KTH Royal Institute of Technology
School of Electrical Engineering and Computer Science (EECS)
SE-100 44 Stockholm, Sweden

Bachelor Thesis

Predicting the Flow of Patients in the Dental Industry

—

Michael Arenander
Mustafa Ali

Examiner: Anders Väsberg
Supervisor: Morteza Esmaeili Tavana

Abstract

In this paper, ML models are applied to discover the predictability of show-ups among patients visiting a specific company, Distriktstandvården. The motivation is to enable overbooking of scheduled time-blocks to increase the efficiency of the company's use of resources. To achieve such efficiency, patient predictability has to be investigated and used. The application of the company's internal data to train and test a machine learning model is presented and assessed in this paper. The result was that despite there being a large quantity of contributing parameters as well as a vast database to train on, the quality of the parameters and complexity of the task proved to only contribute to a precision quality of 69.3%, where 75% was deemed a desirable target. Despite the efforts in optimizing the model, and with the testing of alternative model candidates, the models did not discover strong links using the provided data for predicting patient show-up.

Keywords: Algorithms, artificial intelligence, dental care, dentist, machine learning

Sammanfattning

Maskininlärningsmodeller appliceras i denna rapport för att utforska förutsägbarheten av att patienter dyker upp på bokade besök hos företaget Distriktstandvården. Motivationen till utförandet av rapporten är att möjliggöra överbokning av schemalagda besök för att öka företagets inre effektivitet. För att uppnå en sådan effektivitet behöver förutsägbarheten av att patienter dyker upp utforskas och tillämpas. Användandet av företagets interna data i syfte att träna och testa en maskininlärningsmodell presenteras och bedöms därför i denna rapport. Resultatet var att trots tillgängligheten av en stor samling av relevant data, så uppnåddes endast en precision på förutsägbarheten till 69,3% där 75% var satt som mål. Starka bidragande mönster för maskininlärningsprocessen hittades inte inom det tillhandahållna datat trots extensiva optimeringsförsök av modellen i samband med testandet av flertalet alternativ till modellen.

Nyckelord: Algoritmer, artificiell intelligens, tandvård, tandläkare, maskininlärning

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ML	Machine learning
NN	Neural Network
RFT	Random Forest Tree
API	Application Programming Interface
JRIP	Joint Reserve Intelligence Program
SVM	support vector machine
VFDT	very fast decision trees

1 Introduction

With the computing power of today's microprocessors, artificial intelligence has become more relevant in business than it has ever been before [1]. Using various models that have been developed and researched, such as machine learning systems and different predictors, computers have become faster and more efficient at solving certain complex problems; resulting in a wider field and scope of application for the work computers can provide [2].

The company, Distriktstandvården, which specializes in dental care throughout Stockholm, requested our help to develop the company's business by combining computing power with various popular prediction models. One problem with managing clinics comes from people being required to be physically present for their appointments due to the risk of people not showing up when scheduled. A time-slot that fails to be utilized is an undesired situation for the clinic due to the costs involved in making such a slot ready and available for a client.

The company requested a solution that could optimize the time-slot usage. We saw this request as an opportunity to investigate and provide a solution using ML models by using the company's computation capacity and database. With the help of existing ML models combined with the vast and anonymized database, we saw an opportunity to provide a solution to enable overbooking of clinic time-slots to maximize profits, in a similar fashion as what airline companies perform with their overbooking strategies [3].

To address this, a model designed to find relevant and influential differences is needed. The model built for the Distriktstandvården project is specialized to the anonymous data of the company's patients which they can provide, and as such, is only compatible with such an environment.

In summary, the purpose of this report is to build a ML model that predicts if a client will turn up on their scheduled time based on different, yet anonymized, characteristics that are available in the company's database.

1.1 Background

AI is a very broad term and has been in development since the second world war [1]. The aim of AI is to create a machine that can mimic the cognitive abilities of human behaviour [4].

1.1.1 Neural Networks

Today's models in the field of AI utilize neural networks in order to mimic how the human brain functions [4]. A digitized NN is a series of algorithms which interact and adapt in order to find relationships in the data provided to it. It tries to fulfil the same functionality that our neurons fulfil in a human brain as best as possible [5]. A NN contains layers of interconnected nodes. The nodes which can be observed in **Figure 1** are divided into three different layers, the input node layer, the hidden layer and the output layer [5].

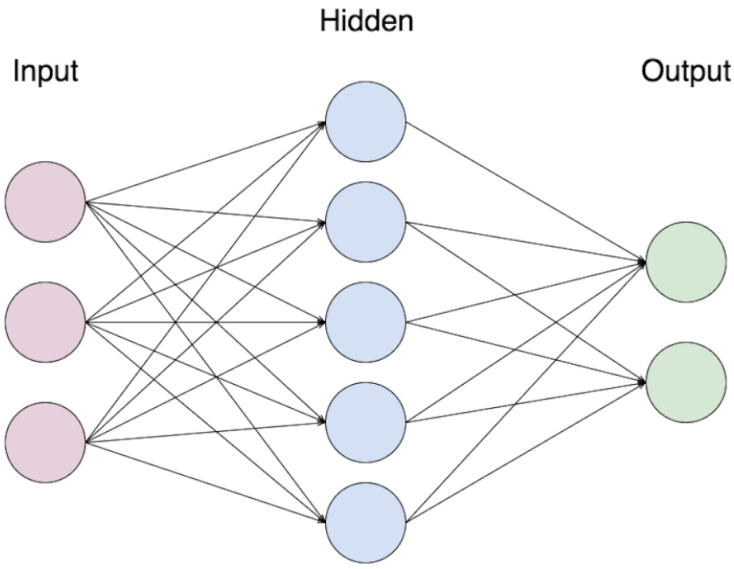

Figure 1: A basic NN model consisting of three layers with different purposes labeled Input, Hidden, and Output. Reprinted from [6].

Figure 1 shows a NN consisting of three layers. The hidden layer typically consists of several nodes. Each of the individual neurons can activate with the help of so-called activation functions; functions that decide whether a neuron will fire, and where depending on the combination of the fired neurons, can decide if others will fire or not. These neurons fire or activate based on calculations on a weighted sum and add a bias based on it [7]. These weights and biases can be adjusted accordingly in order to fit the model.

1.1.2 Machine Learning

ML is a branch of AI as it describes one more specific way of applying the concept of AI using a cyclic learning behaviour that incrementally and rationally arrives at a conclusion using a model [8]. A ML approach to applying AI in practice consists of a few components that need to work together; the model, which is essentially the data structures and functions of the program, the decision function to provide its answer, and an error function which can calibrate the model and thereby incrementally improve the model's ability at achieving its purpose [8].

While there are many emerging areas where ML can be potentially applied and provide benefits, there is a growing demand for solving problems using this technology. This report has decided to focus on one such area where there is a potential application of ML.

1.2 Problem

The company Distriktstandvården which specializes in the dental care industry is having problems with clients not turning up on their scheduled time. The company wants to investigate if it's possible to solve this problem. Patients that do not show up on their scheduled time cost the company a fortune in the long run. Having the knowledge of whether or not a patient will show up on their scheduled time can create a more stable and reliable flow of patients. There is a possible chance that this can be done with the help of AI.

Can we exploit the history of patients' features and behaviors (aligned with the GDPR data protection regulation) with AI tools to predict their future behavior?

1.3 Purpose

If the desired model is achieved, the benefit can be substantial for the company Distriktstandvården. They will in theory be able to create more capital with the help of this ML model in the long run. We will learn more about current implementation and be able to expand our knowledge in this area. Nonetheless, we will also gain valuable experience and contacts for the future.

1.4 Aim

The project aim is to develop a model that will benefit Distriktstandvården's booking system by providing a predicted show-up likelihood.

The aim is divided into three sub-aims. Sub-aim #1 will be to determine and filter out the most relevant, beneficial and achievable models which can be used in combination with the data provided by the company. This will be based on previous studies in the

field of patient flow and application. Sub-aim #2 will be to build a functional prototype which can predict if a patient will come at their scheduled time. Sub-aim #3 will be to improve the model to a prediction quality that can be useful for the company. A target of 75% prediction quality is desired but not required.

1.5 Research Methodologies

This project will first conduct a literature study to identify which algorithm model(s) and design option(s) that are relevant to achieve the goals. The literature study will provide the fundamental building blocks for the algorithm model to start from. The literature study is expected to provide some scientifically-grounded guidance for the development of a prototype. The prototype will then be built using the results collected from the literature study. The prototype will be developed using a ML methodology. The prototype will be trained, tested and iteratively reworked using dummy-data to measure its performance in an environment which is similar to the real-world application. The prototype will then be tested on real data that has been anonymized and tokenized to evaluate the prototype's performance in a real-world situation. The extent in which the prototype can predict a patient's arrival time will be measured by testing the prototype on the real data. Differences between the dummy-data and real data is expected and will be measured as well as evaluated.

1.6 Delimitations

This project will explicitly not be a comparison between various ML models for predicting on-time arrival of patients. Only a limited number of alternatives will be tested, measured and compared. The objective is to find a model that fulfills certain criteria rather than finding the most optimal model.

The training performance will also not be compared and measured as it can differ depending on language and hardware.

2 Background

2.1 The History of Artificial Intelligence

The idea of AI has been around since the second world war, the aim back then was to create a machine that can mimic the cognitive abilities of human behaviour. With time, the machines have slowly been able to imitate the human brain and solve more complex tasks. The period around 1940-1960 is where lots of technological advancement were made in many fields including AI. Warren McCulloch and Walter Pitts were first to produce a mathematical and computer model of the biological neuron, this was the foundation of which AI stands on today [9]. The technology in the field of AI looked promising and developed at a steady state until the 1960s, where computers had little to no memory to conduct large calculations and had difficulty using computer language. Despite all those obstacles there were advances made that are still relevant today, such as solution trees to solve problems: IPL (information process language) and made it possible to write an LTM (logic theorist machine) program which demonstrates mathematical theorem [9].

In 1980 the framework "Expert System" began to enter peoples vision, because of the framework's ability at representing descriptive and behavioral object information. It could handle more complex information and the system was able to compute statistical models in the expert system. Around those years research into NN, ML and statistical learning was the hot-spot for research and development in the field of AI [1]. These expert systems are still used today and the field of AI has also expanded because of the computation power computers have today [1].

2.2 Linear regression

Linear regression is one of the most common and basic algorithms used in ML which originates from mathematical statistics. The model has been around for over 200 years and it studies the relationship between the input and output. If the model has one single input variable then it is referred as single linear regression, if it has multiple then its called multiple linear regression. The algorithm is known for its simplicity which assumes a linear relationship between the input values (x) and the output values (y), as shown in **Figure 2**.

The linear equation assigns one scale factor for each column or input value, this scale factor or column is called a coefficient. Another coefficient is added which gives the regression line another dimension of freedom which allows it to move up and down on a two dimensional plot. An example of a linear regression equation could look like equation 2 which can be observed here:

$$y = bx + c \tag{1}$$

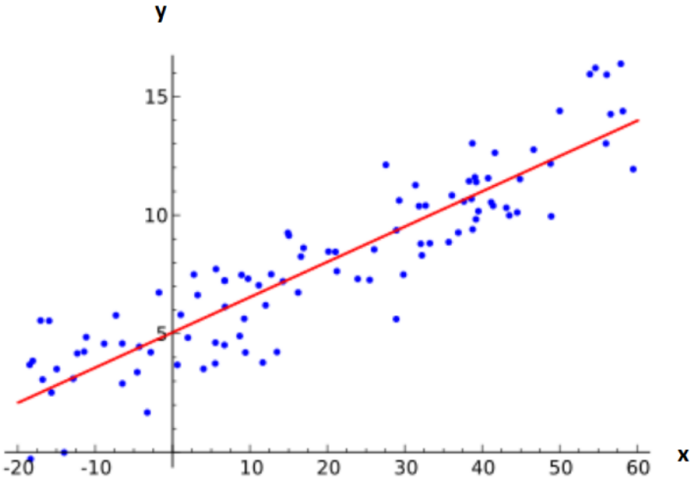

Figure 2: A visualized plot of the linear regression algorithm. Reprinted from [10].

This is a simple regression line with one input in x and one in y. In higher forms of regression there are more inputs in x which vary depending on the input. In higher dimensions the representation is therefore a plane or hyper-plane. It's also common to talk about the complexity of the linear regression depending on the amount of coefficients presented in the regression [11]. The linear regression model represents the best fit line, plane or hyperplane with the data provided to the model; it does so with the help of MSE (mean square error), equation 2. The mean square error rate is a numeric value which is used to represent how good the best fit model is.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \tag{2}$$

The equation is not dependent on the amount of coefficient provided and the result of this equation is a numerical value. The lower this value is the more accurate the model is [12].

2.3 Neural Networks

2.3.1 Neural Network Nodes

NN is a series of artificial neurons that is supposed to mimic the human brain of thinking. These neurons, with the help of algorithms, recognize relationships in data and create a model out of the data presented. This model can then adapt to various inputs in order to present the best possible result. A NN is represented by nodes and edges or so called links in between the nodes [5]. At each node the input data is processed and the output at each node is passed to the adjacent node. How a singular node works is it takes several perceptions and then outputs a single binary output.

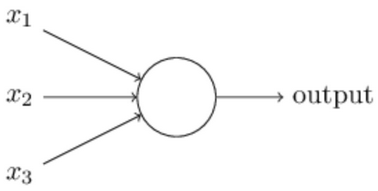

Figure 3: Neuron in a NN taking in three perceptrons. Reprinted from [13].

As shown in **Figure 3** the neuron is taking three perceptrons x1, x2, x3 and producing a single output after the neuron is done processing the perceptrons. These perceptrons can either be inputs from the system but they can also be outputs from adjacent nodes. These perceptrons are presented with a so-called weight which is a numerical representation of the importance of these perceptrons. The neuron's output is either 0 or 1 depending on if the sum of the weights from the perceptron are larger than some threshold value, this value is also called an activation function. Depending on the weight sum, if the weighted sum exceeds the threshold value, the neuron will output the binary value 1 that will be forwarded as the input to the adjacent node, as long as it is not the output node. Before the value is inserted in the so-called activation function, we add a bias. The bias can be represented as how easy it is for the neuron to output a 1. By varying the weights and thresholds we can build different NN models [14].

The threshold value mentioned earlier, also called an activation function, is what determines the output for a single node. After all the weights that come into a node are summed together, and a bias is added to the sum, it is inserted into the activation function in order to be evaluated in some way. The total sum of an input to a node might be anywhere between negative infinity and infinity. Hence, the neuron does not really know how to bind to a value, and therefore is unable to decide if it should fire or not. This is where the activation function is useful, it helps the neuron to decide when it should fire. There are also several types of activation function that work slightly differently, and with their own characteristics [15].

2.3.2 Neural Network Layers

In a NN there are three different types of layers, the input layer, the hidden layer and the output layer. The hidden layer is between the input layer and the output layer, and depending on how the neurons are structured, a model design is constructed. There can be one or more hidden layers, the hidden layer is where most of the calculations for the model are completed. In simple terms the layer performs nonlinear transformations of the inputs entered into the network [16].

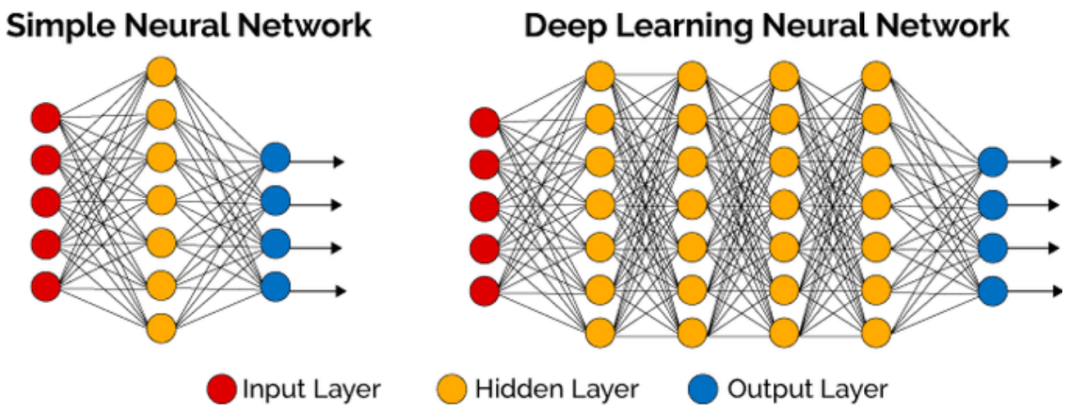

Figure 4: A visualization of the differences between a simple neural network and a multilayered neural network. Reprinted from [17].

If a network contains one hidden layer it is classified as a simple NN, if several then it is classified as a deep learning NN [18]. There is a difference between the two, deep learning is more complicated but can eliminate some of the manual human intervention required to enable large datasets. For simple NN, the data needs to be more structured, resulting in the model needing human intervention by surveying the data to ensure it is relevant before it is inserted into the model. With deep learning, the NN can teach itself what characteristics are important for the model, and determine a hierarchy of the characteristics. Deep NN with several hidden layers can also be leveraged with a labeled dataset, and it can teach itself more accurately [19]. There are two basic ways to teach a model; through supervised learning, and unsupervised learning. The main difference between the two is that within supervised learning, labeled data is used to help predict the outcome, while the other does not.

2.4 Types of Learning

2.4.1 Supervised Learning

Supervised learning is a ML approach that uses labeled data when training the algorithm or model and "supervises" the algorithm to predict more accurately. This is done by using labeled inputs and labeled outputs, and with the help of the labels the model can check its accuracy and learn over time. Supervised learning algorithms can be divided into two main categories; classification and regression.

Classification algorithms are excellent for when a model that classifies an object accurately is needed. They are often used to accurately assign test data into different categories. Examples of the most common classification algorithms are random forest, and support vector machines, among others.

Regression is also another supervised learning method that uses labeled data, but it is used to find relationships. Regression algorithms are experts in using labeled data in

order to find relationships between dependent variables and independent variables. The most common regression algorithms are logistic regression, and linear regression, among others [20].

2.4.2 Unsupervised Learning

Unsupervised learning uses algorithms in order to analyze and cluster unlabeled data. This type of learning for a model teaches the NN how to find patterns in the data without supervision, hence the name unsupervised. There are three main ways of doing so; clustering, association, and dimensionality reduction. Clustering is a method where the model learns how to group unlabeled data together based on their similarities or differences. Association is the method where the model with the help of algorithms tries to find relationships in the variables in the unlabeled dataset. The last method in the unsupervised learning category is dimensionality reduction; this method is often used in the pre-processing stage of an implementation. It is used when the number of features in the datasets is too high, it maps the dataset features into a manageable size while at the same time not losing the integrity of the data [20].

2.4.3 Semi-supervised Learning

There is also a category called semi-supervised learning which combines supervised learning and unsupervised learning. This category is used when the dataset is both labeled and unlabeled. Semi-supervised learning uses algorithms that are good at extracting relevant features from data [20].

2.5 Overview of Machine Learning Models

As seen in **Figure 5**, there are many different algorithms that can be used depending on the task. These models are the most common ones that are used in the field of ML for both supervised learning and unsupervised learning. Each algorithm can be used for a specific type of area depending on the data available and what is required from the model, and each of these models are advantageous in their own way.

When choosing a model to build, the important factor is what type of data is available and if there are any specific requirements that should be met. An example could be that the system should be building a classification model once, and where the memory size does not matter, then a viable model option could be a deep NN. The most important factor when choosing a model is to choose it based on the requirements that it wants to meet, and what data is available.

Algo	Type	Tolerance number features	Parametrization	Memory size	Minimal required quantity	Com m	Overfitting Tendency	Difficulty	Time for Learning	Time for predicting
Linear Regression	R	Weak	Weak	Small	Small	++	Low	Weak	Weak	Weak
Logistic Regression	C	Weak	Simple	Small	Small	++	Low	Weak	Weak	Weak
Decision Tree	R & C	Strong	Simple / intuitive	Large	Small	+++	Very high	Weak	Weak	Weak
Random Forest	R & C	Strong	Simple / intuitive	Very Large	Large	++	Average	Average	Costly	Costly
Boosting	R & C	Strong	Simple / intuitive	Very Large	Large	+	Average	Average	Costly	Weak
Naive Bayes	C	Weak	No params.	Small	Small	++	Low	Weak	Weak	Weak
SVM	C	Very strong	Not intuitive	Small	Large	--	Average	High	Costly	Weak
Neural Network (NN)	C	Very strong **	Not intuitive	Inter	Large	---	Average	Very high	Costly	Weak
Deep Neural Network	C	Very strong **	Not intuitive	Very Large	Very Large	---	High	Very high	Very costly	Weak
K-Means	CL*	Strong	Simple / Intuitive		Small	+		High	Weak	
One class SVM	A	Very strong	Not intuitive	Weak	Large	--	Average	High	Costly	Weak

Figure 5: An image of a table listing the most common classification models. Reprinted from [21].

2.6 Related Work

2.6.1 Use of AI for Improving Patient Flow and Healthcare Delivery

A study published by the journal of computer science and system biology investigated if AI could improve the patient flow in the healthcare industry. What was concluded from that study is that with the help of AI in the emergency department of healthcare, the average stay of a patient could be reduced by 15

This improved the operational efficiency of the emergency department, which made the distribution of the hospital's resources easier. This led to more structure within the hospital and an increase in patient satisfaction. In order for the AI to achieve this it used nine different ML algorithms which included, NN, support vector machine (SVM), random forest, elastic net, multivariate adaptive regression splines, gradient boosting machine, bagging, Kth nearest neighbor, classification and regression trees. All these algorithms were used in order to achieve the model desired with the data available [22].

2.6.2 Real-Time Capacity Management and Patient Flow Optimization in Hospitals Using AI Methods

A study published about management and patient flow optimization in hospitals using AI methods, concluded that working with AI in hospitals can improve patient flow. In simple terms, hospitals are divided into different departments which manage their own queues. Patients may be sent to different departments depending on different factors and because of the lack of synchronization between the departments, resulting in an inefficient patient flow. With the help of the implemented AI to help balance the distribution of resources, there were clear improvements. There was an improvement in waiting time in patient flow by 56% and an improvement in resource utilization by 8.3% [23].

2.6.3 Prediction of hospital no-show appointments through AI algorithms

In a study done in 2019 about patients who will show up on scheduled appointments in the US. In the study, acknowledging a "no show" to an appointment means that the patient did not show up on their scheduled time, cancels within 24 hours of the appointment or is 60 minutes late for a scheduled appointment. In the study two main methods were used in order to write the algorithms, Hoeffding trees and JRIP. In the report there were Hoeffding trees which are known as very fast decision trees (VFDTs) and which have the assumption that the data does not vary much. JRIP (Joint Reserve Intelligence Program) is another covered algorithm which uses a separate and conquering learning algorithm and performs efficiently on scattered datasets.

These models were trained on a dataset consisting of 1 087 979 records, of which 123 299 were no show records. In the study they presented the most significant factor

that helped the AI train and predict the right answer which was if the patient has a history of no show appointments for both algorithms. The JRIP model had a prediction rate of 76.44% and the Hoeffding trees had a prediction rate of 77.13%. The JRIP model despite its competitiveness is not ideal for large datasets like Hoeffding trees which is why Hoeffding trees is a better option for large datasets [24].

2.7 General Data Protection Regulation

GDPR (General Data Protection Regulation) laws do not affect data which has been anonymized to the extent where individuals can not be identified, even by using comparative analysis of multiple datasets [25]. This means that ML can be used to process any data legally that has been anonymized and tokenized to the extent where the data cannot be reverse engineered to the extent where individuals can be identified.

3 Methodology

3.1 Research process

The prior research conducted included many hours of reading into ML and what AI is and how advanced this technology is. In the research conducted, articles were read about ML, classification algorithms, models available online, libraries available for programming, programming languages that support ML, and more relevant info that could benefit the process. In this research, a crucial part was the reviewing of potentially similar research that had already been conducted within the same area of research.

3.2 Data collection and GDPR

The data that was used in the work and the models were all provided by the company, which is directly derived from real patient data. The company that provided us the information provided real patients data to be used in the development of the ML model. Because of ethical, social and legal reasons the data provided did not include any personal data. No names, family trees, social security number, personal address or personal information of some sort was provided. All data was also anonymized and tokenized to mitigate the ability of deducing identities based on the data received, and also to optimize the size of the data elements. GDPR data laws do not affect data which has been anonymized to the extent where individuals can not be identified, even by using comparative analysis of multiple datasets [25]. When building and testing different models, for ethical reasons there was no personal information used or provided to us by the company, thereby ensuring laws were followed.

3.2.1 Sampling

The following data that was provided included a few characteristics which were expected to have some impact on the model.

Actor ID: This is like a social security number, each unique patient has its own actor id number, which is represented by an integer number.

Business ID: The business ID is which clinic the patient visited, each clinic has its own integer number.

Status: This parameter gives us the information if the patient came to the visited appointment. This is what was used as the target parameter for the model to learn and predict. A value of zero signifies that the patient did not show up to the scheduled appointment, and a value of one represents that the patient did show up.

Birth Year: This parameter informs us of the birth year of the patient, this parameter only includes the birth year and no other digits from the social security number.

The tokenized version of this parameter first trimmed the year and then scaled all birth years by a random value.

Regdate: This parameter gives us the time and date of when the appointment took place. This value was converted to an integer and then tokenized to mitigate reverse look-up by trimming all date integers by a random value. This value was later split into year, month, day and hour-components with a random-value shift on each to discover isolated patterns.

Duration: The duration of the appointment in minutes, rounded to the nearest 10-minute mark.

Postcode: The postcode of the patient, this parameter did not include any type of personal address, just the region of where the patient lived. This was the only digit in the form of a string but was converted into an integer. This parameter was tokenized by applying a random value shift.

Settobecalled: This parameter informs the model if the patient was called from the company for an appointment or if the patient made the appointment themselves. It was represented by an integer with value zero for no call, and a value of one for when a call was made.

An example-row of how the data did look like before tokenizing parameters:

ActorID	**BusinessID**	**Status**	**Birthyear**	**RegDate**	**Duration**	**BookingCategoryName**	**SetToBeCalled**
321949	197296	1	1961	2022-05-07 14:58:38	40	16965	0

3.2.2 Sample size

The sample size that the model was trained on was precisely 1 million datasets, also called entries or reports in this paper. 1 million datasets was used for the reason that predictions made with data used up to 100,000 datasets had vastly varying results depending on the randomization of the data. The deviation was not measured, but it was judged on-site by us that values over 100,000 datasets produced results for most models which did not change by much despite changing the data randomization seed value. The decision of running with 1 million datasets was thus a trade-off between maintaining a safe margin, combined with the additional processing time which this would cause when running the data through the model. 80% of the 1 million entries was used for training the model and 20% was used for validation of the model.

3.2.3 Target population

This project was not targeted to a specific age or gender, this project was to help the company Distriktstandvården find possible parameters of what no-show patients might have in common, if anything. If successful, the model should be able to train using non-tokenized real data, and then predict the status of the patient in advance, before the patient's scheduled time.

3.3 Experimental Design/Planned Measurements

3.3.1 Test models

In order to enable replication of what was achieved when testing, getting an understanding about what classification algorithms are, and what can be achieved with each of them was required. This knowledge is needed in order to understand what the algorithms can achieve and what the result is displaying. As the use of the models was understood, we could proceed to the integration and application of the models. The model's objective was to predict if a patient will show up on their scheduled time. To perform this more efficiently, a python library was used. Using the SciKit-learn library, the implementation judged to be the most reasonable to begin with for this project was to start with a random forest tree. This implementation was done with the mentioned SciKit-learn library, using basic knowledge of programming in python and by reading the library-documentation on it's website.

The data needed restructuring in order to fit the algorithms in the SciKit-learn library. Some knowledge in data structure conversion in python and text file input and output was essential to make the data fit the library's demands. The SciKit-learn library also provided various other NN models that could be tested without much additional work which was advantageous. Another advantage with the SciKit-learn library was the possibility of running models using multi-threading when applicable, resulting in shorter time spent training.

3.3.2 Hardware and software used

The software used was done using a computer and the programming language python, there were two libraries used. The first one is the library Pytorch which is an open-source library which specializes in ML through visual input data. Pytorch did not fit the environment which we were trying to build a model for. The second one is the SciKit-learn library which is a free open-source library that is used for a larger variety of ML use-cases. This library was discovered to have some of the model presets ready, which were highly relevant for our specific use-case.

3.4 Assessing Reliability and Validity of the Data Collection

3.4.1 Reliability and validity

When building models in ML, reliability and validity are two important aspects. The models were built on 80% of the 1,000,000 datasets available and tested on the rest 20% of data. With the help of the test prediction and using a simple mathematical function we can predict a numerical percentage of how accurately the model predicts.

$$\frac{\text{Amount of right predictions}}{\text{Total predictions}} = (result * 100) \tag{3}$$

When computing the test, the model can also be compared to the accuracy matrix, which informs how the model is predicting and if there is a bias that should be accounted for and addressed.

4 Implementation

4.1 Preparation

In this thesis, when predicting the flow of patients in dental care, there were a few methods used in the process. What was required from us students was to examine and create a model that could predict if a patient that has a booked time would show up at their scheduled time or not. The model should predict with a show-up confidence percentage as the output, if a patient would arrive at the scheduled time, if it was possible to create such a system. This was the demand from the company to examine the possibilities and create a model for them.

As neither of us students have a degree in ML, the first task was to prepare by reading articles and learn about ML models in general, AI and how to apply them using a specific type of data and purpose. For the first few weeks of the project, the goal was to read up on AI and read about articles of what has been done before with this technology as well as reviewing and practicing the programming language and library that were of interest.

By reading about the possibilities of AI and ML and about what has been done before, there was strong evidence that this was possible at the very least. The following concern was to judge how hard it would be to implement something that advanced with minimal prior knowledge with programming a ML model that could produce a result for our specific situation.

4.2 Creation and Evaluation of the Prototype

The next phase of the project was to create something small and scale it up, one of the simplest ML models was to create a linear regression model. The model was created with the programming language python using the library Pytorch which was known from our prior research. That would give us an estimate on the complexity of the project from a programming standpoint but also give us some result. The model was created with "dummy" data at first and then applied into anonymized and tokenized real patient data which was provided by the company.

After programming a simple model with the help of a library, we could work on improving its performance. The next step was to test the data provided from the company, Distriktstandvården, on one of the more complex and more accurate ML models. The product should be a classification model that could classify a patient into one of two groups, either they would show up or not. Upon studying about models that could be useful for the data provided, it looked like a random tree was the best choice to implement using the library Scikit-learn. The library was known upon prior investigation on what was available online. It had an implementation of random forest trees that could be

heavily modified to produce various versions of the same model which all optimized to different levels that were advantageous or disadvantageous depending on our assessment of it's behaviour when looking at the testing results.

The library also included an easy implementation of an accuracy matrix model that could visually inform how the model is predicting. Additionally, the library also included an implementation for "info gain ranking", which in simple terms means that it visually shows how important each attribute in the data is and how it helps the model make predictions.

4.3 Prototype Varieties

Upon understanding that the models from the library Scikit-learn are easy to implement with some prior knowledge of programming, the rest was also implemented and tested for the data. The most common supervised classification models were implemented such as random forest, logistic regression, naive bayes, gaussian process, and a few more. There were also variations of these implementations such as an open random forest tree, balanced random forest tree, gaussian isometric, and many other variants. The implementation was not very challenging with some prior knowledge to programming in python combined with excessive reading of the library APIs.

5 Result

Upon building different models and testing the models the results presented were conclusive. The models found relationships between the data and could predict if a patient would show up on their scheduled time or not based on the data available. Different models found different relationships in the data-sets, their prediction accuracy was slightly different. Every model implemented was built with every Cartesian product of the data, the models were built and tested on every combination of the data available. This was to ensure that the best model predictor could be harvested. In total 9 different models were tested and provided a valid result that will be presented. All models that were implemented and tested were checked with an accuracy matrix which provided us with the knowledge that the model did not just guess and actually made a prediction. There was no bias to the model which made it predict a certain way when uncertain of the outcome and actually reviewed the data.

An example-row of how the data did look like before tokenizing parameters:

The random forest models implemented and their score predictions

RFT	**RFT-Balanced**	**RFT-open**	**RFT-Custom**
0.6296	0.6335	0.6928	0.6511

Table 1: Random Forest Tree models

As shown in table 1, 4 different random forest trees were implemented and tested. These trees took the longest time to test with every permutation of the data. The best result was given by the open random forest tree from all the different RFT, which gave around 69.3%. Open RFT also took the longest to build and test, it took about 13h to train and validate 512 different models. The custom model was a model that was provided by the library SciKit-learn and could be manipulated to the desired tree. Almost all attributes in the tree could be manipulated, the depth, size, leaves and structure in general could be manipulated. In total there were 19 attributes that could be changed into different values that could reshape the tree in the library.

The Naive Bayes Gaussian models implemented and their score predictions

Gauss	**Gauss-iso**	**Gauss-sig**
0.6056	0.6157	0.6061

Table 2: Naive Bayes Gaussian models

The three different naive bayes models that were implemented and tested, were all variation models of the original naive bayes gaussian model. These differences were also available in the Scikit-learn library which had easy implementation. The best of the three

models gaussian-iso gave a prediction score of around 61.6% which was a reliable score but not better than the open random forest.

The Logistic regression models implemented and their score predictions

Logreg	**Logreg-CV**
0.6064	0.6105

Table 3: Logistic regression models

The logistic regression models were implemented with the same library and with the same simplicity. The best model was the variation of the original logistic regression model which gave a prediction score of around 61.0%.

5.1 Summary of results

After performing extensive training and testing on various models from the SciKit-learner library with different optimization parameters, the best predictor for our specific environment was the open RFT which produced a prediction score of 69.3% when tested on real data.

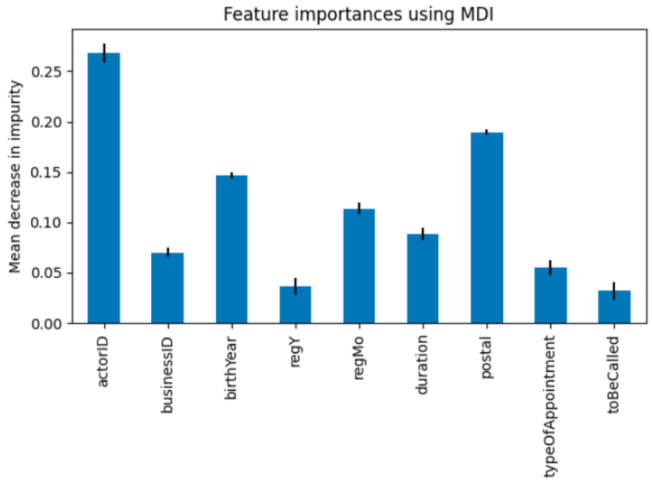

Figure 6: Feature importance ranking of the open RFT with a prediction score of 69.3%. Original source [26].

All models trained did not use all the attributes provided to the model in order to create the best models. For instance Gauss-isometric only used three out of the nine attributes in order to create a model that had an accuracy of around 62%. The open random forest used all the attributes in order to create its model and had a score prediction accuracy of 69.3%. This model gained the most information out of the actor ID which is an individual ID for each person, just like the social security number, the difference is that the id is not connected to a name or any personal information. Other important

attributes in our model were the postal code of the individual and the birth year of the individual.

In the problem formulation, the question asked was: "Can we exploit the history of patients' features and behaviors (aligned with the GDPR data protection regulation) with AI tools to predict their future behavior?". In short the answer is yes, to some extent. The results show that is possible in the case of Distriktstandvården where a model with a prediction accuracy of 69.3% was built. To what extent it fulfills this question will be discussed in the following chapter.

6 Analysis

6.1 Model Bias

Our problem formulation mentioned if there was any way to predict the flow of patients and an open random forest tree with a prediction accuracy of 69.3%. This model with closer inspection, does predictions and is valid, as can be seen from the accuracy matrix in **Figure 7**.

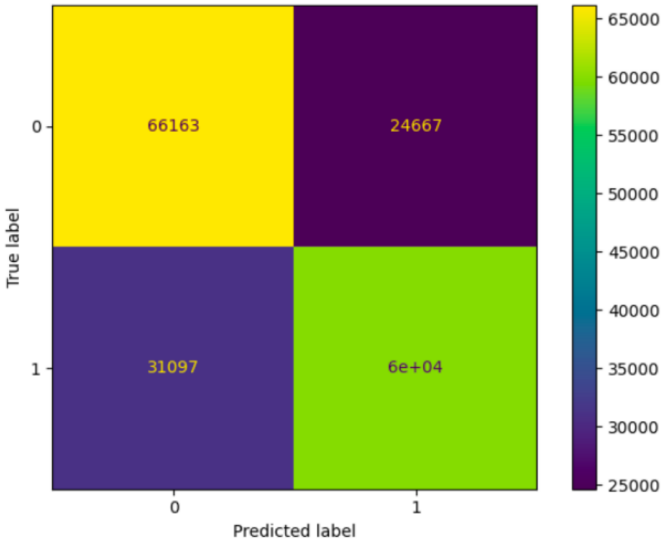

Figure 7: Accuracy matrix of the open RFT with a prediction score of 69.3%. Original source [26].

The accuracy matrix gives information of how the model predicts, the outcome can either be correct or wrong. An output could either be a false positive or a false negative if the model's prediction is not distributed between the false positive and negative evenly, it would indicate that the model has a bias. If the model does have a bias then the model is guessing at the most common output instead of predicting. To combat biased guessing, a balancing of the data needs to be introduced. A balancing of data means adding data which has the least common outputs into the dataset in order to combat the bias. In our case, a perfect balancing algorithm was programmed to do that automatically.

In **Figure 7**, there is an observable matrix of the model for the open random forest tree, which received a prediction score of 69.3%. In 31,097 of the cases the algorithm predicted false positives, and in 24,667 of the cases it predicted false negatives.

6.2 Analysis of the Data

6.2.1 Speculations regarding the Input Data Results

Predicting if a patient will show up on a scheduled dentist time is challenging, generalizing individual behaviour is hard. That is shown in **Figure 6** where the actor ID was the most important attribute in creating the model.

The location where the patient lives is also another significant attribute to why patients do not show up according to **Figure 6**. There could be several location-related factors that increase the no show rate; the address of the patient might be far away from their registered clinic or the area could be a low income area making dental care less prioritized or not even affordable.

The age of the patient also contributes to an extradited appointment, older patients might be less likely to go to the dentist for several reasons. The model found relationships between the patient and the duration of the appointment which contributed to its decision making. An interesting observation to note as well is that the month of the year matters into the decision making, suggesting that holidays and climate could be a potentially contributing factor.

The three least contributing factors were the year of the appointment, the type of appointment, and whether it was the dental clinic who booked the patient's appointment or not. These factors were more important in other models, especially the "tobecalled" attribute. The open random forest tree could not find a relationship in this data like the other models did.

6.2.2 GDPR's Effect on the Results

As the data had been carefully tokenized for each input label in a fashion that would be as non-destructive as possible for ML patterns to work with, the risk of having reduced learning capabilities was disregarded. There is however a possibility that non-tokenized data would provide a slightly better result as most of the factors which had been tokenized were part of the least significant contributing factors in the learning process.

6.2.3 Link Strength between Data and Problem

It is interesting that with only general data, the model found relationships and had a prediction score of almost 70%. To increase this score prediction more specific data needs to be included which might not even increase the score by much. It is hard to create a model that predicts the future with each individual having a unique life.

Data that could be more informative for the model is more personal information such as if the patient has a drivers license or a real time update if the subway is running or not. How long it takes the individual to get to its destination, a real time update on

the health of the patient etc. All these attributes might help the model make a more accurate prediction or it might not, it's impossible to know.

6.3 Analysis of Company Benefits

All the different models and their variation presented in the result sections find different relationships in the data. All models have superior qualities in their own aspect, else they would not be relevant. Different models are used for different dataset types, for our dataset the best prediction model was the open random forest tree variation, which is costly in most aspects but gives us the best prediction. 70% is far better than simple guessing (50%) which can still be relevant for the company in order to boost revenue.

The question remains however whether or not the company would be able to increase their booking capacity by overbooking time-slots where the show-up likelihood is considerably lower than usual. A precise extent in which the model can provide an optimization to the company is thus not able to be answered in this report.

6.4 Range of Models

Creating a model which can make a prediction in the future with close to 100% accuracy without knowing personal habits is challenging because of the nature of humans. Everyone is unique in their own way. Combine that with our limited number of ML models that can find other relationships in the dataset. Models such as image classification, for these models there are always improvements that could be made but the prediction into the future factor does not exist for it, that factor alone can disturb the prediction.

7 Conclusions and Future Work

7.1 Conclusions

In conclusion, with the help of AI and ML there is a way to create a model in order to have a more stable flow of patients. With the data provided to us from the company Distriktstandvården, the best model created was an open random forest which has a prediction accuracy of 69.3%.

Subaim #1 was achieved where an overview of beneficial NN models was discovered. Subaim #2 was also achieved, where a functional prototype model was developed. Subaim #3 was achieved but not completely satisfied as the 75% prediction quality target was not met. The prediction quality was however assessed to be close enough to the target quality that it was judged good enough to be useful for the time being.

7.2 Limitations

One major limitation with this project was the availability and access to ML models. We assessed the construction of a model from scratch to be unrealistic due to the vast amounts of work that would have been required to design one from scratch. Therefore we ran into limitations belonging to the libraries we used as well as limitations in adjustability in the designs of the models. There was computation power available at the company, but it was not easily available to us which resulted in work being performed on laptops. This resulted in a radical decrease in development speed. A workaround solution to speed up computations was to run the algorithms using video graphics cards. However, the libraries available to us that could utilize the graphics cards did not include some of the models which were most relevant for our use case. We therefore ended up with a compromise solution where an open and optimized library called Scikit-learn Library was used, that could be easily set up and run from our laptops. The advantage of using a library was also the fast switchability and trialability of models due to the cohesive system, with the disadvantage that there was a limit to how many models that were available to use, and also how much that could be modified in the model.

We encountered a limitation with the diversity of the data. Despite us having access to a million reported visits, the factors used and available to us to take into consideration when drawing predictions was limited in numbers and relevance. We decided to remove only parameters that would not contribute to the prediction and kept the rest so that the NN could decide on which factors that were the most significant. We could not create factors which did not exist in the provided data, and so had to accept what was available.

7.3 Future Works

Continued research would be needed when it comes to what factors that can contribute more when drawing predictions. We expected the quantity of our factors to be enough to draw some conclusion as well as the factors having a good enough quality to marginally contribute toward the prediction's certainty. However, as this report's conclusion suggests, the quality of the factors was lacking, resulting in a low prediction rate, despite a high quantity of factors and data entries.

We don't expect an even larger database to contribute much to the prediction's quality as an exponential increase by 10 of the data entries did not provide much difference, suggesting a rapidly declining contribution as it scales from 100 000 entries in our specific case. Therefore, a focus on how to increase the quality of the data is expected to contribute more to the goal and application of this type of prediction.

To improve the data quality, further literature studies could provide more suggestions on what factors to take into consideration. Psychological and local factors could also be investigated such as means of travel, personal experience from visits. Feedback from patients could be investigated if it can be collected as a possible contributing factor.

Other models or custom made models could also be a factor which could provide different prediction qualities as the library that was used was limited in how much could be modified. Creating a custom library from scratch or using alternative libraries with more advanced or more advantageous designs for the specific case could prove to benefit the prediction quality even further, and is thus fields of further research.

7.4 Reflections

7.4.1 Reflection on Project Initiation

Before the paper began there was some uncertainty in how much work would be needed to perform the research. Most time was spent performing the literature study as the tools and libraries that were available were known to us from previous experience.

The actual development of the model and code used for testing was surprisingly rapid, taking about a week worth of full-time development. Meanwhile, the testing of the models proved to take a lot longer when attempting to discover the model and set-up that was the most optimal for providing quality predictions.

7.4.2 Parameter Calibration

A margin of error was present when balancing the training data as it was manually calibrated to provide a balanced confusion matrix for the random forest network model. This was done to ensure a minimal amount of guesses would be performed by the model so

that the prediction would be logically drawn rather than looking at the frequency between the results and having it guess based on the most frequent answer when uncertain.

An automated calibration test would have given the balancing of the data a slightly more accurate balancing ratio as the ratio would not have to be manually adjusted in exchange for a possibly larger amount of needed trials before discovering the ratio which provides the highest amount of balance in the confusion matrix.

7.4.3 Project Scope

The scope of the project was slightly overambitious when looking at the amount of uncertainties and risks. There was a lot of technology and terminology which we did not have much experience with and that we had to read up on as it was encountered. There was also an uncertainty in how much time had to be spent to perform the project as well as uncertainties in how much time we could afford to put aside to work on it. The project could therefore have been judged to be a complicated one for us to successfully execute.

Despite the uncertainties and risks however, we managed to still find ourselves progressing according to plan where we first started with the easiest alternative, the linear regression model, before continuing to the more advanced but more fitting model for our data and goal. The linear regression model enabled us to design a foundation where other models could quickly be integrated into and tested. This meant that we could easily switch over to a random forest model, the model which according to our literature studies was recommended for our specific case. We also seized the opportunity of being able to test a few other models and so tested a few other alternatives to see how they would compare to the random forest prediction quality.

7.4.4 Link to the Bigger Picture

Despite successfully creating a model which could predict with a decent certainty, the prediction proved to be too low to be of much use for the company this was developed for. The promise to the company could thus not be entirely satisfied with this project, yet it enabled future work to be based on the results provided in this report.

From a research and academic standpoint, this report presents and provides the results of applying ML models into a use case where there is not much predictability despite a large quantity of data and factors to consider. This paper can therefore provide insight in areas and situations where ML might not be so easily implemented as a solution. Another benefit this paper provides is insight into how a project with these specifications might perform and progress as the work needed to obtain the results and resulting model have been done already and assessed in this paper.

When looking at the bigger picture, the research performed in this paper is not very significant to the entire research field of ML since it merely applies existing science into a

popular application field where ML is being investigated and applied frequently. For us as students learning and testing this kind of technology however, it provides a big impact to our understanding of applied ML as well as great introductory experience to be able to contribute more in future studies. This paper therefore has a more significant impact on the authors and may provide benefits to the research field in the long-run as we have gained confidence and motivation in performing more research in this field in the future. There is also a chance that this paper can inspire more students to do research in this field of study, introducing more research and providing more basic application-experience into the field of research.

References

[1] Haocheng Tan. *A brief history and technical review of the expertsystem research.* 2017. URL: https://iopscience.iop.org/article/10.1088/1757-899X/242/1/012111/pdf (visited on 05/25/2021).

[2] *Prediction.* URL: https://www.datarobot.com/wiki/prediction/ (visited on 05/25/2021).

[3] *Why airlines overbook flights and what bumped passengers can do about it.* 2017. URL: https://www.cbc.ca/news/business/airline-passengers-bumping-overbooking-united-1.4065603 (visited on 05/25/2021).

[4] IBM Cloud Education. *What is Artificial Intelligence (AI)?* 2021. URL: https://www.ibm.com/cloud/learn/what-is-artificial-intelligence (visited on 05/25/2021).

[5] James Chen. *Neural Network.* 2020. URL: https://www.investopedia.com/terms/n/neuralnetwork.asp (visited on 05/25/2021).

[6] Tine Wiederer. *Neural Networks in Javascript.* 2016. URL: https://webkid.io/blog/neural-networks-in-javascript/ (visited on 05/25/2021).

[7] Rai Akanksha. *Activation functions in Neural Networks.* 2020. URL: https://www.geeksforgeeks.org/activation-functions-neural-networks (visited on 05/25/2021).

[8] IBM Cloud Education. *What is machine learning?* 2021. URL: https://www.ibm.com/cloud/learn/machine-learning (visited on 05/25/2021).

[9] *History of Artificial Intelligence.* 2021. DOI: online. URL: https://www.coe.int/en/web/artificial-intelligence/history-of-ai (visited on 05/25/2021).

[10] Sewaqu. *Neural Networks in Javascript.* 2010. URL: https://en.wikipedia.org/wiki/Linear_regression (visited on 05/25/2021).

[11] Jason Brownlee. *Linear Regression for Machine Learning.* 2020. URL: https://machinelearningmastery.com/linear-regression-for-machine-learning/ (visited on 05/25/2021).

[12] Moshe Binieli. *Machine learning: an introduction to mean squared error and regression lines.* 2018. URL: https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/ (visited on 05/25/2021).

[13] Michael Nielsen. *Neural Networks in Javascript.* 2019. URL: http://neuralnetworksanddeeple
com/chap1.html (visited on 05/25/2021).

[14] Michael Nielsen. *Using neural nets to recognize handwritten digits*. 2019. URL: http://neuralnetworksanddeeplearning.com/chap1.html (visited on 05/25/2021).

[15] Vineet Joshi. *Activation Functions*. 2019. URL: https://www.geeksforgeeks.org/activation-functions/ (visited on 05/25/2021).

[16] *Hidden Layer*. URL: https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning (visited on 05/25/2021).

[17] et al Mao WB. *Application of artificial neural networks in detection and diagnosis of gastrointestinal and liver tumors*. 2020. URL: https://www.securityinfowatch.com/video-surveillance/video-analytics/article/21069937/deep-learning-to-the-rescue (visited on 05/25/2021).

[18] IBM Cloud Education. *Neural Networks*. 2020. URL: https://www.ibm.com/cloud/learn/neural-networks (visited on 05/25/2021).

[19] Eda Kavlakoglu. *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?* 2020. URL: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks (visited on 05/25/2021).

[20] Julianna Delua. *Supervised vs. Unsupervised Learning: What's the Difference?* 2021. URL: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning (visited on 05/25/2021).

[21] Paul Blondel. *Application of artificial neural networks in detection and diagnosis of gastrointestinal and liver tumors*. 2017. URL: https://cai.tools.sap/blog/machine-learning-algorithms/ (visited on 05/25/2021).

[22] Samer Ellahham* and Nour Ellahham. *Use of Artificial Intelligence for Improving Patient Flow and Healthcare Delivery*. URL: https://www.hilarispublisher.com/open-access/use-of-artificial-intelligence-for-improving-patient-flow-and-healthcare-delivery.pdf (visited on 05/25/2021).

[23] Jyoti R. et al. *Real-Time Capacity Management and Patient Flow Optimization in Hospitals Using AI Methods*. 2020. URL: https://link.springer.com/chapter/10.1007/978-3-030-45240-7_3 (visited on 05/25/2021).

[24] et al Sarab AlMuhaideb. *Prediction of hospital no-show appointments through artificial intelligence algorithms*. 2019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6894458/ (visited on 05/25/2021).

[25] Intersoft Consulting. *Recital 26 Not Applicable to Anonymous Data**. URL: https://gdpr-info.eu/recitals/no-26/ (visited on 07/03/2021).

[26] Mustafa Ali and Michael Arenander by plotting data using the matlib python library. 2021.