Introduction to Machine Learning for ophthalmologists

Reference:
Consejo Alejandra, Melcer Tomasz, Rozema Jos.- Introduction to Machine Learning for ophthalmologists
Seminars in ophthalmology - ISSN 0882-0538 - Philadelphia, Taylor & francis inc, 34:1(2019), p. 19-41
Full text (Publisher's DOI): https://doi.org/10.1080/08820538.2018.1551496
To cite this reference: https://hdl.handle.net/10067/1564670151162165141

Introduction to Machine Learning for Ophthalmologists

Alejandra Consejo, MSc PhD,[1,2,3] Tomasz Melcer, MSc,[3] and Jos J. Rozema, MSc PhD[1,2]

[1] *Department of Ophthalmology, Antwerp University Hospital, Edegem, Belgium*
[2] *Department of Medicine and Health Sciences, University of Antwerp, Antwerp, Belgium*
[3] *Department of Biomedical Engineering, Wroclaw University of Science and Technology, Wroclaw, Poland*

Abstract

New diagnostic and imaging techniques generate such an incredible amount of data that it is often a challenge to extract all information that could be possibly useful in clinical practice. Machine Learning techniques emerged as an objective tool to assist practitioners to diagnose certain conditions and take clinical decisions. In particular, Machine Learning techniques have repeatedly shown their usefulness for ophthalmologists. The possible applications of this technology go much further than been used as diagnostic tool, as it may also be used to grade the severity of a pathology, perform early disease detection, or predict the evolution of a condition. This work reviews not only the latest achievements of Machine Learning in ocular sciences, but also aims to be a comprehensive and concise overview of all steps of the process, with clear and easy explanation for each technical term, focusing on the basic knowledge required to understand Machine Learning.

KEYWORDS: Automated diagnosis; Artificial Intelligence; Clinical research; Vision sciences

DISCLOSURE

The authors report no conflicts of interest and have no proprietary interest in any of the materials mentioned in this article.

ACKNOWLEDGEMENTS

This work was supported by a grant from the Flemish Fund for Scientific Research (TBM - T000416N).

1. Introduction

Clinical practitioners nowadays have an incredible amount of data at their disposal, generated by new diagnostic and imaging techniques. This presents them with the very real challenge of properly taking advantage of all this new information, not only in the interpretation of each measurement separately, but also of the substantial information that lies in the relationships between different measurements. Biological data is usually highly correlated which hampers the interpretation of the results. In order to assist physicians with this evermore daunting task Machine Learning techniques have been proposed and successfully applied to different branches of medicine as a tool to help diagnose diseases in which routine diagnostic information is not always clear. Besides supporting the clinician when the diagnosis is not clear, there are many other justifications for using Machine Learning, including identification of referable disease in screening and primary care settings. Although these techniques are not yet widespread in clinical practice, there have been many very promising reports on how Machine Learning could benefit clinicians in the near future.

This work aims to explain the basic ideas behind Machine Learning and highlight its potential for clinical and research practices in ophthalmology and optometry. For this purpose, a practical overview is given on how the reader may start using Machine Learning, with many examples of the current uses in ophthalmology as inspiration. The first section gives a brief overview of Data Science, Machine Learning and its various subfields, with an emphasis on supervised Machine Learning and a basic glossary. In section 2 the importance of initial data preparation (or data pre-processing) is discussed, explaining outlier detection, missing values and variable selection techniques. Next, section 3 presents a guide for the most common supervised classifiers, after which assessing the correctness of the models used, interpretation and reporting are discussed in section 4. Section 5 reviews previous applications of Machine Learning in Ophthalmology and Optometry, followed by practical hints for using Machine Learning in section 6. Finally, section 7 gives a short overview and concluding remarks.

1.1. Commonly used terms

Due to the multidisciplinary character of the field Data Science uses many concepts and terms that may be unfamiliar to the average reader. For this reason it is important to introduce a glossary with the most commonly used terms in Machine Learning (Table 1).

The entire pool of data is known as a *dataset*, which is composed of *instances* (e.g. subjects or eyes). Each instance is described by a collection of *variables*, features that contain information on certain aspects of e.g. an eye. These variables may be *numerical* (a continuous variable, e.g. axial length), *binary* (a variable with only 2 options, e.g. subject smokes or does not), *categorical* (a variable with a limited number of options, e.g. eye colour), etc. Instances may be arranged into *classes* (groups, e.g. healthy subjects and keratoconus patients). Meanwhile, the algorithms used by the computer to “learn” are known as *classifiers* (section 3). These classifiers produce a *model*, a mathematical formula or algorithm that predicts a class using knowledge from the training procedure. Datasets are most commonly represented in the form of a table, with rows representing instances (eyes), and columns representing variables. In this form, each instance has the same set of variables, and each variable is of a binary, categorical, ordinal or numeric type. This form is also often called *structured data*, as opposed to *unstructured data*: images (e.g. retinal images [Graham, 2015]), descriptions in natural language (e.g. clinical notes [Zheng et al., 2014]), biological signal recordings (e.g. corneal pulse [Danielewska et al., 2014]), a time series (e.g. longitudinal clinical data on disease progression [Futoma et al., 2016]), etc.

Table 1. Basic glossary of terms in Machine Learning

Term (Synonyms)	Definition	Practical Interpretation or example
Instance (example, case, record)	A single object from which a model will be derived; examples in a training set	Each subject or each eye in a particular dataset
Dataset (data set)	Group of instances	All data together
Variable (field, feature, attribute)	A quantity describing an instance	Parameter to analyse (e.g., age, corneal radius, intraocular pressure,…)
Binary variable	A variable that only admits two possible values (often “yes”/”no”)	e.g. smoker, prior ocular surgery, use of medication, etc.
Categorical variable (nominal variable)	A variable that permits multiple values, but with no inherent order or ranking sequence	e.g. ethnicity, gender, colour of the iris etc.
Discrete variable	A variable that can only take certain number of values.	e.g. number of visits (it could be any value between 0 and infinity, but it could not be for example 2.5)
Numerical variable (continuous variable)	A variable described by a number	e.g. corneal radius, intraocular pressure etc.
Ordinal variable	A variable that admits several values, where there is a natural ordering for the values.	e.g. no diabetic retinopathy, mild, moderate, severe, proliferative.
Regular variable (regular feature, independent variable)	A variable that is used as source of information for predictions (as opposed to a class feature).	e.g. diagnostic measurements.
Class (target, group, dependent variable)	A binary, categorical or ordinal variable which we aim to predict using a model (as opposed to a regular variable)	Final result: e.g. healthy or not healthy
Classifier	Algorithm or group of algorithms used by the computer to learn and classify data	Neural networks, decision trees or Support Vector Machine etc. are examples of classifiers
Model	A mathematical formula or procedure produced during the training process	e.g. a linear equation, a single decision tree, an instance of a neural network
Multiclass/ multi-classification problem	Situation when data needs to be classified in more than two categories (classes)	Final result with more than two classes: e.g. mild, moderate and severe keratoconus

1.2. Data Science and Machine Learning

Before exploring Machine Learning, it is important to point out that there are many related ideas that go by familiar names such as Artificial Intelligence (AI), Neural Networks (NN) and Data Mining (Figure 1a). All these concepts are part of Data Science, a broad interdisciplinary field, encompassing mathematics, statistics, computer science, information technology and data visualization. Data Science studies the process of collecting data, extracting relevant information, presenting new insights, and taking automated data-based decisions [Dhar, 2013]. The term itself often acts as an umbrella for all activities related to automated data processing. We would also like to note here that many sub-disciplines of Data Science are defined by the goals they intend to accomplish, rather than by their actual research topics. Consequently often the same means are used to solve different problems, leading to a significant overlap between subfields.

Artificial Intelligence, a field of computer science so recognizable in mass culture that a well-received Hollywood movie used the term as its title, investigates how to create intelligent agents [Russell & Norvig, 2003]. Intelligent agents are computer programs that are able to operate without direct and precise instructions from its users, thus giving computers basic problem-solving abilities, as seen in humans. This is especially useful for problems such as automatic adaptation of agents to a changing environment, dealing with vaguely defined goals, optimizing the use of resources, and automatic knowledge inference from complex data. This last process, known as Machine Learning, is most important for the context of this work, gives computers the ability to learn without being explicitly programmed [Arthur Samuel, 1959] by constructing and utilising algorithms that can learn and make predictions on data [Kohavi, 1998]. Machine Learning also overlaps with Data Mining, which studies semi-automatic knowledge extraction, including exploratory data analysis and classical statistics. Another subfield of Data Science that has recently became popular due to rapid growth of dataset sizes, is Big Data analytics. This field focuses on analysis of datasets so large that standard methods can no longer work correctly due to constraints in computer speed or storage. Neural Networks (NN), recently also referred to as Deep Learning, are algorithms often used in supervised tasks and are designed to mimic human neuron cells (see section 3.2). In the last years the term Deep learning gained interest since it was shown that for extraordinary amounts of data it outperforms traditional learning algorithms. Deep learning-based models are bigger, more complex and require more computation resources.

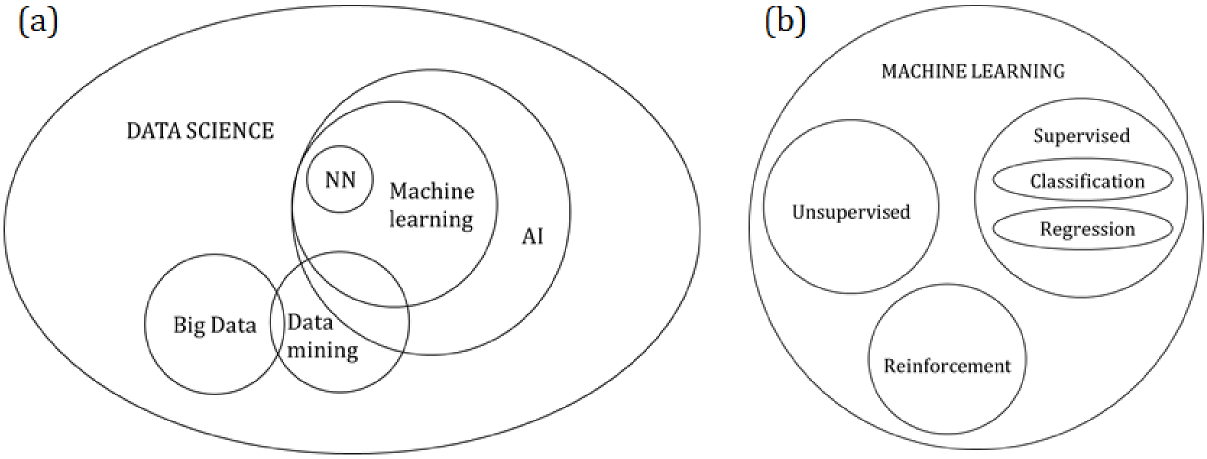

FIGURE 1. (a) Overview of the most important fields of Data Science. (b) Overview of the most important learning techniques used in Machine Learning. NN: Neural Networks; AI: Artificial Intelligence.

Machine Learning may be divided into several major branches, among which supervised learning, unsupervised learning and reinforcement learning are the most important ones (Figure 1b). Supervised learning devises relations between a specific variable (e.g. presence of glaucoma) and other variable (e.g. diagnostic measurements, age, ethnicity, etc.) to make predictions for the specific variable (e.g. "how likely is glaucoma in this 60-year-old person of Japanese ancestry with intraocular pressure of 22 mmHg and a heart disease?"). Supervised Learning problems may be divided further based on the nature of the variable parameter being considered: if the predicted variable is numerical (e.g. a continuous parameter, such as corneal radius) the problem is called a *regression problem*, if the predicted variable is categorical (e.g. healthy/not healthy) it is called a *classification problem*, and if the predicted variable has more than two possible predictions (e.g. primary open angle glaucoma/narrow-angle glaucoma/secondary glaucoma) it is called a *multi-class classification problem*. Supervised learning is usually considered to be easier to perform, interpret and validate than the other branches of Machine Learning.

Unsupervised learning, on the other hand, focuses on the discovery of previously unknown, but useful correlations that span the available set of variables [Jain et al., 1999]. Example questions that can be modelled by these methods are whether there are any qualities that divide a population into natural groups (clustering analysis) or finding the most atypical cases in a cohort (anomaly detection). These goals are less specific in nature, making these methods especially suitable for Exploratory Data Analysis, a Data Mining method. The fact that one of the main objectives of unsupervised learning is to identify previously unknown patterns makes it particularly useful for ophthalmology when investigating patterns of glaucomatous visual field defects and other retinal diseases. The results of Unsupervised Learning are considered to be more difficult to use when solving a specific problem. Finally, Reinforcement Learning is a decision process based on a reward system, which is useful to model interactions between intelligent agents. However, these methods are rarely used in biomedical applications.

As most applications in Ophthalmology and Optometry aim to determine which group a particular instance belongs to (e.g. a certain pathology is present or not), this work will focus in the following sections only on classification tasks by means of Supervised Learning.

It is also worth mentioning that Machine Learning techniques can offer various analyses that classical statistics cannot. The line between classical statistics and Machine Learning is unclear and far from being well defined. Some researchers consider them completely independent from one other, while others consider that some Machine Learning techniques are complex statistics-based systems. Generally, Machine Learning models are considered dynamic as opposed to statistical models' static nature. Both statistics and Machine Learning techniques aim to find correlations between variables, but they differ in the sense that Machine Learning is able to learn from data without relying on rules-based programming, while statistics modelling is able to successfully analyse data, but not to learn from it.

1.3. The process of Supervised Learning

Supervised Learning may be described as training a mathematical model with a historical dataset by giving it all variables it is supposed to use, along with the correct classifications, to subsequently use that model to predict the classes for new data, previously unknown to the model (and possibly the user). The training process consists of two major steps: model construction and performance assessment (Figure 2).

Model construction consists of finding a mathematical formula that describes how to assign cases to classes as accurately as possible based on historical data. The formula itself is often in form of a complex "black box" procedure with little explanation why should it work. As there are no guarantees that the formula will accurately predict classes, it is necessary to explicitly test its reliability. To use an analogy, the process is similar to teaching a student in a tutoring session and then using a written assessment to evaluate whether the student mastered the topic. Most models also provide a "score" of how strongly it is considered that a particular class is correct for any given case, which for binary classification problems may take the form of a probability (a value between 0 and 1). In other cases the score may not be directly interpretable, however, except for interpreting that a lower value represents more confidence in one of the classes, and a higher value in the other class. In the school analogy, the test is an example of a binary classification problem that aims to state whether or not a student has mastered the material covered during the course. Meanwhile, the score received by the student reflects the likelihood with which he truly understands the matter, with a high score indicating a high likelihood, and a low score a low likelihood. By selecting a threshold value for this score the user can decide the trade-off between false positives and false negatives that best represents the practical costs associated with different types of misclassifications (see section 4.1). This is related to assessment scores in the classroom analogy by which the teacher declares what is a passing score. Different cut-offs will result either in passing some students that did not master the material, or not passing some students who learned most of the required content, but were unlucky to get more difficult exercises.

As there is no guarantee that a model will be accurate, it is essential to explicitly test its reliability. The assessment is performed by asking the model to predict classes for data where classes are known to the user, and subsequently comparing the model's predictions to the actual classes (section 4). It is crucial that this assessment is performed on data that was not used for the model construction to avoid a misleading result. Continuing the classroom analogy, the problem can be exemplified by a student that chooses to just memorize the given example problems, rather than trying to understand them. In that case the student would ace an assessment based on problems solved during the tutoring session, but struggle with a new problem. For this reason it is common to divide all available historical data into *training* and *validation* sets: use only the training set for the model construction and validate using only the second set. Other, more complex validation schemes can be found in section 4.

During model construction there are many choices to be made regarding the training process, such as what variables (section 2.3) and which classification algorithm (section 3) to use for the training process. Furthermore, there are algorithm-specific details to choose, e.g. the depth of a decision tree (section 3.1) or the number of neighbours for the k-Nearest Neighbours (section 3.4). These choices are commonly referred to as *meta-parameters*, as opposed to the *parameters* used in the mathematical formula describing the model. These choices may influence the prediction accuracy, reliability of the training procedure, interpretability of the model, computation time, and memory usage. It is usually not possible to provide a priori selection of "best" meta-parameters that are suitable for one specific problem. In a typical application of Supervised Learning a user must therefore try many choices of meta-parameters and use a validation method (see section 4.3) to select the best choice. To use the student analogy again, some topics may be better taught using a whiteboard, or a pre-recorded video, or in a laboratory by performing an experiment; the choice of metaphors, textbooks or problem sets will also affect the teaching process.

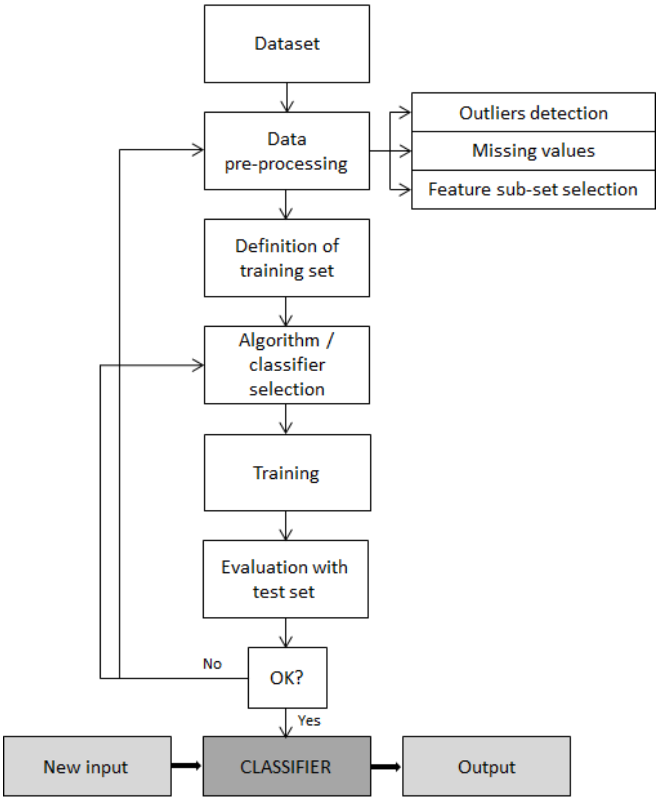

FIGURE 2. Flow chart depicting the general process of Supervised Learning. Upwards arrows indicate the most immediate approaches to correct classifier performance. The steps in the process may be modified to suit the needs of the particular problem, however.

2. Data pre-processing

Usually data needs to be processed before applying Machine Learning. One of the reasons is that various classification algorithms often have very specific requirements or assumptions regarding the data. Typical requirements are being in the form of a table, having only numerical variables, no missing data values, no outliers, etc. Another common reason for pre-processing is the size of the dataset. While generally bigger datasets are better and enable more complex methods that result in better predictions, it is sometimes impractical to collect more data (e.g. due to additional expenses or time required). Instead, practitioners apply pre-processing techniques that reduce the complexity of a dataset, making the prediction task easier to learn. Data pre-processing may also help reduce data quality issues and improve the results since the more concrete and accurate the training data is provided, the better and more reliable the results will be [Zhang et al., 2003]. On the other hand, data pre-processing can also be used to augment datasets to mimic real-world clinical applications by reducing the quality of images, rotating orientation etc. This section reviews the fundamental steps of data pre-processing, such as outlier detection, dealing with missing values and dimensionality reduction techniques.

2.1. Outliers

Outliers are extreme values that deviate considerably from other observations. There are essentially two types of outliers. First there are data points that do not make sense from the point of the physical or biological process being modelled (e.g. bad measurement, transcription error, etc.). Studying these points may help verify the methodology of the data collection process. The other type are outliers that carry information about a possible, but very unlikely outcome. Inspecting these points might bring new knowledge and better understanding of the phenomenon being predicted. As a rule of thumb, it is fine to discard or correct the first kind (if possible without introducing errors), while it is not always proper to remove data points from the latter kind.

Automated methods for outlier detection are an active research topic [Hodge et al., 2004] [Schiff et al., 2017], and include methods such as anomaly detection, novelty detection or exception mining. If the probability distribution for a variable is known a priori (e.g. a normal distribution) identifying outliers can be done by observing the likelihood of data under that distribution. Another very simple approach is to consider any observation whose variable value fall outside three standard deviations from the variable mean value as an outlier. Other, more sophisticated methods are based on proximity and projection methods. The choice of an outlier detection method is strongly dependent on the choice of classification algorithms and dataset size. Some methods reliably deal with outliers without additional processing (e.g. decision trees), while others are far more sensitive to outliers (e.g. Ordinary Least Squares regression).

2.2. Missing values

Regardless of how carefully a database was made, some values might be missing, typically due to procedural errors (e.g. forgetfulness, faulty equipment etc.) or circumstances outside the researcher's control. Like before with the outliers, the missingness of data points may be a random occurrence that does not bring any new information to the classification process. Meanwhile, there may be circumstances in which the missingness is important information by itself that is worth preserving for classification purposes (e.g. the impossibility to collect that piece of data may be related to the studied phenomenon). For values that are not missing randomly, it may sometimes be useful to introduce an additional binary variable denoting whether the value was missing.

The impact of missing values depends heavily on the algorithm used. While some algorithms tolerate missing data without pre-processing (e.g. some methods based on decision trees [Chen et al., 2016]), there are others that require all values to be available for all variables. Hence several treatments have been proposed to amend this issue [Little et al., 1987]. One simple method is to *discard* cases with missing data, which is statistically correct if the values are missing randomly. It is also common to remove variables with missing data. However, this may be problematic if the variable is considered useful for prediction. Next, one may perform *parameter estimation*, a statistical method to estimate the variable values that maximize the likelihood of the instances given the non-missing variables [Dempster et al., 1977]. Another option is to use *imputation* techniques, which fill in the missing values with plausible values, such as the mean value of a variable or a value taken from another, randomly selected case [Donders, 2006]. Finally, there are other, more complex variations of these methods, such as 'hot deck and cold deck' or 'prediction models' [Batista et al., 2003].

2.3. Feature Selection and Dimensionality Reduction

Feature Selection and Dimensionality Reduction methods aim to reduce the number of variables in a dataset. While usually having a large number of variables provides more discriminating power, having too many variables in combination with a limited dataset size and an overly flexible algorithm could lead to an overfitted model that incorporates accidental, spurious relationships found in the training dataset, but do not exist in the general population. As such, overfitted models tend to have a low prediction accuracy for new data. To illustrate this with the student analogy, imagine that a student is requested to look at random ink blot patterns and report what he sees in them, like in the Rorschach test. The student will quickly form some mental image based on a combination the ink blot's geometry (the training data) and his earlier preconceptions (the parameters of the classification model). While the ink blot is randomly shaped, the student has interpolated the available data into a familiar image that is not really there.

Overfitting may be dealt with by reducing the dimensionality of the data, either by selecting a subset of variables (*Feature Selection*) or computing a new, more compressed set of variables. This has the additional benefit that it reduces computation times as the speed of classification algorithms often depend on the size of the dataset [Yu et al., 2004]. Common feature removal methods include:

Missing values ratio. Variables with too many missing values are less likely to carry important information. Hence, variables with number of missing values greater than a given threshold may be removed, keeping in mind that relevant variables should be kept, even with high levels of missing data.

Low variance filter. Similar to the previous approach, variables with only minor variations in the data are less likely to carry substantial information. Numerical variables with a standard deviation below a given threshold, or a categorical variable predominantly set to a single value (e.g. mostly males subjects), are good candidates for removal.

High correlation filter. Strongly correlated variables tend to carry similar information, meaning not much information is lost when some of these are removed. The correlation between individual pairs of nominal variables may be done using the Pearson's Product Moment Coefficient, while the Pearson's chi square value may be used to assess all pairs of nominal variables at once.

Factor analysis groups highly correlated variables. This may either be done by Exploratory Factor Analysis (EFA), which primarily aims to identify the underlying correlations between variables, and Conformation Factor Analysis (CFA), which tests whether the data fits a hypothesized measurement model [Harman, 1960].

Decision Trees Ensembles. This is a family of algorithms, such as random forests or gradient boosting, that are useful for variables selection in addition to being effective classifiers (section 3). One approach to dimensionality reduction is to generate a large and carefully constructed set of trees for a certain target variable and assess each variable being used to find the most informative subset of variables. If a variable is often selected as best split, it is most likely an informative variable to retain.

L^1 normalization, also called L1-norm or lasso [Tibshirani, 1996], is a regression analysis that performs both variable selection and *regularization* (i.e. the process of introducing additional constraints for the model to prevent overfitting). This enhances interpretability of the model. The additional constraints penalize having non-zero parameters in the linear model, leading the classifier to prefer models that only use a small subset of variables.

Forward Sequential Feature selection (FSFS) works by starting from an empty set of variables, and then repeatedly testing if adding variables one by one improves predictive accuracy in a significant way [Jain et al., 1997]. Each test is essentially training a new model on a different subset of variables. As a result the process is easy to understand [Guyon et al., 2003], but slow for highly dimensional data sets. Hence, they are usually applied after using other dimensionality reduction methods.

Backward Feature Elimination (BFE) starts with all variables in the dataset and sequentially removes one input variable at a time. After each removal, a new model is trained and its predictive accuracy is tested. The process stops when accuracy drops below an a priori chosen threshold [Koller et al., 1996]. Here too the algorithms tend to be slow for large datasets and should be applied only after using other data reduction methods.

A popular dimensionality reduction method that is not based on removing variables is Principal Component Analysis (PCA), which searches for the linear combinations of the original variables that better represent the dataset variance. This allows reducing the dataset while retaining information on the most typical ways the cases differ from each other [Jolliffe, 2005]. The method may only be applied to numerical variables and requires prior *data normalization* (i.e. bring all values within a certain interval, such as [0, 1]). In addition, PCA variables are linear combinations of the original values, so their clinical meaning might be very difficult to interpret.

As a conclusion of this section, we highlight a set of characteristics that according to Saeyes et. al (2007) can guide the choice for a technique suited to the goals and resources of practitioners. The clear advantage of filter techniques over others is that they are very efficient and fast to compute. However, they ignore feature dependencies which might lead to undesirably discard a variable that doesn't carry much information by itself but that could be useful in combination with others. Wrapper methods (such as FSFS and BFE) model variable dependencies and are less computationally expensive than randomized methods, however they are more prone than other techniques to over fit or get stuck in a local optimum (greedy search). Finally, embedded methods (such as Decision Trees or L^1 Lasso) generally show better computational complexity than wrapper methods and are also able to model feature dependencies.

3. Classifiers

A classifier is any algorithm that uses a training dataset to create a model of knowledge on some topic. There are many algorithms that can perform this operation. The choice depends on the dataset size, variable types, available computation resources, additional requirements put towards the classification process (such as model interpretability or modelling probabilities for classes), model capacity (how complex relations can be found by the training process) or inductive bias (a set of assumptions that lie behind choices made by the classifier during the training process, e.g. "similar cases should result in similar predictions", or "there are no relationships between pairs of regular variables").

While there are some guidelines that can help with choosing a classifier a priori (section 6), it is best to try several algorithms and compare the results using a validation technique (section 4) to find the model with the best predictive performance. This section aims to give an overview of the most popular classifiers and an intuitive idea of their performance.

3.1. Decision trees

A decision tree is essentially a series of directed questions that partitions data recursively to form groups or classes [Quinlan, 1986], much like the game of 20 questions. Each time a question is resolved, the model either goes along a path to another question or to a prediction if there are no more questions available. Decision trees are amply used for clinical applications. As an example, an oversimplified algorithm for keratoconus detection is shown in Table 2 and Figure 3. Decision trees are traditionally used supervised learning algorithm, but they can also be used for unsupervised learning, like clustering.

Table 2. Training set to detect keratoconus from healthy eyes

	K_m (D)	CCT (µm)	CT_m (µm)	ACD (mm)	Corneal scarring	AL (mm)	Class
I(1)	44	540	535	3.30	No	23.50	Normal (N)
I(2)	47	530	515	3.55	No	23.25	Keratoconus (K)
I(3)	48	480	470	3.60	Yes	24.50	Keratoconus (K)

I(i) = instance; K_m = mean keratometry; CCT = central corneal thickness; CT_m = minimum corneal thickness; ACD = anterior chamber depth; AL=axial length.

A major problem with decision trees is the stability of the training process: small changes to the training dataset may result in wildly different tree models. To illustrate this, consider a training dataset in which there are two choices for the initial question, both rated as very good by the classifier. In such case, a small perturbation to the initial training dataset (e.g. adding new cases) may push the algorithm to change that initial question, resulting in very different partitions of data used for choosing next questions. It is also observed that small changes to a training dataset may also considerably change the predictions—a characteristic considered troubling in many applications. This problem is often solved by assembling multiple tree models together, as will be discussed in the next section.

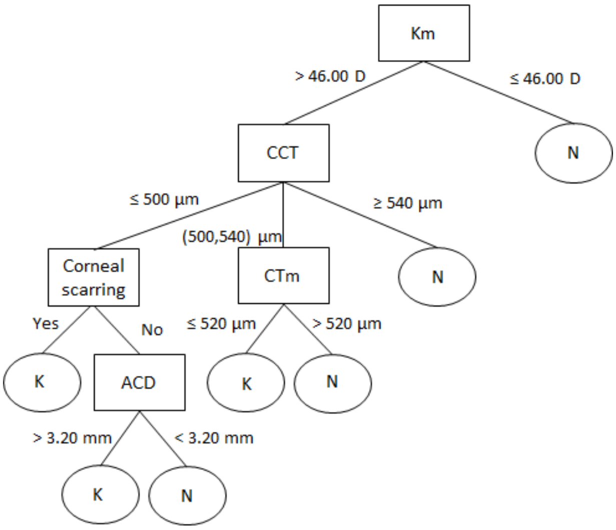

FIGURE 3. Oversimplified example of a decision tree (not to be used clinically) based on the training set from Table 2. Cut-off values taken from a previous work[27] Km = mean keratometry; CCT = central corneal thickness; CTm = minimum corneal thickness; ACD = anterior chamber depth.

Commonly used decision trees include ID3, C4.5, CART, C5.0, CHAID, QUEST, and CRUISE. Of these algorithms C4.5 is especially applicable and extensively used as it accepts both continuous and discrete values, can handle missing data and permits weighting different variables of a given dataset.

3.2. Tree ensembles

Ensembles are models that consist of many simpler models, whose predictions are aggregated to form a final answer. This may slightly increase prediction performance and can turn a collection of unstable models into a single stable model. Using a classroom analogy again, ensembles are like a group test in which the students can openly discuss their knowledge and ideas, thus leading to a better answer than when individual students should solve it on their own. While ensembles can be based on any type of classifiers, they are usually based on decision trees. Random Forest [Breiman, 2001], for example, is a classifier that uses many tree models, each trained independently on a different subsample of data and variables. Another example is Gradient Boosting models [Friedman, 2001], such as XGBoost [Chen et al., 2016], which are ensemble models consisting of a series of trees that are each trained to correct mistakes made by previous trees. In recent years, XGBoost is especially successful for structured datasets, as evidenced by many Machine Learning competitions [Chen et al., 2016]. Ensemble learning is a form of supervised learning, but not exclusively as some ensemble techniques are also used in unsupervised learning scenarios.

3.3. Artificial Neural Networks

As suggested by the name, Artificial Neural Networks were originally inspired by the communication between the brain's neurons. Neural Networks consist of a large group of cells called *neurons* that are organized in layers and interconnected like a flow chart in such a way that one may only follow the flow in a forward direction from the input layer towards the output layer, never once returning to a previous cell (i.e. a directed acyclic graph, Figure 4). Each neuron performs a very simple computation based on the information coming in from previous neurons (usually a weighted mean, followed by a non-linear, monotonic transformation), before sending the result on to the next neurons. After passing through several layers the network is able to classify the original input to the output it deems most appropriate. Over the last decades these ideas were improved further without following biological analogies.

First proposed in the 1950s, Artificial Neural Networks are one of the oldest classifiers in use [Haykin, 2004]. After an initial boom in the 1960s and new inventions boosting the field in 1980s [Olazaran et al., 1996], research into Artificial Neural Networks have entered a second renaissance thanks to the increased availability of computing power. Nowadays these general-purpose models outperform older, specifically designed methods for unstructured datasets (e.g. recognition of objects on images [Szegedy et al., 2015], speech recognition [Hinton et al., 2012] or language translation [Bahdanau et al., 2015]). Modern Neural Networks employ a large number of neurons and connections (e.g. a recently proposed architecture for natural language translation used 137 billion connections [Shazeer et al., 2017]). For this reason, these models are also called Deep Learning due to immense complexity of these models. Artificial Neural Networks can be both supervised or unsupervised.

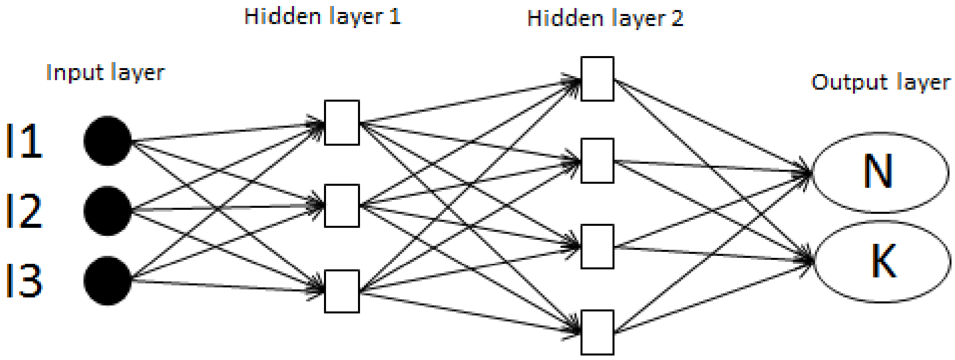

FIGURE 4. Basic four-layer neural network for the classification of normal eyes (N) and keratoconus (K). Squares represent neurons and lines correspond with the weighted output from previous neurons.

3.3.1 Self-Organizing Maps

A self-organizing map (SOM), also called self-feature map (SOFM), is a type of unsupervised ANN designed by Kohonen [Kohonen, 1990]. It is a data visualization technique that helps to understand high dimensional data by reducing the dimensions of data to a two-dimensional map. It also works as a clustering technique that groups similar data together. Therefore SOM reduces data dimensions and displays similarities among data.

3.4. Naïve Bayes

This is a statistical classifier based on Bayes' Theorem that makes the naïve assumption that each variable contributes independently to the probability of a case belonging to a particular class. In practice, it is very rare to fulfil this assumption, however. Regardless, the classifier may still achieve good results due to its simplicity and resilience to noisy, missing or irrelevant variables [Langley et al., 1994]. An example of Naïve Bayes (or "Idiot's Bayes" [Hand et al., 2001]) graphical model based on the keratoconus example of Table 2 is shown in Figure 5. The Naïve Bayes classifier usually falls in the category of supervised learning.

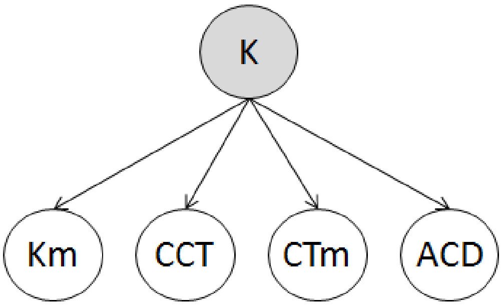

FIGURE 5. Naïve Bayes classifier as a probabilistic graphical model[39] with K representing the Keratoconus class node and the rest of circles representing the feature nodes (Km = mean keratometry; CCT = central corneal thickness; CTm = minimum corneal thickness; ACD = anterior chamber depth). Note that there are no connections between feature nodes, representing their independence from each other.

3.5. K-nearest neighbours (kNN)

This technique is based on the observation that instances (cases) of a given dataset generally reside near other instances with similar characteristics. Proximity in this context does not mean spatial proximity, but rather proximity in variable space. For each instance of the testing set we observe k (k=1, 3, 5...) examples from the training set which are the most similar, closest to that instance. The algorithm declares that the instance's class is the one that is most common among these neighbours.

This algorithm generates a very natural decision boundary (i.e. a borderline between areas of the variable space classified into different classes). The boundary is effectively a set of points (a line, a plane, etc.) whose distances to different classes is equal (figure 6). This property makes the algorithm results easy to visualize and interpret, as well as very flexible for low values of k. Low values of k turn out to also be prone to overfitting. For example, in the case of k = 1 the classifier essentially memorizes the training dataset, perfectly reproducing the classes of the training cases, thus rendering in-sample performance metrics useless. A proper validation procedure is therefore required to determine the model's efficiency.

Besides this risk of overfitting, kNN may also be slow in larger training samples as searching the nearest neighbours for each sample is often computationally intensive. Also, rescaling some of the variables may significantly alter the model and the predicted outcomes, and it is not easy to define distances for variables with discrete values. kNN is usually used as supervised learning method.

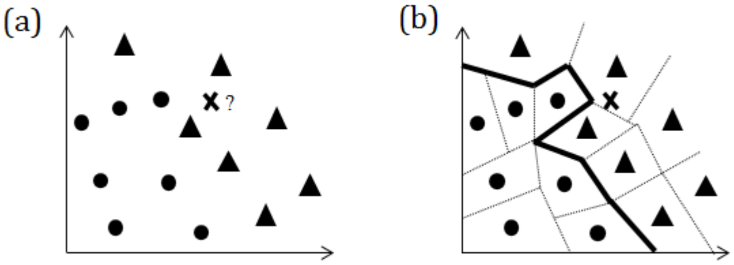

FIGURE 6. (a) Keratoconus eyes (represented by triangles) and healthy eyes (represented by circles) and an unknown sample (represented as a cross) that we aim to classify. (b) Partitions of the space into regions and decision boundary. The unknown sample would fall into the keratoconus eyes area. Thick line represents a decision boundary proposed by a kNN model for k = 1.

3.6. Support Vector Machine

Support Vector Machine (SVM) has recently become one of the most popular supervised learning classifiers due to its good outcomes when classifying difficult examples [Burges, 1998][Kotsiantis et al., 2007]. In its most general form SVM is a linear classifier [Cortes et al., 1995], which means that it seeks to find a hyperplane (e.g. a line for a dataset with two variables, a plane for a dataset with three variables, etc.) that perfectly separates data points belonging to different classes. Although there are many possible hyperplanes, the classifier selects the one that has the biggest margin that separates it from data points (Figure 7).

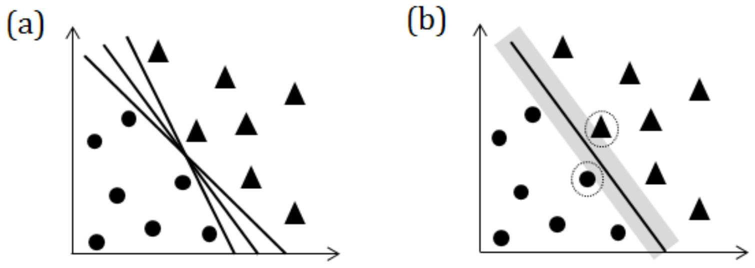

FIGURE 7. (a) Keratoconus eyes (represented by triangles) and healthy eyes (represented by circles) separated by different possible hyperplane (lines) that separate the classes. (b) Optimal linear classifier and margin (grey). The data points of each class that constrain the margin width are surrounded by a dashed line.

In general most datasets are not linearly separable, however, meaning that there is no hyperplane that perfectly separates the different classes (Figure 8a). To solve this problem SVM models can transform variables into *kernel space* by means of a *kernel trick* [Hofmann et al., 2008], a mathematical device that allows variable transformation allowing complex decision boundaries. An example of a polynomial transform is shown in Figure 8b: by adding a third dimension representing the distance of a data point to the centre of variable space, the dataset that was formerly not linearly separable may now easily be separated by a plane. In other words, the kernel function is to take data as input and transform it into the required form. Different SVM algorithms use different kernel functions. These functions can be different types. For example, linear, nonlinear, polynomial, radial basis function (RBF), or PUK, among others. The mathematical representation of some of these kernels is listed as follows:

The polynomial kernel of degree *d* is defined as

$$K\ (x, y) = (x \cdot y\ + k)^d \qquad (1)$$

where *k* is the constant. The kernel with *d* = 1 is the linear kernel function. Another very widely used kernel is the Gaussian radial basis function (RBF) kernel, often used when there is no prior knowledge about the data. It is defined as

$$K\ (x, y) = exp(-\|x - y\|/2\gamma)^d \qquad (2)$$

where γ > 0 is a parameter that controls the width of the Gaussian. A more sophisticated type of kernel function that can be used in SVM is the Pearson VII universal kernel (PUK). The PUK kernel function of multi-dimensional input space is given by the following formula:

$$K(x, y) = 1/[1 + (2\sqrt{\|x - y\|^2}\sqrt{2^{(1/\omega)}} - 1/\sigma)^2]^\omega \qquad (3)$$

where the parameters ω and σ control the half-width (also named Pearson width) and the tailing factor of the peak. The main reason to use the PUK kernel is its flexibility to change, by varying parameters ω and σ. The usefulness of the PUK kernel is that, by selecting the appropriate parameter setting, it might be used as a kind of universal kernel which could replace the set of commonly applied kernel functions, i.e the linear, polynomial and Gaussian kernels [Abakar, 2012].

By considering margins SVM is less sensitive to outliers than a common logistic regression. However, the computation time may be prohibitively long for large datasets.

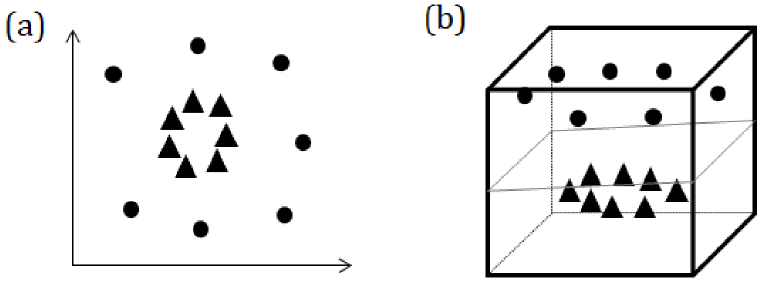

FIGURE 8. (a) Keratoconus eyes (represented by triangles) and healthy eyes (represented by circles) separated by a curved line that separates the classes. (b) Three-dimensional view of the problem using a kernel. The tilted plane that separates both classes is a plane that separates classes with the biggest margin.

3.7. K-mean

K-means clustering is one of the simplest unsupervised learning algorithms. It aims to find groups (clusters) in the data, with the number of groups represented by the variable *K*. The main idea is to define *K* centers, one for each cluster. Generally, different location causes different result. So, the better choice is to place them as much as possible far away from each other [Teknomo, 2006].

3.8 Principal Component Analysis

Principal Component Analysis (PCA) can be used as a plain statistical technique for data reduction, as it was mentioned in the previous section, but it can be also utilized in unsupervised learning. In unsupervised learning it is often used to find patterns in high-dimensional datasets or as an aid in clustering and segmentation models [Ding, 2014]. PCA introduces a lower-dimensional representation of the dataset. Further, PCA gives a new set of variables called 'principal components' which could be further used as inputs in a supervised learning model.

To conclude this section on Classifiers we would like to highlight that there is no a single answer to the question 'What machine learning classifier should I use?', it always depends. It depends on the size, quality and nature of the data, also on the available computational time and the expected output. For example, decision trees are often used by beginners because they are easy to interpret, but still they are more often used in compositions such as random forest showing generally a very good accuracy but longer training time than other classifiers. Artificial Neural Networks also show very good accuracy and they can be implemented in supervised and unsupervised learning problems. ANN can efficiently learn and model non-linear and complex relationships, which makes them very popular. However, this ability comes with a price, ANN training often requires great computational complexity. Contrarily, Naïve Bayes technique is faster and easier to understand and implement although its performances is sometimes compromised depending on the nature of the problem. Support Vector Machines usually show excellent accuracy, relatively fast training times and the use of linearity which make these techniques amply used nowadays. Regarding the purely unsupervised classifiers, K-mean is more primal but easier to understand than other algorithms and its training process is relatively fast. PCA, on the other hand, is a great choice to reduce dimensionality of the variable space with minimum loss of information, but it can be a slower learner than other techniques. As an attempt to exemplify which supervised classifier is best to use, Amancio and colleagues [Amancio et al., 2013] proposed a systematic comparison of supervised classifiers comparing the influence of parameter configuration on the accuracy. In their study, they found that using default parameters in the artificial dataset, kNN usually outperforms the other methods. Multilayer Perceptron showed to perform better than Bayesian Network, Decision Trees or SVM. However, they also showed that by altering those initial parameters the performance of SVM could implement up to 20%. This is just an example to illustrate that, even though there exist guidelines, there are no universal answers to which classifier is best.

4. Assessing classification performance

As mentioned before the assessment of the model's quality is a vital element of Machine Learning, not just to determine the predictive performance of a final model, but also to select the most suitable pre-processing and classifier meta-parameters. This section discusses popular performance metrics and common validation procedures that ensure metrics correctly represent actual performance.

4.1. Performance metrics

Suppose a dataset with 50 glaucoma and 50 healthy eyes. A model that never misclassifies glaucoma eyes, but misclassifies half of healthy eyes as glaucoma, will get 75 cases right. A second model that never makes a mistake for healthy eyes, but misclassifies half of glaucoma eyes as healthy, will also get 75 cases right. However, for a clinical standpoint there is a major difference in usefulness between both models. This illustrates that there are many aspects to accuracy testing and that there is no one single metric that can fully explain model performance.

The contingency matrix is the basis for most practical metrics and consists of 4 types of parameters corresponding with the numbers of cases that were either correctly or incorrectly classified (Table 3):

True Positive (TP): number of sick cases correctly categorised as sick.

True Negative (TN): number of healthy cases correctly categorised as healthy.

False Positive (FP, type I error): number of healthy cases wrongly classified as sick.

False Negative (FN; type II error): number of sick cases wrongly classified as healthy.

This concept can be expanded to multi-class problems, but will requires larger tables (Table 4. Ideally the False Positive (FP, Type I errors) and False Negative (FN, Type II errors) rates of a model should both be as close to zero as possible. Meanwhile, if TP ≈ FN and FP ≈ TN, the model is no better than random guessing.

Table 3: Two-class problem contingency matrix

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

TP=true positive, FN=false negative, FP=false positive, TN=true negative

Table 4: Empty four-class problem confusion matrix. Values in the diagonal (grey cells) would correspond to correct classifications

		Predicted condition			
		Class 1	**Class 2**	**Class 3**	**Class 4**
Actual	**Class 1**	-	-	-	-

condition	**Class 2**	-	-	-	-
	Class 3	-	-	-	-
	Class 4	-	-	-	-

The values in the contingency matrix depend on the dataset size and the relative proportions of the classes in the testing dataset, making comparisons between datasets cumbersome. For this reason several normalized measures were introduced:

Accuracy: (TP+TN)/(TP+FN+FP+TN), the ratio of correct classifications to the total number of classifications. A value of 1 corresponds with a perfect classification, whereas the ratio of classes in the testing dataset represents performance of random guessing. While accuracy is a common way to express quality of a model with a single number, it changes with the proportion of classes in the testing dataset, making comparisons difficult.

Precision: TP/(TP+FP), the proportion of positive results that are true positives, e.g. ratio of truly sick cases among all cases considered sick by the classifier. This is a measure of how much the model can be trusted when it classifies a case as positive.

Sensitivity or True Positive Rate (TPR): TP/(TP+FN), the ratio of correctly classified sick cases among all those subjects that actually carry the disease. This refers to the test's ability to correctly identify patients with the condition.

Specificity: TN/(TN+FP), the proportion of negatives correctly identified as such, i.e. percentage of healthy people who are correctly identified as not having the condition,

False Positive Rate (FPR): FP/(FP+TN), the ratio of the number of false positives among the number of all actual positive cases. This measures how much the model can be trusted when it classifies a case as negative.

If the output of a classification model is in the form of a score it is possible to create a Receiver Operating Characteristic (ROC) curve, a plot of the True Positive Rate versus the False Positive Rate for a model (Figure 9), with TPR and FPR computed for different thresholds for the model's score. Different points on the curve represent the different trade-offs in allowing false positives versus false negatives. A random performance model will result in an ROC being a straight diagonal line from the bottom left corner (TPR = FPR = 0) to the top right corner (TPR = FPR = 1), whereas a model with the perfect performance level will be a horizontal line from the top left to the top right corner. It is important to notice that classifiers with meaningful performance levels usually lie in the area between the random ROC curve and the perfect ROC curve.

ROC curves are also the basis for a popular metric called *Area Under the ROC curve* (AUC, AUROC). By measuring the area under the curve on an ROC model we estimate the probability of a positive (sick) case having higher score than a negative (healthy) case. A perfect model has AUC equal to 1, whereas for a random guessing model AUC is equal to 0.5. In other words, the closer the AUC is to 1, the better performance the classifier shows. This measure does not depend on the threshold selected by the practitioner and it can be compared across different dataset sizes and for datasets with different class balance. These advantages make it a popular choice to compare classifiers. However, some authors have questioned its efficiency [Hanczar et al., 2010] [Lobo et al., 2008].

By definition a ROC curve can only report the performance of binary classifiers. It is however possible to calculate an analogue for multi-classification problems, along with an AUC. This may be done by reducing the multi-class problem to a series of "one-vs-all" binary classification sub-problems, where each binary classification problem considers one of the original classes as the positive class, and all other classes are grouped into a negative class. AUC is computed for each of the binary sub-problems, and a mean of sub-problems' AUC is considered a metric for the original multi-class problem.

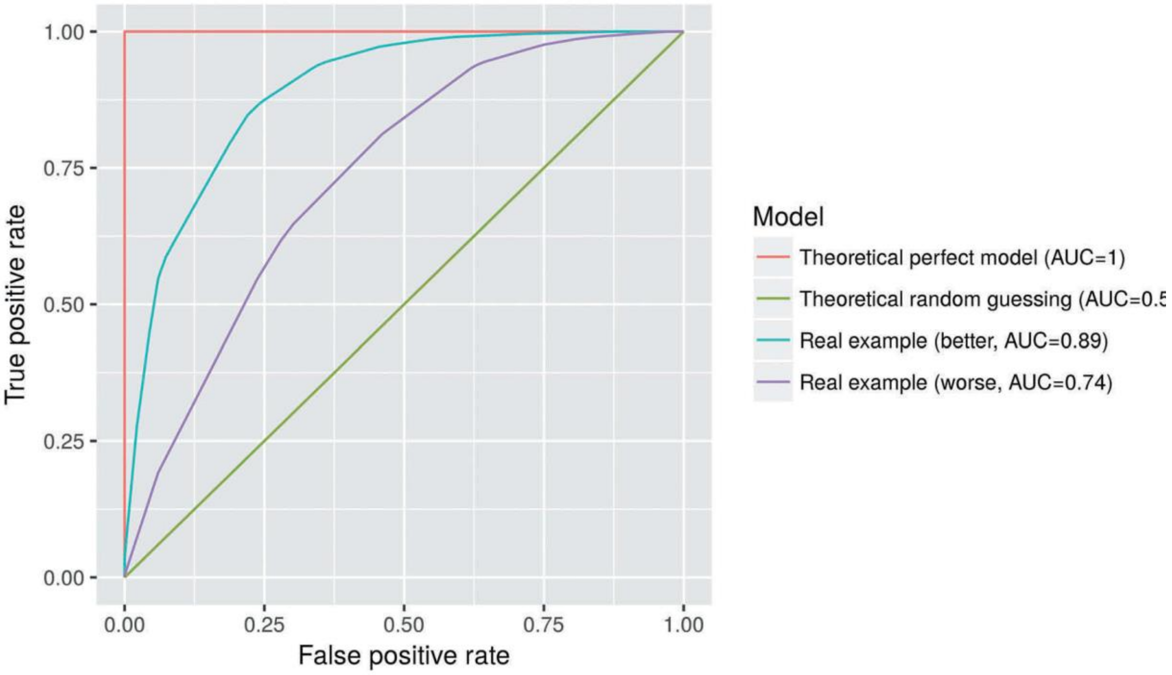

FIGURE 9. Examples of ROC curves for different theoretical and real models. Data corresponding to real examples is based on random forest calculation for an experiment about ocular dicrotism.[48.]

4.2. Validation procedures

An *in-sample metric* assesses model performance using training data, which may work well for low-capacity models. In-sample metrics may easily be fooled, however, when applied to a model with enough flexibility to memorize the training dataset. This does not necessarily mean that the model is overfitted, but rather indicates a need to use a dataset other than the training dataset to accurately measure performance. Also, in practice, it is rarely possible to accurately estimate model capacity, making it difficult to rely on in-sample metrics.

An additional problem with estimating model performance occurs at the point where the user compares multiple models and selects the most suitable one using a test set. Just like the training set may not be used to assess the performance of the model, it is not appropriate to consider a test set used to select the best performing model for model validation. In the classroom analogy this is similar to having a test to select the best student in a class to participate in a general knowledge quiz. Since each student has a specific amount of knowledge and individual interests (e.g. mathematics, biology…), it depends on the agreement between the questions in the selection test and those in the school quiz whether the student team that was ultimately selected has a chance of winning, since the initial assessment can randomly bias in favour of or against some topics.

For this reason one should use three independent datasets in the ideal case to avoid any bias: one training set, one test set (used to select the best model) and a validation set (used to compute an unbiased estimate of the final model's performance). The simplest approach to perform such a validation procedure is by dividing all available data into three datasets (training, validation and testing). This comes at the cost of having to train the model on a smaller amount of data than is actually collected, often resulting in a model with worse predictive performance. Similarly, the validation metrics are computed on a smaller subset, resulting in a less reliable estimate. The training problem is usually solved by developing a second model using the full dataset after meta-parameters were selected and their performance evaluated. While we will not have a proper estimate for the performance of this additional model, it is commonly assumed that it will work at least as well as the model with the same meta-parameters, but built only on the training dataset. This assumption comes from an intuition that adding more data should not make the model worse.

A popular choice to avoid the validation problem is to use a procedure that trains models multiple times, on many different subsets of data. Some of procedures in this class are *Cross-Validation* (CV) [Stone, 1974] and the Bootstrap method [Efron, 1979]. Cross-validation consists of dividing the available dataset into equally-sized subsets called "*folds*", usually 5 or 10 in number or single-case folds ("Leave One Out" CV [Lachenbruch, 1968]). Then, the procedure consists of training and testing as many models as there are folds: each time, a different fold is chosen as the test set, while all other folds compose the training set for the model. Out-of-sample measures for each fold are then averaged to form an estimate of model performance that is effectively computed on all available data, while taking possible overfitting issues into account [Kohavi, 1995].

5. Machine learning in ophthalmology and optometry

From the early nineties onward Machine Learning has become an increasingly popular tool to assist clinicians to identify certain ocular conditions. The large amount of data generated by new diagnostic techniques, specially imaging techniques, often hinders the interpretation of the results. To overcome this challenge machine learning techniques arose as an effective complementary tool to the practitioner´s criteria, not only for diagnosis but in recent years also as monitoring and prognostic tool of certain ocular conditions.

For a better understanding of this section is important to mind that when reporting performance of classification methods (AUCs, sensitivity, specificity) among studies, severity disease and sample size should be considered. For instance, it will be likely easier to classify glaucoma eyes as glaucomatous when the average mean deviation is -6 dB compared to -2.5 dB. Also, it is also essential to understand that to compare the sensitivity of techniques, ideally they should be compared at fixed specificities, since when sensitivity increases specificity decreases (and vice versa).

5.1. Glaucoma

Glaucoma is an ocular disease that, if untreated, causes damage to the optic nerve, leading to vision loss and possibly blindness. It is one of the ocular pathologies where the most efforts regarding Machine Learning implementation have been made. Early works with Machine Learning in glaucoma research used Neural Networks to differentiate the visual fields of healthy and glaucomatous eyes [Goldbaum et al., 1990] [Goldbaum et al., 1991]. Other works related to visual function testing compared different classifiers for their ability to discriminate between healthy and glaucomatous eyes [Goldbaum et al., 2002]. Comparisons between machine learning techniques and the built-in indices of commercially available instruments were also performed [Lietman et al., 1999]. Unsupervised Machine Learning was also used in the early steps of automated glaucoma detection by classifying visual field data [Henson et al., 1996]. However, after the popularization of imaging techniques for glaucoma monitoring, Machine Learning tended to restrict to supervised learning. Using optimized data from a confocal scanning laser ophthalmoscope and different types of classifiers such as SVM, Neural Networks and Linear

Discriminant Analysis on nearly 300 retinal tomography images Bowd et al. [Bowd et al., 2002] reported AUC ranged from 0.90 to 0.96. Further works used optic disc measurements acquired with Optical Coherence Tomography (OCT) to differentiate glaucoma in 189 eyes with AUC of 0.87 [Huang et al., 2005]. An accuracy of 83% was obtained using OCT data of 135 eyes and by combining unsupervised learning and Decision Trees [Huang et al., 2007]. Burgansky-Eliash [Burgansky-Eliash et al., 2005] compared different classifiers on 89 eyes and reported AUC of 0.98 using SVM trained on eight OCT features related to visual field. Pixel by pixel data obtained using imaging data has also been evaluated using Machine Learning classifiers to improve pattern detection. In fact, applying image processing techniques prior analysis has become an useful aid to increase the detection success rate [Khalil et al., 2014][Kumar et al., 2018].Similarly, automated detection of glaucoma using Machine Learning techniques has proved to be a successful method with accuracy over 85% [Khalil et al., 2014]. In the last years, unsupervised learning has resurged in glaucoma research to identify patterns of glaucomatous visual field loss. Over 2000 healthy and glaucomatous eyes were automatically separated in three different clusters depending on severity of the field loss with a Bayesian-independent component analysis [Goldbaum, 2009]. Elze and colleagues utilized over 13 000 images to identify 17 different vision loss prototypes easy to interpret for clinicians [Elze et al., 2014]. Also using unsupervised learning, Yousefi et at. [2016] showed that the detection of glaucomatous progression can be improved by assessing longitudinal changes in localized patterns of glaucomatous eyes. However, little work has been done for detection of glaucoma in early stages [Asaoka et al., 2017] or prediction of the disease [Bowd et al., 2004]. Future research will likely focus on these matters.

5.2. Diabetic retinopathy

As diabetes progresses there in an increasing risk that the disease may affect retinal blood vessels, leading to diabetic retinopathy (DR). DR is the main cause of vision deficiency and blindness among working-age adults, and its detection by image processing and Machine Learning techniques has recently gained interest. The potential of neural networks as a useful tool for telemedicine was already pointed out by some early works [Williamson et al., 1998][Gardner el at., 1996]. Jelinek et al. [Jelinek et al., 2011], reported 80% accuracy (90% sensitivity, 80% specificity) in differentiating healthy and DR eyes using SVM on 1100 images. In clinical practice the early detection of DR is often a challenge. A combination of k-nearest neighbour and linear discriminant classifiers was used on 430 retinal images to detect early cases of DR with a resulting 0.95 AUC [Niemeijer et al., 2007]. Focusing on feature selection, using Naïve Bayes and SVM, Sopharak et al. [Sopharak et al., 2009] reported 92.28% sensitivity, 98.52% specificity and 98.41% accuracy in the detection of exudates, one of the preliminary signs of DR. Detecting DR is as important as grading the disease. Many works on DR and Machine Learning aimed to develop an accurate grading system of DR. Image processing techniques and Artificial Neural Networks were successfully applied on 1273 subjects to develop a DR screening and grading system with no missing threatening cases of DR at a setting with 94.8% sensitivity and 52.8% specificity [Usher et al., 2003]. Similarly, from 124 processed raw images and Artificial Neural Networks reported 80% accuracy of correct grading classification, 90% sensitivity and 100% specificity [Yun et al., 2008]. DREAM (Diabetic Retinopathy Analysis using Machine Learning) is another grading system of DR developed using different classifiers on 1200 images (100% sensitivity, 53.16% specificity and 0.904 AUC) [Roychowdhury et al., 2014]. Morphological image processing and SVM were applied on 331 fundus images to classify the severity of the DR within different groups (82% sensitivity, 86% specificity) [Acharya et al., 2009]. Quellec et al. [Quellec et al., 2012] presented a learning framework for DR grading that avoids manual segmentation (i.e., process of partitioning a digital image into several segments). In addition, using thousands of images they showed how the AUC improves when incrementing the number of examination records. Recently, a worldwide competition organized by Kaggle and sponsored by the California Healthcare Foundation put 35 thousand eye images at the disposal of over 600 competing teams to identify signs of diabetic retinopathy [Graham, 2015]. This competition has been so far, the largest application of Machine

Learning for the diagnosis of this ocular condition. Moreover, deep learning in the analysis of retinal images has made a tremendous progress in the last few years. In opposition to previous works focused on computing explicit features specified by experts, deep learning learns directly from images given a large data set of labelled examples. From the same extensive public database different research works aimed to prove the better performance of deep learning over more traditional machine learning techniques for the automated detection of DR [Gulshan et al., 2016][Abramoff et al., 2016][Gargeya & Leng, 2017].

Furthermore, since April 2018 the U.S. Food and Drug Administration (FDA) permits marketing of the first medical device to use artificial intelligence to detect greater than a mild level of DR in adults who have diabetes. Retinal telescreening for evaluation DR in the primary care setting may be useful in reaching rural and underserved patients [Jani et al., 2017][Mookiah, 2013].

5.3. Age-related macular degeneration

Age-related macular degeneration (AMD) is the leading cause of irreversible blindness in people over 50 in the developed world. Most of AMD-related lesions are detected visually by clinicians, which leads to subjectivity and consequently inter-observer variability. Accurate, automated identification of AMD-related lesions is the challenge of Machine Learning techniques within this area. Lahmiri et al [Lahmiri et al, 2014] detected ring-shaped exudates in retina digital images using digital imaging processing and SVM for classification of 45 colour fundus photographs reaching an excellent performance. Other authors used SVM, kNN, Naïve Bayes and Neural Networks to present a dry AMD screening tool with a built-in risk index, reaching 93.70% accuracy, 91.11% sensitivity and 96.30% specificity [Mookiah et al., 2014]. Similarly, automated characterization of geographic atrophy (GA) from colour fundus photographs was done employing Random Forests [Freeny et al., 2015]. Unlike the former works, no data pre-processing was applied in a study solely based on clinical signs and patient's data retinal information that used different classification algorithms, such as Random Forest and SVM, to improve AMD diagnosis reaching over 0.90 AUC [Fraccaro et al., 2015].

5.4. Keratoconus

Classification of corneal topography has been of interest since the early days of Machine Learning in Ophthalmology [Maeda et al., 1995], aiming especially at the detection of early keratoconus [Smolek et al., 1997]. This disease causes a gradual deformation of the cornea, often leading to a serious impairment of the visual quality.

Zernike polynomials were used as variables and Decision Trees as classifier on 244 eyes to differentiate keratoconus eyes from normal eyes with 93% accuracy [Twa el al., 2005]. A follow-up of this work examined Pseudo-Zernike polynomials on 254 eyes measured with a Optikon Keratron topographer. Neural Networks, C4.5 Decision Trees and Naive Bayes classifiers were utilized. Overall, they found the speed, accuracy, stability and interpretability of decision trees preferable to other methods for the dataset. In addition, they showed that Zernike polynomials provide a better variable representation than pseudo-Zernikes. [Marsolo et al., 2007]. A Neural Network, based on basic topographic and tomographic variables acquired by videokeratography, was used to classify 166 eyes as keratoconus or normal, reaching 0.991 AUC, sensitivity of 94% and a specificity of 100% [Saad et al., 2014].

Using 318 Orbscan II maps classified into four classes, normal, astigmatism, keratoconus and photorefractive keratectomy [Souza et al., 2010] reported 0.99 AUC for detecting keratoconus apart from the other non-keratoconus patterns with both SVM and Neural Networks. [Arbelaez et al., 2012] demonstrated using SVM on 3502 eyes divided into four different categories (i.e., normal, keratoconus, subclinical keratoconus and eyes with corneal surgery history) and measured using a Scheimpflug camera combined with Placido corneal topography that precision improves when including posterior corneal surface data, especially for classifying subclinical keratoconus. Another attempt for detecting subclinical keratoconus used Classification Trees on 372 eyes and 55 variables derived from anterior and posterior corneal Scheimpflug measurements. It reached 93.6% sensitivity and 97.2% specificity when classifying between normal eyes and subclinical keratoconus [Smadja et al., 2013]. Similarly, an algorithm based on semi-supervised learning proposed by [Cheboli et al., 2008] reached 95% accuracy in the detection of keratoconus suspects. Artificial Neural Networks were used on a 288-keratoconic-eye database to predict astigmatism in patients with keratoconus after ring implantation [Valdés-Mas et al., 2014]. This work reported a correlation coefficient of 0.92 between predicted and real values. On the other hand, the importance of using automated classifiers trained on bilateral data to discriminate between healthy and normal fellow corneas was pointed out by [Kovács et al., 2016]. Recently, [Ruiz-Hidalgo et al., 2016] reported 98.8% accuracy classifying between keratoconus and normal eyes and 93.1% accuracy discriminating between forme fruste and normal eyes, using 22 biometrical parameters as variables and SVM as classifier on 860 eyes. As indicated, many works have already showed a high success rate when classifying keratoconus from healthy eyes using different Machine Learning classifiers. Nowadays the open challenges are improving the success rate for the early diagnosis of the condition; classify the stage of the disease and predict its evolution.

5.5. Refractive error assessment

Using data from an autorefractometer and Hartmann-Shack device, Machine Learning was implemented to predict refractive error [Libralao et al., 2004]. However due to the large number of classes and small number of samples per class involved, the error was relatively high in comparison with other Machine Learning applications, e.g. around 18 % error using SVM in the assessment of the sphere, enhanced to 13 % error also using SVM but with artificially balanced classes. A more sophisticated approach using a classifier ensemble (i.e., combing several classifiers individually trained) was used for the same purpose and enhanced it to 95% certainty [Libralao et al., 2005]. A recent study using Neural Networks that aimed to predict the power vectors of refraction based on 460 eyes showed comparable results between subjective refraction and the Machine Learning results [Ohlendorf et al., 2017].

5.6. Corneal endothelium

The first attempt at detecting the corneal endothelium cell boundaries was presented in the early nineties, based on a combination of image processing techniques and Neural Networks [Zhang et al., 1991]. A more intricate approach, also based on Neural Network, optimizes the weights between the input and first hidden layers to detect the boundaries of the human corneal endothelium [Hasegawa et al., 1996]. Neural Networks were later also used in identifying abnormalities in the different corneal layers acquired with confocal microscopy, reaching almost 100 % accuracy [Sharif et al., 2015]. In other works Neural Networks were applied to epithelial and stromal thickness data in order to differentiate between keratoconus and normal eyes with a relatively high success rate (99.2 specificity and 94.6% sensitivity) [Silverman et al., 2014].

5.7. Dry eye disease

Dry eye disease, one of the most common eye conditions, occurs when either the eye does not produce enough tears or when the tears evaporate too quickly. Despite the high prevalence of the condition, diagnostic methods are usually slit lamp- or questionnaire-based, often leading to an inconclusive diagnosis. Machine Learning techniques have shown their potential to standardize dry eye diagnostics. Classification of 55 infrared Meibomian gland images by means of image processing and linear SVM achieved specificity of 96% and sensitivity of 98% when differentiating between dry eye and healthy eyes [Koh et al., 2012]. Another approach used 105 slit-lamp images to evaluate the thickness of the tear film's lipid layer by means of image processing and various classifiers, of which SVM was the most accurate (81.90%) [Remeseiro et al., 2012] [Ramos et al., 2011].

5.8. Nuclear cataracts

The presence and severity of cataracts and lens opacification is often assessed by comparing a slit lamp exam with a set of standardized photographs (e.g. LOCS III [Chylack et al., 1993]) which tends to be inefficient, time consuming and imprecise. To overcome this limitation a method to grade cataracts based on SVM was proposed by [Cheung et al., 2011]. When applied to a set of 5750 slit-lamp images the performance of the automatic method was more successful (99.4% images correctly graded) than the subjective method (97.0%). A follow-up of this work [Xu et al., 2013] proposed another automated method based on feature selection, parameter selection and a regression model training simultaneously on a large dataset of 5378 slit-lamp images, confirming the superior performance of automatic machine-learning methods over traditional manual methods in grading cataract severity.

5.9. Other applications

Other applications in ophthalmology using Machine Learning that have been reported are for example, diagnosis of achromatopsia or congenital stationary night blindness by analysing electroretinograms using Artificial Neural Networks, presented as a potentially powerful tool to enhance routine clinical examinations [Bagheri et al., 2014]. Artificial Neural Networks have also shown their potential to improve the ability of optical coherence tomography to detect optic neuritis [Garcia-Martin et al., 2013].

Furthermore, the detection of the excessive ciliary muscle activity using kNN and SVM applied to a set of 40 thermal images has been reported [Harangi et al., 2011], as well as analysing eye tracking to diagnose dyslexia using SVM [Rello et al., 2015].

Also, image processing techniques supported by Neural Networks showed their potential as a successful methodology for iris segmentation using visible wavelength iris images captured at-a-distance and on-the-move, which are challenging conditions that lead to severely degraded image data [Proenca et al., 2010][Proenca, 2010]. Finally, there is a classic example in Machine Learning text books for beginners called 'The contact lens data', an oversimplified contact lens fit problem used to teach the usefulness of decision trees [Witten et al., 2016]. A more accurate approach of this example could be consider to study which are the most critical parameters when fitting contact lenses, especially nowadays due to the rise of potentially new important fitting parameters [Consejo et al., 2017][Consejo et al., 2018], question that has been intriguing optometrists for decades.

6. Practical tips of using Machine Learning

6.1. Machine Learning Software

There are many free open-source tools to implement Machine Learning techniques. Among them Weka (University of Waikato, New Zealand, www.cs.waikato.ac.nz/ml/weka) is easy to use by offering a Java-based graphical user interface and tools for data pre-processing (filters), classification, regression and visualization [Hall et al., 2009]. Similar programs are Orange

(University of Ljubljana, Slovenia, https://orange.biolab.si/), which is popular for its user-friendly approach, and KNIME (Zurich, Switzerland, www.knime.org/), another well-known Machine Learning platform with a free edition. There are also more intricate and powerful commercial software options that require various levels of programming skills, such as MATLAB (The MathWorks, Inc., USA), the free statistical software package R (R Development Core Team, www.r-project.org/) and the programming language Python (Python Software Foundation, www.python.org/). From these options Python is the most popular programming languages used for Machine Learning, but ultimately the selection will depend on the characteristics of the problem at hand and user's familiarity the programming languages. It is important to note that the computational effort required to process very large datasets requires a powerful computer. For such applications several cloud solutions are available, such as Amazon Machine Learning (Amazon Web Services, Inc., USA, aws.amazon.com/machine-learning/) and Google Cloud Platform (https://cloud.google.com/).

6.2. Further reading

So far, the references in this work have mostly been restricted to recent peer-reviewed papers and conference proceedings. In addition to these, there are many books covering the principles of Machine Learning, but unfortunately only few of them are written at a beginners' level. Some of the most popular and novel books for readers with programming skills and interested in building their own algorithms are [Witten et al., 2016] or [Segaran, 2017]. A brief overview of what Machine Learning can offer can be found in [Dutton et al., 1997], while Hastie (2008) [Hastie, 2008] is one of the most popular and complete manuals on the subject. There are also many resources for readers interested in a particular classification algorithm, such as Bishop's textbook regarding Neural Networks [Bishop, 1995], one of the most cited in the field. Similarly, Murthy (1998) [Murthy, 1998] provides an overview of decision trees, Furnkranz (1999) [Furnkranz, 1999] gives an overview of rule-based methods, and Wettschereck et al. (1997) [Wettschereck et al., 1997] presents a review of instance-based learning classifiers, such as kNN. Tutorials on SVM may be found in [Burges, 1998] or Cristianini (2000) [Cristianini et al., 2000].

7. Conclusions

Machine Learning is a powerful, relatively easy to implement tool with infinite possibilities to enhance clinical practice and its potential applications in ophthalmology are mostly limited by the researcher's imagination. Even though the majority of works use Machine Learning as a diagnostic tool, the possibilities of this technology go much further than just the identification of a certain ocular conditions, such as grading pathologies, early detection of diseases and predicting the evolution of certain conditions.

The large number of imaging and image processing techniques available nowadays present new opportunities to develop decision-support tools that assist clinicians with the diagnosis of almost any ocular condition. One important issue with images that the pre-processing by means of filters or contrast techniques prior to machine learning analysis may cause the physical meaning of the variables to get lost, thus making the results more difficult to interpret. Thus Machine Learning risks becoming a 'black box'. For this the reason most researchers prefer to stick to more tangible variables such as ocular parameters, rather than images, and make an a priori selection of the most important parameters for use during the training phase. This would restrict the possibilities to only those parameters that have a logical meaning, however, excluding potential unknown influences. It is therefore best to adopt a flexible approach in one's choice of machine learning technique, always adopting it to the data type and context. Regarding the size of the original dataset it is always better to have a larger number of subjects (instances), but one should be cautious with the number of variables (parameters) included to avoid overfitting. Finally, it is important to apply appropriate data pre-processing to maximize the chance of obtaining a successful classification tool.

References

1. Abakar KA, Yu C. Performance of SVM based on PUK kernel in comparison to SVM based on RBF kernel in prediction of yarn tenacity. IJFTR. 2014; 39:55-59.
2. Abràmoff MD, Lou Y, Erginay A, Clarida W, Amelon R, Folk JC, et al. Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. Invest Ophthalmol Vis Sci 2016;57(13):5200-5206.
3. Acharya UR, Lim CM, Ng EYK, Chee C, Tamura T. Computer-based detection of diabetes retinopathy stages using digital fundus images. Proc Inst Mech Eng Part H J Eng Med. 2009;223(5):545-53.
4. Airoldi EM. Getting started in probabilistic graphical models. PLoS Comput Biol. 2007;3(12):e252.
5. Amancio DR, Comin CH, Casanova D, Travieso G, Bruno OM, Rodrigues FA, et al. A systematic comparison of supervised classifiers. PloS one 2014;9(4):e94137.
6. Arbelaez MC, Versaci F, Vestri G, Barboni P, Savini G. Use of a support vector machine for keratoconus and subclinical keratoconus detection by topographic and tomographic data. Ophthalmology. 2012;119(11):2231-8.
7. Asaoka R, Hirasawa K, Iwase A, Fujino Y, Murata H, Shoji N, et al. Validating the Usefulness of the “Random Forests” Classifier to Diagnose Early Glaucoma With Optical Coherence Tomography. Am J Ophthalmol. 2017;174:95-103.
8. Bagheri A, Adorno DP, Rizzo P, Barraco R, Bellomonte L. Empirical mode decomposition and neural network for the classification of electroretinographic data. Med Biol Eng Comput. 2014;52(7):619-28.
9. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. 2014.
10. Batista GE, Monard MC. An analysis of four missing data treatment methods for supervised learning. Appl Artif Intell. 2003;17(5-6):519-33.
11. Bishop CM. Neural networks for pattern recognition. Oxford university press; 1995.
12. Bowd C, Chan K, Zangwill LM, Goldbaum MH, Lee T, Sejnowski TJ, et al. Comparing neural networks and linear discriminant functions for glaucoma detection using confocal scanning laser ophthalmoscopy of the optic disc. Invest Ophthalmol Vis Sci. 2002;43(11):3444-54.
13. Bowd C, Zangwill LM, Medeiros FA, Hao J, Chan K, Lee T, et al. Confocal scanning laser ophthalmoscopy classifiers and stereophotograph evaluation for prediction of visual field abnormalities in glaucoma-suspect eyes. Invest Ophthalmol Vis Sci. 2004;45(7):2255-62.
14. Breiman L. Random forests. Mach Learning. 2001;45(1):5-32.
15. Burgansky-Eliash Z, Wollstein G, Chu T, Ramsey JD, Glymour C, Noecker RJ, et al. Optical coherence tomography machine learning classifiers for glaucoma detection: a preliminary study. Invest Ophthalmol Vis Sci. 2005;46(11):4147-52.
16. Burges CJ. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery. 1998;2(2):121-67.
17. Cheboli D, Ravindran B. Detection of Keratoconus by Semi-Supervised Learning. . 2008.
18. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; ACM; 2016.
19. Cheung CY, Li H, Lamoureux EL, Mitchell P, Wang JJ, Tan AG, et al. Validity of a new computer-aided diagnosis imaging program to quantify nuclear cataract from slit-lamp photographs. Invest Ophthalmol Vis Sci. 2011;52(3):1314-9.
20. Chylack LT, Wolfe JK, Singer DM, Leske MC, Bullimore MA, Bailey IL, et al. The lens opacities classification system III. Arch Ophthalmol. 1993;111(6):831-6.
21. Consejo A, Bartuzel MM, Iskander DR. Corneo-scleral limbal changes following short-term soft contact lens wear. Cont Lens Anterior Eye 2017;40(5):293-300.
22. Consejo A, Behaegel J, Van Hoey M, Wolffsohn JS, Rozema JJ, Iskander DR. Anterior eye surface changes following miniscleral contact lens wear. Cont Lens Anterior Eye 2018 (doi.org/10.1016/j.clae.2018.06.005)
23. Cortes C, Vapnik V. Support-vector networks. Mach Learning. 1995;20(3):273-97.
24. Cristianini N, Shawe-Taylor J. An introduction to support vector machines. . 2000.
25. Danielewska ME, Iskander DR, Krzyzanowska-Berkowska P. Age-related changes in corneal pulsation: ocular dicrotism. Optometry & Vision Science. 2014;91(1):54-9.
26. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society.Series B (methodological). 1977:1-38.
27. Dhar V. Data science and prediction. Commun ACM. 2013;56(12):64-73.
28. Ding C, He X. K-means clustering via principal component analysis. Proceedings of the 21st International Conference on Machine Learning, 2004, pp 1-9.
29. Donders ART, van der Heijden, Geert JMG, Stijnen T, Moons KG. Review: a gentle introduction to imputation of missing values. J Clin Epidemiol. 2006;59(10):1087-91.
30. Dutton DM, Conroy GV. A review of machine learning. The Knowledge Engineering Review. 1997;12(4):341-67.

31. Elze T, Pasquale LR, Shen LQ, Chen TC, Wiggs JL, Bex PJ. Patterns of functional vision loss in glaucoma determined with archetypal analysis. J R Soc Interface 2015; 12(103):10.1098/rsif.2014.1118.
32. Efron B. Bootstrap methods: another look at the jackknife. The annals of Statistics. 1979:1-26.
33. Feeny AK, Tadarati M, Freund DE, Bressler NM, Burlina P. Automated segmentation of geographic atrophy of the retinal epithelium via random forests in AREDS color fundus images. Comput Biol Med. 2015;65:124-36.
34. Fraccaro P, Nicolo M, Bonetto M, Giacomini M, Weller P, Traverso CE, et al. Combining macula clinical signs and patient characteristics for age-related macular degeneration diagnosis: a machine learning approach. BMC ophthalmology. 2015;15(1):10.
35. Friedman JH. Greedy function approximation: a gradient boosting machine. Annals of statistics. 2001:1189-232.
36. Fürnkranz J. Separate-and-conquer rule learning. Artif Intell Rev. 1999;13(1):3-54.
37. Futoma J, Sendak M, Cameron CB, Heller K. Predicting Disease Progression with a Model for Multivariate Longitudinal Clinical Data. Proceedings of the 1st Machine Learning for Healthcare Conference; ; 2016.
38. Garcia-Martin E, Herrero R, Bambo MP, Ara JR, Martin J, Polo V, et al. Artificial neural network techniques to improve the ability of optical coherence tomography to detect optic neuritis. Seminars in Ophthalmology. 2015; 30(1): 11-19.
39. Gardner GG, Keating D, Williamson TH, Elliott AT. Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool. Br J Ophthalmol. 1996;80(11):940-4.
40. Gargeya R, Leng T. Automated identification of diabetic retinopathy using deep learning. Ophthalmology 2017;124(7):962-969.
41. Goldbaum MH, Jang GJ, Bowd C, Hao J, Zangwill LM, Liebmann J, et al. Patterns of glaucomatous visual field loss in sita fields automatically identified using independent component analysis. Trans Am Ophthalmol Soc 2009;107:136-144.
42. Goldbaum MH, Sample PA, White H, Weinreb R. Discrimination of normal and glaucomatous visual fields by neural network. Invest Ophthalmol Vis Sci. 1990;31:503.
43. Goldbaum MH, Sample PA, Chan K, Williams J, Lee T, Blumenthal E, et al. Comparing machine learning classifiers for diagnosing glaucoma from standard automated perimetry. Invest Ophthalmol Vis Sci. 2002;43(1):162-9.
44. Goldbaum MH, Sample PA, White H, Colt B, Raphaelian P, Fechtner RD, et al. Interpretation of automated perimetry for glaucoma by neural network. Invest Ophthalmol Vis Sci. 1994;35(9):3362-73.
45. Graham B. Kaggle diabetic retinopathy detection competition report. 2015.
46. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA 2016;316(22):2402-2410.
47. Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of machine learning research. 2003;3(Mar):1157-82.
48. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter. 2009;11(1):10-8.
49. Hanczar B, Hua J, Sima C, Weinstein J, Bittner M, Dougherty ER. Small-sample precision of ROC-related estimates. Bioinformatics. 2010 Mar 15;26(6):822-30.
50. Hand DJ, Yu K. Idiot's Bayes—not so stupid after all? International statistical review. 2001;69(3):385-98.
51. Harangi B, Csordás T, Hajdu A. Detecting the excessive activation of the ciliaris muscle on thermal images. Applied Machine Intelligence and Informatics (SAMI), 2011 IEEE 9th International Symposium on; IEEE; 2011.
52. Harman HH. Modern factor analysis. . 1960.
53. Hasegawa A, Itoh K, Ichioka Y. Generalization of shift invariant neural networks: image processing of corneal endothelium. Neural Networks. 1996;9(2):345-56.
54. Hastie T, Tibshirani R. Classification by pairwise coupling. Annals of statistics. 1998;26(2):451-71.
55. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York; 2009.
56. Haykin S, Network N. A comprehensive foundation. Neural Networks. 2004;2(2004):41.
57. Henson DB, Spenceley SE, Bull DR. Spatial classification of glaucomatous visual field loss. Br J Ophthalmol. 1996 Jun;80(6):526-31.
58. Hinton G, Deng L, Yu D, Dahl GE, Mohamed A, Jaitly N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process Mag. 2012;29(6):82-97.
59. Hodge V, Austin J. A survey of outlier detection methodologies. Artif Intell Rev. 2004;22(2):85-126.
60. Hofmann T, Schölkopf B, Smola AJ. Kernel methods in machine learning. The annals of statistics. 2008:1171-220.
61. Huang M, Chen H. Development and comparison of automated classifiers for glaucoma diagnosis using Stratus optical coherence tomography. Invest Ophthalmol Vis Sci. 2005;46(11):4121-9.
62. Huang M, Chen H, Lin J. Rule extraction for glaucoma detection with summary data from StratusOCT. Invest Ophthalmol Vis Sci. 2007;48(1):244-50.
63. Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM computing surveys (CSUR). 1999;31(3):264-323.
64. Jain A, Zongker D. Feature selection: Evaluation, application, and small sample performance. IEEE Trans Pattern Anal Mach Intell. 1997;19(2):153-8.
65. Jani PD, Forbes L, Choudhury A, Preisser JS, Viera AJ, Garg S. Evaluation of diabetic retinal screening and factors for ophthalmology referral in a telemedicine network. JAMA ophthalmology 2017;135(7):706-714.
66. Jelinek HF, Rocha A, Carvalho T, Goldenstein S, Wainer J. Machine learning and pattern classification in identification of indigenous retinal pathology. Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE; IEEE; 2011.

67. Jolliffe I. Principal component analysis. Wiley Online Library; 2002.
68. Khalil T, Khalid S, Syed AM. Review of Machine Learning techniques for glaucoma detection and prediction. Science and Information Conference (SAI), 2014; IEEE; 2014.
69. Koh YW, Celik T, Lee HK, Petznick A, Tong L. Detection of meibomian glands and classification of meibography images. J Biomed Opt. 2012;17(8):0860081-7.
70. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. Ijcai; Stanford, CA; 1995.
71. Kohavi R, Provost F. Glossary of terms. Mach Learning. 1998;30(2-3):271-4.
72. Kohonen T. Self-organizing map. Proc IEEE 1990;78:1464-1480.
73. Koller D, Sahami M. Toward optimal feature selection. Technical Report. Stanford InfoLab. 1996.
74. Kotsiantis SB, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques. 2007.
75. Kovács I, Miháltz K, Kránitz K, Juhász É, Takács Á, Dienes L, et al. Accuracy of machine learning classifiers using bilateral data from a Scheimpflug camera for identifying eyes with preclinical signs of keratoconus. Journal of Cataract & Refractive Surgery. 2016;42(2):275-83.
76. Kumar BN, Chauhan R, Dahiya N. Detection of glaucoma using image processing techniques: A critique. Seminars in Ophthalmology. 2018. 33. (2) 275-283.
77. Lachenbruch PA, Mickey MR. Estimation of error rates in discriminant analysis. Technometrics. 1968;10(1):1-11.
78. Lahmiri S, Boukadoum M. Automated detection of circinate exudates in retina digital images using empirical mode decomposition and the entropy and uniformity of the intrinsic mode functions. Biomedical Engineering/Biomedizinische Technik. 2014;59(4):357-66.
79. Langley P, Sage S. Induction of selective Bayesian classifiers. Proceedings of the Tenth international conference on Uncertainty in artificial intelligence; Morgan Kaufmann Publishers Inc.; 1994.
80. Libralao GL, de Almedia O, Netto AV, Delbem A, Leon A, de Carvalho F. Machine learning techniques for ocular errors analysis. Machine Learning for Signal Processing, 2004. Proceedings of the 2004 14th IEEE Signal Processing Society Workshop; IEEE; 2004.
81. Libralao G, Almeida O, Carvalho A. Classification of ophthalmologic images using an ensemble of classifiers. Innovations in Applied Artificial Intelligence. 2005:6-13.
82. Lietman T, Eng J, Katz J, Quigley HA. Neural Networks for Visual Field Analysis: How Do They Compare with Other Algorithms?. J Glaucoma. 1999;8(1):77-80.
83. Little RJ, Rubin DB. Statistical analysis with missing data. John Wiley & Sons; 2014.
84. Lobo JM, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models. Global Ecol Biogeogr. 2008;17(2):145-51.
85. Maeda N, Klyce SD, Smolek MK. Neural network classification of corneal topography. Preliminary demonstration. Invest Ophthalmol Vis Sci. 1995;36(7):1327-35.
86. Marsolo K, Twa M, Bullimore MA, Parthasarathy S. Spatial modeling and classification of corneal shape. IEEE Transactions on Information Technology in Biomedicine. 2007;11(2):203-12.
87. Melcer T, Danielewska ME, Iskander DR. Wavelet representation of the corneal pulse for detecting ocular dicrotism. PloS one. 2015;10(4):e0124721.
88. Mookiah MRK, Acharya UR, Chua CK, Lim CM, Ng E, Laude A. Computer-aided diagnosis of diabetic retinopathy: A review. Comput Biol Med 2013;43(12):2136-2155.
89. Mookiah MRK, Acharya UR, Koh JE, Chua CK, Tan JH, Chandran V, et al. Decision support system for age-related macular degeneration using discrete wavelet transform. Med Biol Eng Comput. 2014;52(9):781-96.
90. Murthy SK. Automatic construction of decision trees from data: A multi-disciplinary survey. Data mining and knowledge discovery. 1998;2(4):345-89.
91. Niemeijer M, van Ginneken B, Russell SR, Suttorp-Schulten MS, Abramoff MD. Automated detection and differentiation of drusen, exudates, and cotton-wool spots in digital color fundus photographs for diabetic retinopathy diagnosis. Invest Ophthalmol Vis Sci. 2007;48(5):2260-7.
92. Ohlendorf A, Leube A, Leibig C, Wahl S. A machine learning approach to determine refractive errors of the eye. Investigative Ophthalmology & Visual Science. 2017.
93. Olazaran M. A sociological study of the official history of the perceptrons controversy. Soc Stud Sci. 1996;26(3):611-59.
94. Proenca H, Filipe S, Santos R, Oliveira J, Alexandre LA. The ubiris. v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. IEEE Trans Pattern Anal Mach Intell 2010;32(8):1529-1535.
95. Proenca H. Iris recognition: On the segmentation of degraded images acquired in the visible wavelength. IEEE Trans Pattern Anal Mach Intell 2010;32(8):1502-1516.
96. Quellec G, Lamard M, Abràmoff MD, Decencière E, Lay B, Erginay A, et al. A multiple-instance learning framework for diabetic retinopathy screening. Med Image Anal. 2012;16(6):1228-40.
97. Quinlan JR. Induction of decision trees. Mach Learning. 1986;1(1):81-106.
98. Ramos L, Penas M, Remeseiro B, Mosquera A, Barreira N, Yebra-Pimentel E. Texture and color analysis for the automatic classification of the eye lipid layer. Advances in computational intelligence. 2011:66-73.
99. Rello L, Ballesteros M. Detecting readers with dyslexia using machine learning with eye tracking measures. Proceedings of the 12th Web for All Conference; ACM; 2015.
100. Remeseiro B, Penas M, Mosquera A, Novo J, Penedo M, Yebra-Pimentel E. Statistical comparison of classifiers applied to the interferential tear film lipid layer automatic classification. Computational and mathematical methods in medicine. 2012;2012.

101. Roychowdhury S, Koozekanani DD, Parhi KK. Dream: Diabetic retinopathy analysis using machine learning. IEEE journal of biomedical and health informatics. 2014;18(5):1717-28.
102. Rozema JJ, Zakaria N, Ruiz Hidalgo I, Jongenelen S, Tassignon MJ, Koppen C. How Abnormal Is the Noncorneal Biometry of Keratoconic Eyes? Cornea. 2016 Jun;35(6):860-5.
103. Ruiz Hidalgo I, Rodriguez P, Rozema JJ, Ni Dhubhghaill S, Zakaria N, Tassignon MJ, et al. Evaluation of a Machine-Learning Classifier for Keratoconus Detection Based on Scheimpflug Tomography. Cornea. 2016 Jun;35(6):827-32.
104. Russel S, Norvig P. Artificial Intelligence: A Modern Approach, 2003. EUA: Prentice Hall.
105. Saad A, Guilbert E, Gatinel D. Corneal enantiomorphism in normal and keratoconic eyes. Journal of Refractive Surgery. 2014;30(8):542-7.
106. Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics 2007;23(19):2507-2517.
107. Samuel AL. Some studies in machine learning using the game of checkers. IBM Journal of research and development. 1959;3(3):210-29.
108. Schiff GD, Volk LA, Volodarskaya M, Williams DH, Walsh L, Myers SG, et al. Screening for medication errors using an outlier detection system. Journal of the American Medical Informatics Association. 2017;24(2):281-7.
109. Segaran T. Programming collective intelligence: building smart web 2.0 applications. " O'Reilly Media, Inc."; 2007.
110. Sharif MS, Qahwaji R, Ipson S, Brahma A. Medical image classification based on artificial intelligence approaches: A practical study on normal and abnormal confocal corneal images. Applied Soft Computing. 2015;36:269-82.
111. Shazeer N, Mirhoseini A, Maziarz K, Davis A, Le Q, Hinton G, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538. 2017.
112. Silverman RH, Urs R, RoyChoudhury A, Archer TJ, Gobbe M, Reinstein DZ. Epithelial Remodeling as Basis for Machine-Based Identification of KeratoconusIdentifying Keratoconus Based on Epithelial Remodeling. Invest Ophthalmol Vis Sci. 2014;55(3):1580-7.
113. Smadja D, Touboul D, Cohen A, Doveh E, Santhiago MR, Mello GR, et al. Detection of subclinical keratoconus using an automated decision tree classification. Am J Ophthalmol. 2013;156(2):237,246. e1.
114. Smolek MK, Klyce SD. Current keratoconus detection methods compared with a neural network approach. Invest Ophthalmol Vis Sci. 1997;38(11):2290-9.
115. Sopharak A, Dailey MN, Uyyanonvara B, Barman S, Williamson T, Nwe KT, et al. Machine learning approach to automatic exudate detection in retinal images from diabetic patients. Journal of Modern optics. 2010;57(2):124-35.
116. Souza MB, Medeiros FW, Souza DB, Garcia R, Alves MR. Evaluation of machine learning classifiers in keratoconus detection from orbscan II examinations. Clinics. 2010;65(12):1223-8.
117. Stone M. Cross-validatory choice and assessment of statistical predictions. Journal of the royal statistical society.Series B (Methodological). 1974:111-47.
118. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; ; 2016.
119. Teknomo K. K-means clustering tutorial. Medicine 2006;100(4):3.
120. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society.Series B (Methodological). 1996:267-88.
121. Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. J Clin Epidemiol. 1996;49(11):1225-31.
122. Twa MD, Parthasarathy S, Roberts C, Mahmoud AM, Raasch TW, Bullimore MA. Automated decision tree classification of corneal shape. Optom Vis Sci. 2005 Dec;82(12):1038-46.
123. Usher D, Dumskyj M, Himaga M, Williamson TH, Nussey S, Boyce J. Automated detection of diabetic retinopathy in digital retinal images: a tool for diabetic retinopathy screening. Diabetic Med. 2004;21(1):84-90.
124. Valdés-Mas M, Martín-Guerrero JD, Rupérez MJ, Pastor F, Dualde C, Monserrat C, et al. A new approach based on Machine Learning for predicting corneal curvature (K1) and astigmatism in patients with keratoconus after intracorneal ring implantation. Comput Methods Programs Biomed. 2014;116(1):39-47.
125. Wettschereck D, Aha DW, Mohri T. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artif Intell Rev. 1997;11(1-5):273-314.
126. Williamson TH, Keating D. Telemedicine and computers in diabetic retinopathy screening. Br J Ophthalmol. 1998 Jan;82(1):5-6.
127. Witten IH, Frank E, Hall MA, Pal CJ. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann; 2016.
128. Xu Y, Gao X, Lin S, Wong DWK, Liu J, Xu D, et al. Automatic grading of nuclear cataracts from slit-lamp lens images using group sparsity regression. International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer; 2013.
129. Yousefi S, Balasubramanian M, Goldbaum MH, Medeiros FA, Zangwill LM, Weinreb RN, et al. Unsupervised Gaussian mixture-model with expectation maximization for detecting glaucomatous progression in standard automated perimetry visual fields. Transl Vis Sci Technol 2016;5(3):2-2.
130. Yu L, Liu H. Efficient feature selection via analysis of relevance and redundancy. Journal of machine learning research. 2004;5(Oct):1205-24.
131. Yun WL, Acharya UR, Venkatesh YV, Chee C, Min LC, Ng EYK. Identification of different stages of diabetic retinopathy using retinal optical images. Inf Sci. 2008;178(1):106-21.
132. Zhang S, Zhang C, Yang Q. Data preparation for data mining. Appl Artif Intell. 2003;17(5-6):375-81.

133. Zhang W, Hasegawa A, Itoh K, Ichioka Y. Image processing of human corneal endothelium based on a learning network. Appl Opt. 1991;30(29):4211-7.

134. Zheng C, Rashid N, Wu Y, Koblick R, Lin AT, Levy GD, et al. Using natural language processing and machine learning to identify gout flares from electronic clinical notes. Arthritis care & research. 2014;66(11):1740-8.