NTNU

Kunnskap for en bedre verden

Department of Computer Science

TDT4173 - Assignment 1

Supervised Learning

K-Nearest Neighbors

Group:
Group 10

Authors:
Kari L. Ness (kariln)
Tomas Samset (tomassam)
Sebastian M. Andresen (sebasman)

November 30, 2020

Abstract

The field of machine learning (ML) is one of the top trending technologies today, and is responsible for several technological advancements in various fields like medical image recognition, financial forecasting, and text mining. This paper explores the K-Nearest Neighbor (KNN) algorithm, a simple to implement and popular supervised learning algorithm. It provides an overview of the algorithms foundations, such as the different steps involved, how the algorithm's performance can be evaluated, its advantages and its limitations. With these foundations in mind, we compare the KNN algorithm to other popular supervised learning methods and their different properties and use cases. Determining the better model to choose from between these different supervised models for a given use case is shown to depend significantly on factors like the size of the training data, how many features each data point has, and how the data points are distributed. KNN is identified to be sensitive to data with high feature dimensionality, and potentially being computationally expensive. However, the KNN algorithm is still shown to perform competitively in many problem domains due to its versatility, simplicity, robustness to noisy data and good performance when the size of the data increases. The paper also identifies some current applications of the algorithm and points out potential areas for related future research within variations of KNN and machine learning in general.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

R^2	R-squared
1-NN	1-Nearest Neighbor
AI	Artificial intelligence
ANN	Artificial neural network
CAD	Computer-aided diagnostics
CRM	Customer Relationship Management
K	Number of neighboring samples used in each prediction
KNN	K-Nearest neighbor
ML	Machine Learning
NN	Neural network
RBF	Radial basis function
SVM	Support vector machine

1 Introduction

Machine learning is a subfield of artificial intelligence (AI) and enables a computer to improve its performance through experience. Currently, machine learning is one of the top trending technologies, but while machine learning gets increasing attention and interest in today's society, machine learning as a field is not new. Several scientists such as Thomas Ross and Alan Turing did substantial work on building machines that were able learn. In 1959, Arthur Samuel defined the term machine learning as a *"Field of study that gives computers the ability to learn without being explicitly programmed"*, [24]. Later on, Tom M. Mitchell defined learning as *"A computer program is said to* ***learn*** *from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*, [17].

One usually characterize machine learning methods based on the learning strategy. The main machine learning categories are broadly classified into supervised, unsupervised, and reinforcement learning, as shown in figure 1. In supervised learning, the model is trained on paired data, with both input and output, in order to predict future events. In unsupervised learning, the model is trained on unlabeled data with no guidance. In this way, the model is looking for hidden patterns in the given data. Reinforcement learning is based on a series of feedback/reward cycles, and learning is done by interacting with its environment [5].

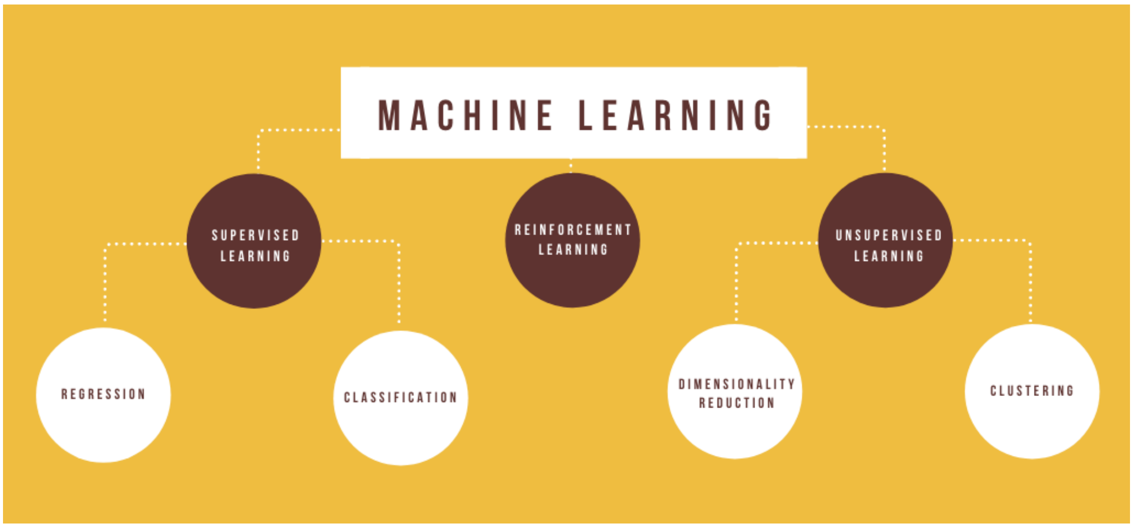


Figure 1: Machine learning categories

Supervised learning trains on classified data in order to create a mapping function that can predict unknown events. The labeled data in supervised learning is called a **training set**, and consists of several **training examples**. Each training example ($x^{(i)}$,$y^{(i)}$) consists of one or more **features** or input variables, $x^{(i)}$, and one **target** or output variable, $y^{(i)}$. Through training on the training set, one wishes to learn a function $h : X \mapsto Y$. h is commonly known as a **hypothesis**, and is regarded good if it is able to accurately predict future values of y, [21]. A process flow diagram of supervised learning can be seen in figure 2.

Supervised learning is generally classified into two categories; regression and classification. Regression and classification are powerful methods that enable the user to classify and process data using machine language, and are widely used in industries such as medicine and finance [28]. Regression is used to solve problems where we have continuous values, and classification is used to classify examples into a discrete set of possible categories [17]. This paper will explore one of the most widely used supervised learning algorithms, K-Nearest Neighbor.

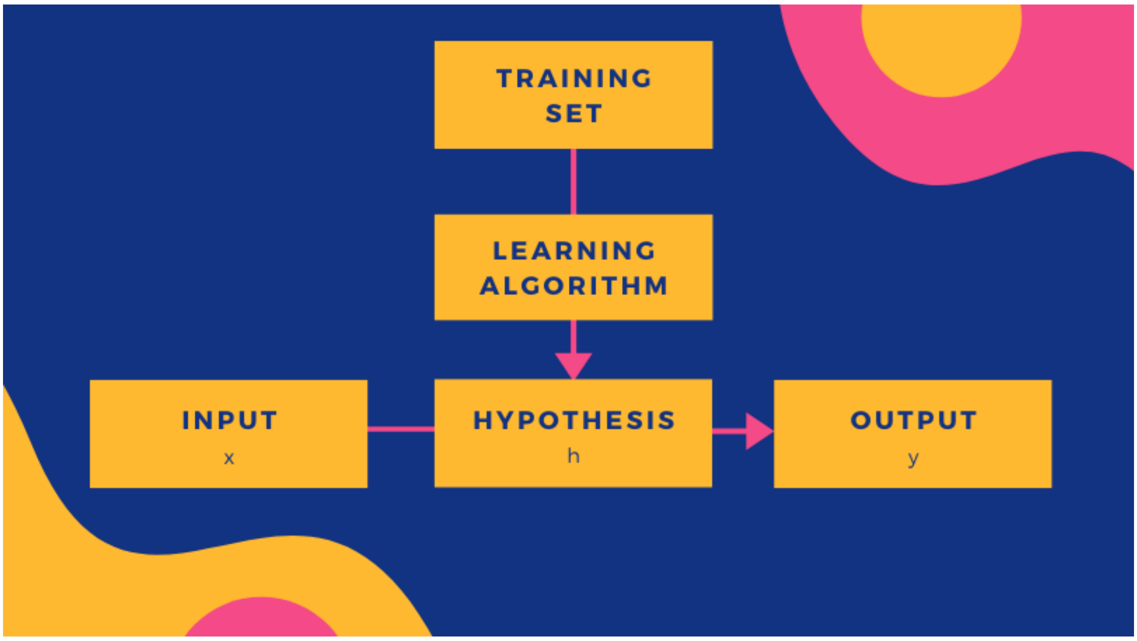


Figure 2: Supervised learning process

2 Foundations

2.1 Core characteristics

As mentioned in section 1, the KNN algorithm is a supervised machine learning algorithm that can solve both regression and classification problems [19]. KNN uses instance-based learning, or lazy learning, where the learning and generalization is delayed until prediction time, and not done preemptively on the training set [17].

The algorithm retains the entire training set while learning and assigns a class for each query based on the most occurring class of its K-nearest neighbors in the training data [12]. In its essence, KNN presumes that if the k-nearest instances to a query are similar in their known attributes, they are likely also similar in terms of their unknown attributes. Algorithm 1 provides a sketch of a simplified KNN classification algorithm. If used as a regression algorithm, the KNN algorithm will calculate the average of the values occurring in the K nearest neighbors instead of returning a categorical class label.

Algorithm 1 Simplified KNN Algorithm

```
Input: Training set samples $D$, Test sample $d$, $K$
Output: Predicted label of test sample
  for every sample $s$ in $D$ do
      compute distance between $s$ and $d$
  end for
  sort the distances in $D$ in increasing numerical order and select the first $K$ samples
  assign the class which occurs most frequently to $d$
```

To measure the similarity or the distance between instances, one would have to define a distance metric. One of the more popular metrics to use is the Euclidean distance, but other metrics like Euclidean squared, City-block, and Chebychev are also commonly used [1]. The choice of distance metric and K are essential factors to the algorithms fit and performance.

2.2 Training and optimization

In order for a learning model to be able to generalize and make predictions on unseen training data, it often makes certain assumptions based on the data, which is called the model's *inductive bias* [16]. For KNN, the model's inductive bias is the assumption that similar input points should lead to similar output points. For example, the prediction of an instance will be similar to other instances nearby in Euclidean distance [17].

Figure 3 shows the decision surface for a 1-Nearest Neighbor algorithm, 1-NN. 1-NN is KNN in its simplest form, where $K = 1$. In this form, unknown samples are classified by only using the single nearest sample. The figure shows a jagged decision surface, and shows a fitting with a K that is too low. When the K is too low, the algorithm will risk overfitting on the training set. Overfitting, in this context, refers to the model learning the training data too well. This means that the model is too closely fit to a limited set of data and will not find a general rule suitable for the prediction of unseen samples [6] [2].

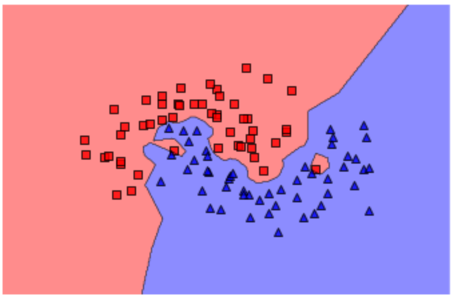
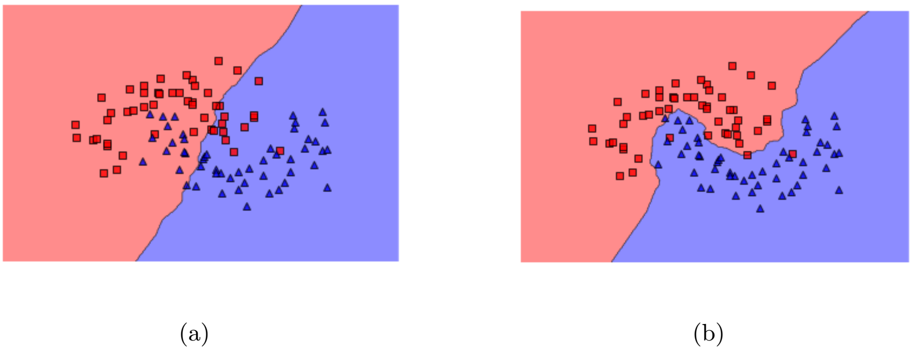


Figure 3: Decision surface induced by a 1-NN, a KNN implementation where K = 1

As K increases, more neighbors are averaged in each prediction, which will make the decision surface drawn for the algorithm becomes smoother. However, by choosing a K that is too high the algorithm risks underfitting on the training set. Underfitting happens when the model is incapable of capturing the variability of the data, and thus are not able to generalize well enough on neither the training data nor the test data [2]. Figure 4 (a) shows the KNN decision surface for the same data set as figure 3, with $K = 80$. In this case, the high K-value has caused the algorithm to underfit, just splitting the decision surface in half. In figure 4 (b), which shows the decision surface with $K = 5$, one can see a more balanced decision surface. The algorithm has generalized better and is even ignoring outliers.



(a)

(b)

Figure 4: Decision surface induced by a KNN algorithm with (a) K = 80 and (b) K = 5

2.3 Performance evaluation

Based on the predictions, one can create a confusion matrix of the correct and wrongly predicted labels per class. The confusion matrix is useful for measuring the precision and recall scores on a per-class basis. It shows how many predictions were right or wrong for each class, and will also allow us to see how the algorithm performs on specific classes. Table 1 shows a confusion matrix for a binary classification problem. The horizontal axis corresponds to predicted labels, and the vertical axis corresponds to the true labels.

	0	**1**
0	True Negative	False Positive
1	False Negative	True Positive

Table 1: Binary classification confusion matrix

The correct predictions are represented by *true negatives* and *true positives* in the confusion matrix. The optimal goal would be to have as many true negatives and true positives as possible. In many cases, one will have to consider the impact of a false positive or a true negative and whether those are important for the given use case. For example, when classifying diseases in medicine, a false positive disease prediction can lead to unnecessary medical treatment, and a false negative can lead to diseases being left untreated.

There are a variety of metrics available to evaluate the performance of a classification algorithm. Some measures commonly used for a KNN classifier are accuracy, precision, and recall. Accuracy is defined as the ratio of correctly predicted labels to the total number of predictions (as seen in equation 1). This metric is used to evaluate the overall performance of the classifier [1].

By analyzing the confusion matrix, we can compute the precision and recall metrics. The precision metric is the ratio of correctly predicted positive labels to the total predicted positive labels (as seen in equation 2). The recall is the ratio of correctly predicted positive labels to all actual correct predictions (as seen in equation 3). It is beneficial when these values to be as high as possible.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{1}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3}$$

Using the KNN algorithm on a regression problem, one has to use different evaluation metrics than for the classification. One commonly used measurement is R^2 (R-squared), which measures how well the algorithm predictions fit the data. R^2 is always between 0 and 1, where 0 corresponds to an algorithm that predicts the mean value of all training values, and 1 corresponds to a model that predicts perfectly.

2.4 Advantages and limitations

Two of the main advantages of using the KNN algorithm are its simplicity and effectiveness. As seen in algorithm 1, the algorithm is relatively simple to implement with its few steps, making the algorithm intuitive and straightforward to understand. It does not require a learning model before prediction, and it is non-parametric. Being non-parametric in this context implies that there are either no parameters or a fixed number of parameters regardless of the data's size. Instead of being learned, the parameters are decided based on the size of the training data [1]. This flexibility will likely yield more powerful models, but at the cost of more computational resources than parametric models. On the other hand, parametric models make stronger assumptions on the data, which gives them faster computation in return. The parametric model's performance, however, is highly reliant on these assumptions being correct.

In addition to this, the KNN, despite its simplicity, performs competitively in many problem domains. It is robust to noisy training data and has been proven still effective when the size of the training dataset increases in size [17].

However, the algorithm also has a few well known disadvantages. Being an instance-based learning method, the KNN algorithm can be computationally expensive. All the training data is stored in memory and the distance between each point needs to be calculated at prediction-time based on all their attributes [12] [17].

KNN is also sensitive to irrelevant features and multi-dimensional data, and the instances should not have more than 20 attributes to avoid *the curse of dimensionality* [17]. The curse of dimensionality is a problem that arises when we analyze data in higher-dimensional space; as the dimensions increase, the volume of the space grows so fast that the data quickly become too sparse to make any accurate predictions [1]. An example of this sensitivity is if an instance has 20 features, but only 2 of these features are actually relevant to determining the classification for this particular domain. The 18 other irrelevant features will significantly influence the distance measured between instance, and in return, cause misleading predictions.

3 Alternative Methods

In addition to the K-Nearest Neighbor algorithm, many other supervised machine learning algorithms can be used to solve a classification or regression problem. This section briefly introduces some of these methods and compares some of their properties to the KNN algorithm. These models all have their advantages and disadvantages and when to use which approach will often vary based on the amount of training data, the dimensionality of this data, and how the data points are distributed.

3.1 Linear regression

Linear regression is a popular machine learning algorithm for regression problems due to its simple implementation and quick training process. The algorithm attempts to model the linear relationship between an output variable and one or more input variables. This makes it a parametric algorithm, as opposed to KNN, which is non-parametric (see 2.4) [27]. Figure 5 illustrates an example decision surface induced by a linear regression model [9]

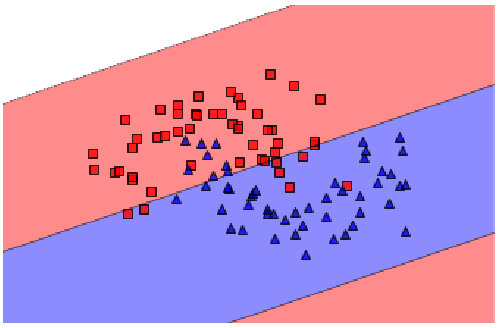


Figure 5: Linear regression decision surface [9]

While KNN is a *lazy* learning algorithm, linear regression is an *eager* algorithm, meaning that the model fits the training data preemptively before predicting new instances. This allows the linear regression model to be faster than KNN at prediction-time. Linear regression does require that the solution is linear, which might not always be the case in reality.

3.2 Logistic regression

Despite having the word ’regression’ in its name, logistic regression is a popular machine learning algorithm for classification problems. It uses a logistic function to output a probability between 0 and 1, which can be used to predict a class label [4]. Figure 6 illustrates an example decision surface induced by a logistic regression algorithm [9].

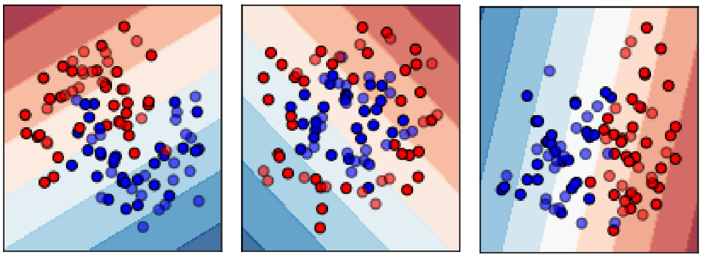


Figure 6: Logistic regression decision surface. [9]

Similar to linear regression, logistic regression is a parametric model that only supports linear solutions. While being faster than KNN, a logistic regression model can also output the confidence levels (probability) of the predictions it makes, where the KNN algorithm only outputs the predicted label [21]

3.3 Naive Bayes

Naive Bayes is a parametric classification model based on Bayes’ theorem. It applies the theorem with the *naive* assumption that all features are mutually independent, which is rarely the case in the real world [32]. Despite this unrealistic assumption, the Naive Bayes algorithm has shown to work effectively in some real world applications such as document classification and spam filtering [17]. Figure 7 shows the decision surface induced by a naive Bayes algorithm on three different datasets with varying data distribution [9].

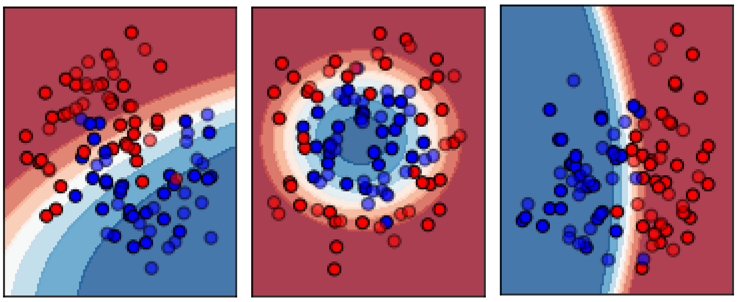


Figure 7: Naive bayes decision surface [9]

Compared to KNN, the Naive Bayes algorithm is shown to be much faster due to KNN’s required computation at prediction time. The assumption of mutually independent features makes the naive Bayes algorithm robust to irrelevant features. Unlike the KNN algorithm, it is not as affected by the curse of dimensionality when the number of feature dimensions increases [7].

3.4 Support Vector Machines

Support Vector Machines (SVM) are machine learning techniques that can be used for both classification and regression problems. It has variants to account for both linear and non-linear problem cases. The SVM's basis is to separate the labels by computing $N-1$ dimensional hyperplanes from N features and finding the optimal hyperplane based on the distance between the boundary nodes or support vectors. For non-linear problems, a kernel function is also used to ensure the linearly separable distribution of labels [4]. A kernel function is a function used to measure the similarity between two inputs [17]. Figure 8 illustrates the comparative decision surfaces in a linear SVM model compared and a SVM model using a radial basis function (RBF) kernel on the same dataset [9].

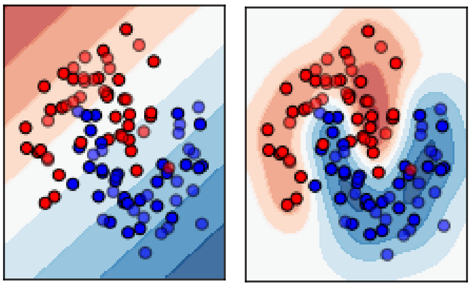


Figure 8: Linear SVM (left) and RBF SVM (right) decision surfaces [9]

Similarly to KNN, the SVMs, using the appropriate kernel functions, performs well on non-linear data. Where the KNN algorithm should not have more than 20 dimensions to avoid the curse of dimensionality, the SVM is effective in cases with a large number of features, such as in text classification where this is common. SVM is, however, not as suitable for large data sets as the KNN, and will underperform when the number of training data points is exceeded by the feature dimensionality of the points [30].

3.5 Decision trees

The decision tree is a tree-based machine learning algorithm that can solve both regression and classification problems. The way decision trees work is similar to how humans would sort some problems. Figure 9 shows a simplified example of this decision making process. It uses inverted trees, where each node represents a condition on a feature, and the conditional output decides the next node to navigate to. A prediction is given when the algorithm reaches a leaf node [15].

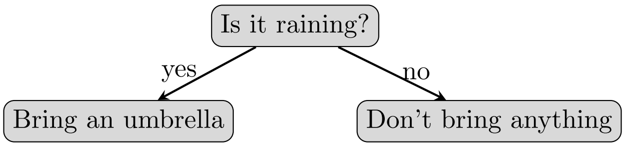


Figure 9: A simple decision tree on whether or not to bring an umbrella before going outside

Just like the KNN, decision trees are non-parametric algorithms. When comparing the performance of decision trees to KNN, the decision trees are faster due to the lazy learning approach KNN uses. Some of the main advantages of using decision trees are that their predictions are understandable and explainable, and that decision trees, like the KNN algorithm, work well with non-linear data distribution [30]. Some notable limitations are that the model can easily overfit if the tree is built too deep, and that the tree can grow complex on complicated data, requiring more computing power to train [20]. The tree structure also makes the model unstable, as a small change in the data can lead to a large change in the tree structure. Figure 10 illustrates the decision surfaces induced by a decision tree algorithm on three different datasets with varying distribution of data [9].

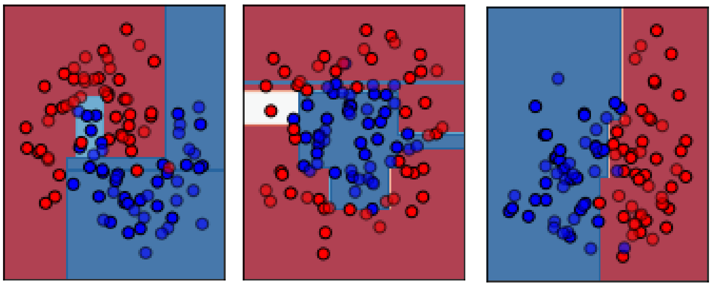


Figure 10: Decision tree induced decision surfaces [9]

3.6 Artificial neural networks and deep neural networks

Artificial neural networks (ANN), or just neural networks (NN), is a machine learning approach inspired by the neural networks that exists in the human brain [17]. It is based on having a layer of input nodes (neurons) that transmit signals to layers of additional nodes, multiplying the input values with a weight, and passing it through a non-linear activation function to the next set of nodes. A *deep neural network* is created by adding multiple neuron layers to this structure [25]. Figure 11 illustrates the decision surfaces induced by a neural network model on three different example data sets [9].

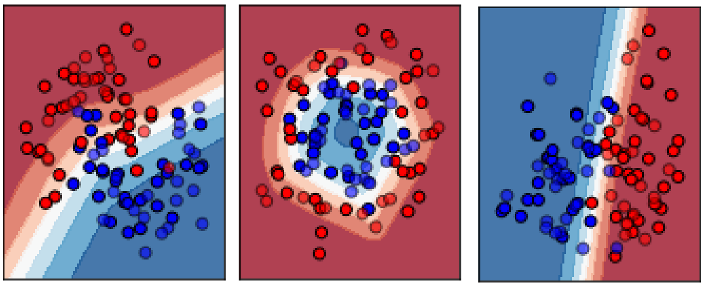


Figure 11: Neural net induced decision surfaces [9]

Neural networks have shown to be able to learn and solve complex problems, but they require a larger training set in order to achieve satisfactory performance. When solving some simpler problem cases where the training data is limited, the KNN algorithm can perform better than a neural network [20]. In addition, the neural networks are theoretically complex and might require a lot of hyperparameter tuning in comparison to the KNN.

4 Current Applications

KNN was first introduced as a novel approach to non-parametric pattern classification by Fix and Hodges in 1951 [12]. Since the method was first introduced, formal properties has been established, refinements has been made and the algorithm has been implemented in a wide range of ares. As mentioned in section 2, KNN can be used for both regression and classification problems, and has a wide range of applications in both of the ML types. Some of the recent applications of KNN will be mentioned in this section.

4.1 Text mining

Text mining was defined by Hearst in 1999 as the process of discovering unknown information from unstructured textual data. The process of text mining is often classified into three tasks; information retrieval (collection data), information extraction (extracting relevant information from the collected data), and data mining (discovery of patterns and structures in the extracted information) [3]. Text mining has been increasingly relevant due to the rapid digitalization of the past decades. Digitalization and the use of the internet have led to an information explosion, where the need for categorization and organization has been vital to reap the benefits of the new technology. Some relevant text mining techniques are text classification, text segmentation, and word classification. Due to its simplicity, KNN is the most widely used algorithm as a classifier of text and is therefore widely researched and studied [13].

One of the cases where text classification by KNN has proved to be valuable is through the classification of e-mails. By identifying features in the e-mails, one can allocate incoming e-mails in predefined classes with high accuracy [22]. Customer Relationship Management (CRM) is one of the fields where e-mail classification has proven valuable. CRM is a technology that aids enterprises with the management of customer relationships in an organized way. By classifying e-mails, one organizes the information and communication with customers, thus automating or easing the interaction between the CRM operators and their customers [14].

4.2 Image recognition

Image recognition is a collective term used for computer technologies that are able to recognize objects or targeted subjects in images or videos[8]. For example, as seen in figure 12, if one input an image of a cat, an image recognition algorithm F(X) is able to classify the image motif as a cat and not a dog. This is a task that humans do easily, but that has proved to be a complex task for computers.

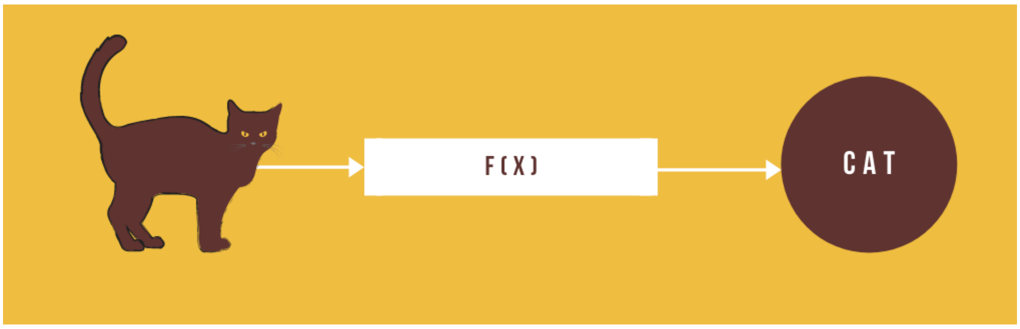


Figure 12: Image recognition

Most machine learning algorithms aim to automate processes to avoid human mistakes and unnecessary human interference. A bottleneck in several digitalization processes has been the categorization of handwritten material. KNN has been implemented to interpret human handwriting by calculating the distance between feature vectors to classify the unknown data points. As well as being the simplest image recognition algorithm to implement, KNN has shown high accuracy and efficiency in the categorization of handwritten digits and letters. Examples of tasks that have been automated through handwritten recognition are the reading of postal addresses in the postal service, reading bank checks, and organizing material in digital libraries [11].

Medicine is another field that has seen rapid growth in the utilization of image recognition by machine learning. One of the buzz-words within medicine today is computer-aided diagnostics (CAD), a new research field that apply image recognition. CAD aids medical professionals with tasks that usually involve visual inspection or manual localization to determine a clinical diagnosis. By doing this, machine learning can provide a second opinion to the medical professionals and reduce the risk of human error [26]. KNN has proved to be a useful algorithm in CAD, and has for example shown high accuracy in the detection of breast cancer. Breast cancer is abnormal tissue growth in breasts and has a survival rate significantly influenced by early detection and treatment. Detection often happens through mammography, an image of the anatomical structure of the breast. KNN has proved to be an efficient and helpful aid for the medical professionals in the field [23].

4.3 Financial forecasting

Forecasting economy is challenging due to complex interactions from multiple variables. However, due to the quantifiable nature of finance, and the vast amount of historical data, machine learning has been implemented throughout the field. The results have been promising and are often even seen to outperform human experts in the field[29].

One of the most important financial tasks of KNN is stock market forecasting. The forecasting predicts future stock prices based on numerous factors such as market patterns, company performance, demographics, currency exchange rates, inflation, and credit ratings. All these factors are numerically quantifiable and possible to analyze using KNN. Analyses like these help exposing market trends that are useful when creating an investment plan [29].

4.4 COVID-19

During the global COVID-19 pandemic caused by the SARS-CoV-2 virus, the global community has been hit by an extensive health care crisis. Several countries faced the growing numbers of infected patients without being dimensioned for a crisis of such scale. It has been observed that the countries that several of the communities that has managed to control the virus outbreak has a emergency response characterized by early and centralized response. These responses include infection detection programmes, symptom mapping and fast diagnosis. Due to the nature of the pandemic, the the challenges involved a vast amount of information that had to be analyzed rapidly to avoid exponential growth COVID-19 cases[33].

The application of machine learning in order to ease the workload of the medical professionals has therefore gained increased attention. By implementing KNN on data of COVID-19 patients, computer scientists have for example worked on the prediction of health risk of patients and the prediction of illness based on symptoms [18][10]. If machine learning is implemented correctly, it can hopefully be a tool to distribute health services such as emergency aid and COVID-19 to the people who need it the most. As an aid in organizing patient information and automating processes, machine learning might relieve stress on overloaded health care systems around the globe.

5 Conclusions and Further Work

Throughout this paper, the implementations and applications of KNN have been investigated. Due to the simplicity of the algorithm, it is both easy to understand and implement on a wide range of machine learning challenges, both within classification and regression. In addition to its simplicity, KNN is robust to noisy training data and performs effectively even when the size of the data set increases. However, it is also computationally expensive and has a limited feature dimensionality due to the real-time nature of the algorithm. Thus, when KNN is compared to alternative methods, other methods can often be seen to be faster, but not always as accurate as KNN.

Despite its simplicity, it is seen that KNN performs competitively in several problem domains, and it is therefore not surprising that KNN is one of the most used supervised learning methods. The method is implemented as a data analysis tool in a wide range of fields, with the healthcare sector as one of the more noteworthy. Machine learning and KNN has contributed to faster and more precise medical diagnostics by automating processes and being a second opinion for medical professionals.

KNN is an old algorithm that has been implemented and researched significantly over the past years. In order to adapt KNN to its versatile use, one has seen several improvements and specializations of the standard KNN algorithm. Some of the changes done to the algorithms have proved to run efficiently, with higher accuracy and speed than the standard KNN [31]. In addition, research done on combining KNN with other machine learning techniques such as neural network has had promising results. Doing research on the different variations of KNN and exploring how this new knowledge can be implemented with other machine learning techniques would be an interesting topic for further work.

KNN and machine learning is projected to have a positive impact on several fields. With more research done on improvements to the algorithm and increasing computing power, one can expect the algorithm to be able to solve increasingly complex challenges.

Bibliography

[1] Haneen Abu Alfeilat, Ahmad Hassanat, Omar Lasassmeh, Ahmad Tarawneh, Mahmoud Alhasanat, Hamzeh Eyal-Salman, and Surya Prasath. Effects of distance measure choice on k-nearest neighbor classifier performance: A review. *Big Data*, 7, 08 2019.

[2] Haider Allamy. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). 12 2014.

[3] McNaught J. Ananiadou S. *Text mining for biology and biomedicine.* Artech House, 2006.

[4] Martin Bichler and Christine Kiss. A comparison of logistic regression, k-nearest neighbor, and decision tree induction for campaign management. In *AMCIS 2004*, page 230, 01 2004.

[5] Mashrur Chowdhury, Amy Apon, and Kakan Dey. *Data Analyticsfor Intelligent Transportation Systems.* Elsevier, 2017.

[6] Tom Dietterich. Overfitting and undercomputing in machine learning. *Computing Surveys*, 27:326–327, 1995.

[7] Jerome Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data Min. Knowl. Discov.*, 1:55–77, 03 1997.

[8] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing.* Prentice Hall, Upper Saddle River, N.J., 2008.

[9] Jaques Grobler. Comparing classifiers, 2020.

[10] Ahmed Hamed, Ahmed Sobhy, and Hamed Nassar. Accurate classification of covid-19 based on incomplete heterogeneous data using a knn variant algorithm. *Research Square*, 2020.

[11] Norhidayu Abdul Hamid and N. N. A. Sjarif. Handwritten recognition using svm, knn and neural network. *ArXiv*, abs/1702.00723, 2017.

[12] Sadegh Bafandeh Imandoust and Mohammad Bolandraftar. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *S B Imandoust et al. Int. Journal of Engineering Research and Applications*, 3(5):605–609, 2013.

[13] Rajni Jindal and Shweta Taneja. A novel weighted classification approach using linguistic text mining. *International Journal of Computer Applications*, 180(2), 2017.

[14] Gutha Jaya Krishnaab and Vadlamani Ravi. Evolutionary computing applied to customer relationship management: A survey. *Science Direct*, 56:30–59, 2016.

[15] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook, 2nd ed.* 01 2010.

[16] Tom Mitchell. The need for biases in learning generalizations. 10 2002.

[17] Tom M. Mitchell. *Machine Learning.* McGraw-Hill Education, 1997.

[18] Mahdi Shakibi Mohammad Pourhomayoun. Predicting mortality risk in patients with covid-19 using artificial intelligence to help medical decision-making. *medRxiv*, 2020.

[19] Mohssen Mohammed, Muhammad Badruddin Khan, and Eihab Bashier Mohammed Bashier. *Machine learning: algorithms and applications.* CRC Press, Taylor & Francis Group, Boca Raton, 2017.

[20] Mahfuzah Mustafa, Mohd Nasir Taib, Zunairah Murat, and Norizam Sulaiman. Comparison between knn and ann classification in brain balancing application via spectrogram image. *Journal of Computer Science & Computational Mathematics*, 2:17–22, 04 2012.

[21] Andrew Ng. Cs229 lecture notes.

[22] A.Kousar Nikhath, K.Subrahmanyam, and R.Vasavi3. Building a k-nearest neighbor classifier for text categorization. *(IJCSIT) International Journal of Computer Science and Information Technologies*, 7(1):254–256, 2016.

[23] R. A. Nurtanto Diaz, N. Nyoman Tria Swandewi, and K. D. Pradnyani Novianti. Malignancy determination breast cancer based on mammogram image with k-nearest neighbor. In *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, volume 1, pages 233–237, 2019.

[24] A.L. Samuel. Some studies in machine learning using the game of checker. (English). *IBM J. Res. Develop.*, 3(3):210–229, 1959.

[25] Md Sarker, Sanjida Noor, and Uzzal Acharjee. Basic application and study of artificial neural networks. *SK International Journal of Multidisciplinary Research Hub*, 4:1–12, 05 2017.

[26] Li Q Shiraishi J, Appelbaum D, and Doi K. Computer-aided diagnosis and artificial intelligence in clinical imaging. *Semin Nucl Med.*, 41(6):449–462, 2011.

[27] Rebecca C. Steorts. Sta325 lecture notes.

[28] M. Talabis, R. McPherson, I. Miyamoto, J. Martin, and D. Kaye. *Information Security Analytics*. Elsevier, 2014.

[29] K. Taunk, S. De, S. Verma, and A. Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260, 2019.

[30] Aydın Ulaş, Olcay Taner Yıldız, and Ethem Alpaydın. Cost-conscious comparison of supervised learning algorithms over multiple data sets. *Pattern Recognition*, 45(4):1772 – 1781, 2012.

[31] Zacharias Voulgaris and George Magoulas. Extensions of the k nearest neighbour methods for classification problems. *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, AIA 2008*, 02 2008.

[32] Harry Zhang. The optimality of naive bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA*, volume 2, pages 562–567, 01 2004.

[33] Huachun Zou, Yuelong Shu, and Tiejian Feng. How shenzhen, china avoided widespread community transmission: A potential model for successful prevention and control of covid-19. *Infectious Diseases of Poverty*, 9, 12 2020.

Appendix