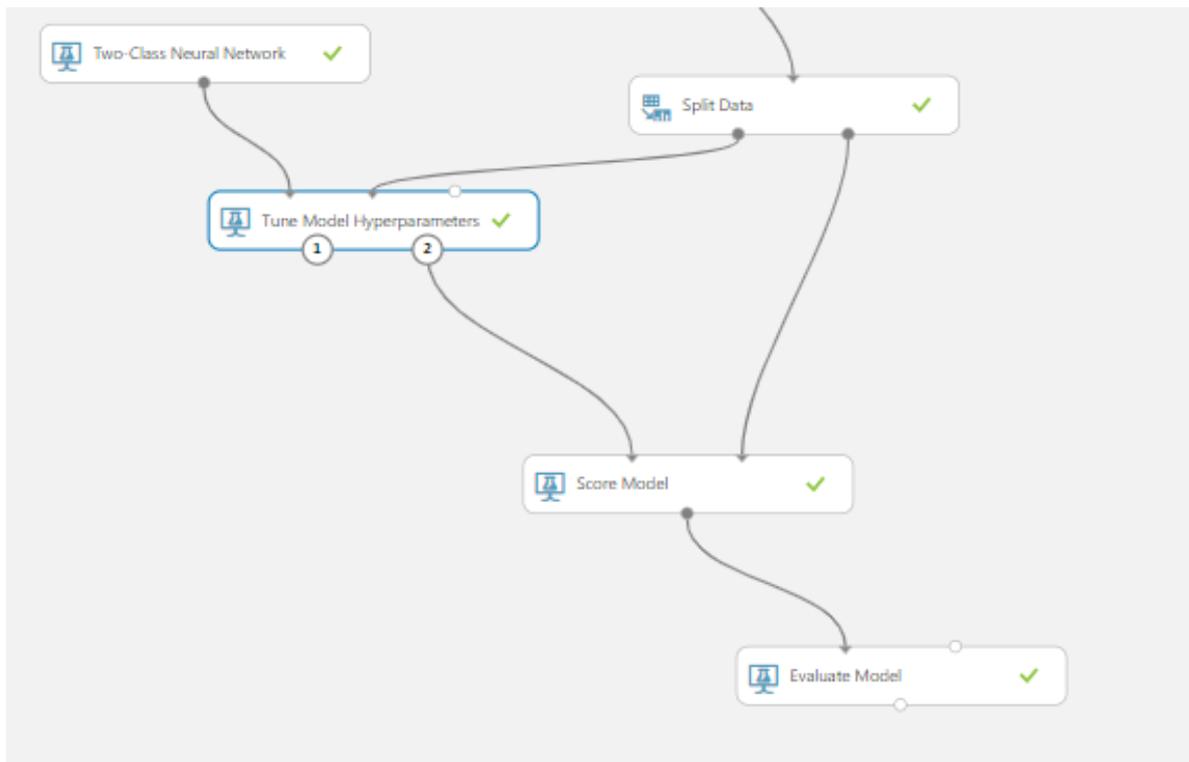


Figure 4 shows how dataset is connected to an algorithm. “Split Data” components allow users to split the data into training and testing data and then connect the data to a train model called “Tune Model Hyperparameters”. “Two-Class Neural Network” then uses this data to train itself and provides visual results in Score Model and Evaluate Model component.

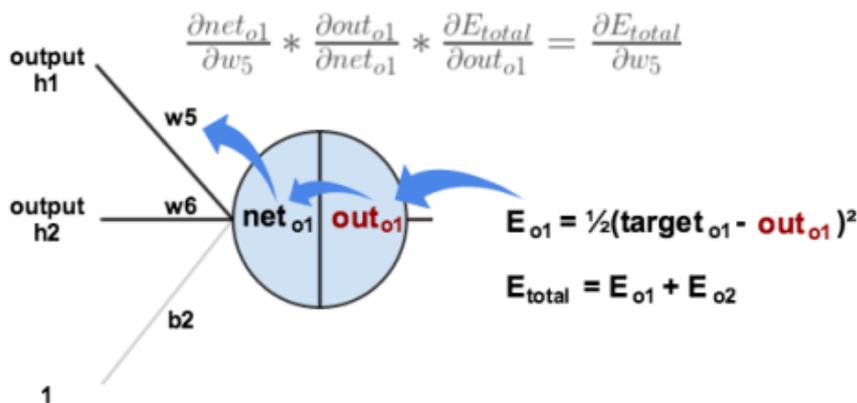


Figure 5 showing an output neuron in an ANN when backpropagation is performed and how much weight number 5 (w_5) contributes to the total error of the ANN ($\frac{\partial E_{total}}{\partial w_5}$). First, the deviation between the desired ($target_{o1}$) output and the actual output (out_{o1}) the neuron produce is calculated, the right arrow illustrate the first step together with the Formula. The total error from o_1 and o_2 is then calculated (E_{total}). The next step is to see how much the output of the neuron change compared to the total net input to the neuron. The third step is to

see how much the net input to the neuron change compared to the weights, in this illustration w_5 is the chosen weight. There will be three different values obtained and these are then finally put together with the Formula placed at the upper part of the Figure. Breaking down the chain rule Formula the $\frac{\partial net_{o1}}{\partial w_5}$ is how much the net input to o_1 change compared to w_5 . The $\frac{\partial out_{o1}}{\partial net_{o1}}$ is how much the output (out_{o1}) change to the total net input (net_{o1}). Finally, the $\frac{\partial E_{total}}{\partial out_{o1}}$ is the how much the total error from the output change with respect to the output (o_1).

R Script

```

1 # Map 1-based optional input ports to variables
2 dataset1 <- maml.mapInputPort(1) # class: data.frame
3
4 dataset1$readmitted[dataset1$readmitted==">30"]<-"NO"
5 ##Set greater than than 30 days to NO since >30 days is assumed to
6 ##not need readmission
7
8 maml.mapOutputPort("dataset1");

```

Figure 6 showing R-script for transforming multiclass into two-class

```

rows_pos = df_train_all.OUTPUT_LABEL == 1
df_train_pos = df_train_all.loc[rows_pos]
df_train_neg = df_train_all.loc[~rows_pos]

df_train = pd.concat([df_train_pos, df_train_neg.sample(n = len(df_train_pos), random_state = 42)], axis = 0)

df_train = df_train.sample(n = len(df_train), random_state = 42).reset_index(drop = True)

print('Train balanced prevalence(n = %d):%.3f'%(len(df_train), calc_prevalence(df_train.OUTPUT_LABEL.values)))

```

Figure 7 Shows the process of taking out readmittance rows “df_train_pos” and non-readmittance “df_train_neg” patients. The data is then balanced out by picking as many rows from non-readmittance as there is readmittance patients.

readmitted	number_diagnoses	number_inpatient	number_outpatient	number_emergency	admission_type_id	num_medications	age
1	0.248337	0.242717	0.179141	0.150791	0.135295	0.123278	0.107728

admission_source_id	discharge_disposition_id	num_procedures	num_lab_procedures	time_in_hospital
0.094978	0.058693	0.057891	0.053883	0.017752

Figure 8 Showing full results from Pearson’s correlation test with numerical values

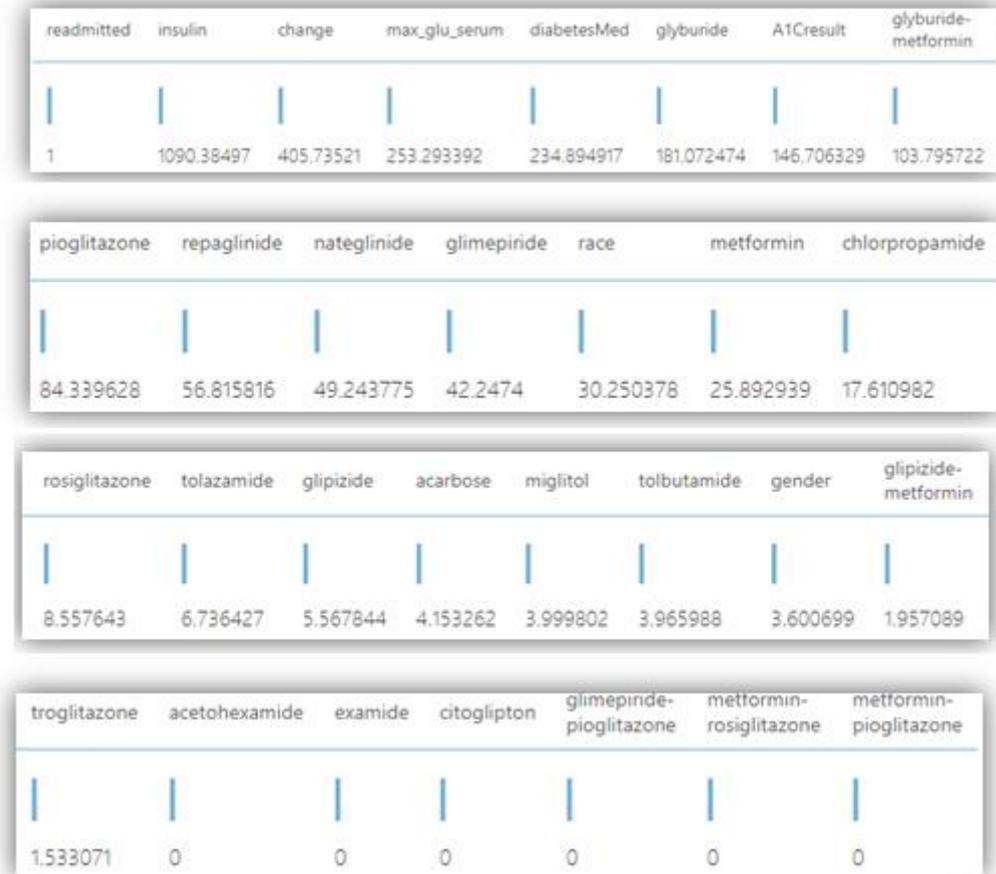


Figure 9 Showing full results from a Chi-Squared correlation test with categorical values

Time table	K-nn	ANN	BDT	LR
Round 1	0:02:23	0:10:37.918	0:02:17.725	0:00:25.377
Round 2	0:02:04	0:12:20.997	0:02:31.676	0:00:38.721
Round 3	0:02:00	0:12:06.315	0:02:19.990	0:00:38.234
Round 4	0:02:01	0:11:29.295	0:02:24.491	0:00:09.016
Round 5	0:01:59	0:12:06.888	0:02:20.057	0:00:40.864
Round 6	0:02:08	0:13:14.718	0:02:25.976	0:00:40.316
Round 7	0:02:03	0:11:42.860	0:02:33.854	0:00:37.743
Round 8	0:01:59	0:11:57.583	0:02:32.337	0:00:34.128
Round 9	0:02:00	0:11:38.665	0:02:24.895	0:00:39.807
Round 10	0:02:00	0:13:05.197	0:02:18.055	0:00:32.891
Round 11	0:01:59	0:12:15.792	0:02:39.002	0:00:35.620
Round 12	0:02:02	0:12:12.656	0:02:29.511	0:00:35.363
Round 13	0:02:07	0:13:05.387	0:02:24.350	0:00:35.798
Round 14	0:02:01	0:12:26.853	0:02:30.889	0:00:35.486
Round 15	0:01:59	0:12:18.675	0:02:29.195	0:00:33.544

Table 1 shows elapsed time for all test rounds

