# Comparative Study of Four Supervised Machine Learning Techniques for Classification

**Amr E. Mohamed**
Department of Mathematical Sciences
University of Essex, UK.

## Abstract

*A comparative study of four well-known supervised machine learning techniques namely; Decision Tree, K-Nearest-Neighbor, Artificial-Neural-Network and Support Vector Machine has been conducted. This paper concentrated on the key ideas of each technique and its advantages and disadvantages. Practical application has been conducted at the end of the study to compare their performance. Some measures have been used for evaluating their performance, such as sensitivity and specificity. This study had shown that there is no one measure can provide everything about the classifier performance and there is no such classifier that can satisfy all the criterion.*

***Keywords***: Supervised Learning, Classification, Decision Tree, KNN, Artificial Neural Network, Support Vector Machine.

## *1 Introduction*

In the most recent years, the amount of information that we can extract from the data has rapidly increased. Machine learning is not just about storing large amounts of data, but it is part of Artificial Intelligence (AI). Artificial Intelligence is the improvement of the computer programs to perform tasks that usually require the human intervention, such as decision making. Making the right decision for a specific problem is the main factor for achieving our goals. For this purpose, many machine learning techniques are used for both classification and regression problems. Classification is used when the prediction goal is a discrete value or a class label. When the prediction goal is continuous, regression is the appropriate method to use.

## *2 Machine learning*

The essence of machine learning is to compile the data we observe with the experience that the program learns to generate the information that we can make use of. For example, the process of differentiating the valid emails from the spam emails. The input will be some documents or words which included in the emails and the output should be yes or no that indicating the email is spam or not-spam respectively, but we do not have an algorithm to accurately identify the spam emails. Machine learning provides a solution for this task which we provide examples of the emails that we manually labelled with spam or valid and the program can automatically learn to distinguish between them (Alpaydin, 2014).

### 2.1 Supervised learning

The main idea of supervised learning is to learn a mapping between the input and the output whose correct values provided by a supervisor. There are two main types of supervised learning, classification and regression, where there is an input $\boldsymbol{x}$and output $y$, and the main role is to find a mapping between the input and the output. In classification, the task is to assign the training input to one of the predefined classes. In the simple case when we have two classes in the spam email example, the predefined classes are (1 or 0) indicating the type of the email (spam or not spam), and the role of the algorithm is to classify the training examples to one of the two classes. The good learner is the one that can discriminate the two classes perfectly if there are no different data points have the same label and there are no identical points have different labels. The classification problem can be formulated as follows; given a sample of training data$(\boldsymbol{x}_i, y_i)_{i=1}^{N}$ that is drawn from a population according to unknown probability distribution $P(\boldsymbol{x}, y)$and a loss function $L(y, f(\boldsymbol{x}))$which estimates the error for a given $\boldsymbol{x}$, $f(\boldsymbol{x})$is predicted instead of the actual value of $y$.

We need the loss function to penalize errors in prediction; it means that the closer the predicted value $f(\boldsymbol{x})$to the actual value of $y$, the lower the loss functions. The aim is to find a function that can minimize the error on the unseen data. The ability of an algorithm to classify the unseen data correctly is known as it's generalization. In classification, what we want to learn is a class and the loss function can take the following form: $\mathrm{L}(y, f(\boldsymbol{x})) = \sum I\ (y \neq f(\boldsymbol{x}))$, where misclassification is assumed and $f(\boldsymbol{x})$is the predicted class label. $I$ is the indicator function and it can be defined as $0/1$, where $0$indicates that the predicted class and the actual class are the same, and $1$indicates that the classes are different (Alpaydin, 2014)(Jakkula, 2006).

## 3 Support Vector Machine

Support Vector Machine (SVM) is one of the most powerful training techniques for supervised learning. Support vector machine was first introduced by Vapnik in 1992 (Boser, Guyon, &Vapnik, 1992). It was used for many applications for classification, regression and feature selection. In classification, support vector machine determines an optimal separating hyperplane using the concept of margin which is the essence of the SVM. The margin is the distance between the hyperplane and the closest points to it on either side, which we want to maximize for better generalization. There is a tradeoff between maximizing the margin and minimizing the number of the misclassified examples. There are some bounds that govern the relation between the model performance and its capacity. This can be used to balance the trade-off between the model bias and the model variance. In the following sections, the formulation of the support vector machine for classification will be introduced for both cases, when the data is strictly linearly separable and when the data is not fully linearly separable (Burbidge & Buxton, 2001) (Dietterich & Kong, 1995).

### 3.1 Linearly Separable Classification

Suppose we have N training examples that strictly linearly separable, where each point $\boldsymbol{x_i}$has $K$attributes and belongs to one of the two classes $T$ and $F$, which take the following values $+1$and $-1$respectively. The training data takes the following form: $(\boldsymbol{x_i}, y_i)$, where $i = 1, \dots, N$, $y_i \in \{+1, -1\}, \boldsymbol{x} \in R^K$. It means that $y_i = +1$if $\boldsymbol{x_i}\in$class $T$, and$y_i = -1$if $\boldsymbol{x_i}\in$class $F$. For the data to be linearly separable, we can draw a hyperplane that separates the two classes, this hyperplane can be described by:

$$\boldsymbol{w^T x} + b = 0 \qquad (1)$$

where $\boldsymbol{w}$is a normal vector that is perpendicular to the plane and $b$is the bias which determines the point location relative to the origin. In this case, we can separate the two classes by a pair of two parallel bounding planes:

$$\boldsymbol{w^T x} + b \geq +1 \qquad for\, \boldsymbol{x} \in T \qquad (2)$$

$$\boldsymbol{w^T x} + b \leq -1 \qquad for\, \boldsymbol{x} \in F \qquad (3)$$

The decision rule given by $f(x) = sign\ (\boldsymbol{w^T x} + b)$. These two boundary constraints can be combined into:

$$y_i(\boldsymbol{w^T x} + b) \geq 1\ \forall i \qquad (4)$$

The data points that lay on the bounding planes $\boldsymbol{w^T x} + b = \pm 1$are called support vectors.

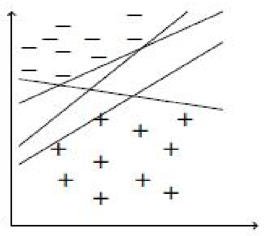

Figure 1: Many hyperplanes which can be fit to classify the data.

In the illustration of figure 1, there are many hyperplanes that can be drawn to separate and classify the training data. However, there is only one can classify the training examples correctly by achieving the maximum separation with the largest margin. The aim of the support vector machine algorithm is not only to classify the points correctly, but also want them some distance away for better generalization. This means that the points that are not support vectors carry no information.

Support Vector Machine searches for a separating hyperplane by maximizing $\frac{1}{\|w\|}$ and it is equivalent to minimizing $\frac{1}{2}\|w\|^2$. It can lead to a simple Quadratic Programming (QP) optimization problem (Jakkula, 2006)(Lee, Yeh, &Pao, n.d.):

$$min\frac{1}{2}\|w\|^2 \quad st. \quad y_i(\boldsymbol{w^T x} + b) \geq 1 \ \forall i \tag{5}$$

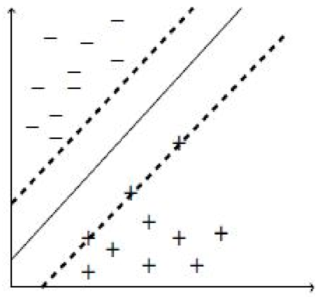

Figure 2: Margin boundary and support vector points.

## 3.2 Data that is not fully linearly separable

In the real world, many datasets can't be separated linearly and it might have a curved decision boundary that separate the data. SVM can be extended to handle data that is not fully separable or the classes are linearly inseparable. To handle such cases, we would like the SVM to allow a few examples to fall on the wrong side of the separating hyperplane. The SVM technique can be modified to do that by adding a "soft margin" that allows some points to be on the wrong side without affecting the result or violating the constraints. In this soft margin, the data points on the incorrect side of the hyperplane have a penalty and it is varying according to its position and how far this point from the margin boundary. The soft margin can be determined by introducing non-negative slack variables $\xi_i$ that take account of the fact that some data points may be misclassified due to noise. The slack variable measures the amount of violation of the boundary constraints.

$$w^T\boldsymbol{x} + b + \xi_i \geq +1 \qquad for\ \boldsymbol{x} \in T \tag{6}$$

$$w^T\boldsymbol{x} + b - \xi_i \leq -1 \qquad for\ \boldsymbol{x} \in F \tag{7}$$

These two equations can be combined into:

$$y_i(w^{\boldsymbol{T}}\boldsymbol{x} + b) + \xi_i \geq 1, \quad \xi_i \geq 0 \ \forall i \tag{8}$$

The aim is to reduce the incorrectly classified data points to the minimum and to classify the data points correctly with sufficient distance from the margin. This objective function can be formulated as follows:

$$\min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \quad st. \quad y_i(w^{\boldsymbol{T}}\boldsymbol{x} + b) + \xi i \geq 1 \tag{9}$$

Where C is a positive parameter which controls the trade-off between the slack variable penalty and the size of the margin. If C is too small, it allows more data to lie on the wrong side and it may underfit the training data. If C is too large, then the SVM algorithm may overfit the data which leads to poor generalization (Burbidge & Buxton, 2001)(Lee et al., n.d.).

### 3.2.1 Kernel Trick

Many datasets can't be separated and far from linear, but could be linearly separated by mapped into a higher dimensional space by using a nonlinear mapping. For this purpose, kernels are used to nonlinearly map the training data from the input space $R^n$ to a higher dimensional feature space $\mathcal{F}$ by a nonlinear function $\Phi$. The feature space refers to the collections of features that are used to characterize the training data. Then the training data $\boldsymbol{x}$ in the input space $R^n$ will become $\Phi(x) \in R^l$ in the feature space where $l$ is the dimensionality of the feature space.

This mapping is defined by the kernel $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \Phi(\boldsymbol{x_i})^T \Phi(\boldsymbol{x_j})$and the kernel is obtained by the dot product of the training data vectors. This means that the separating hyperplane will be linear in the feature space F, but nonlinear in the input space $R^n$. Kernels are useful if the functions can be recast into a higher dimensional space by some nonlinear mapping because we only need the inner product of the mapped inputs in the feature space to be determined. The kernel trick allows SVM to form nonlinear boundaries. There are a lot of kernel functions can be used, the appropriate kernel function is determined according to the nature of the case and it is picked by trial and error on the test set (Jakkula, 2006) (Leeet al., n.d.).

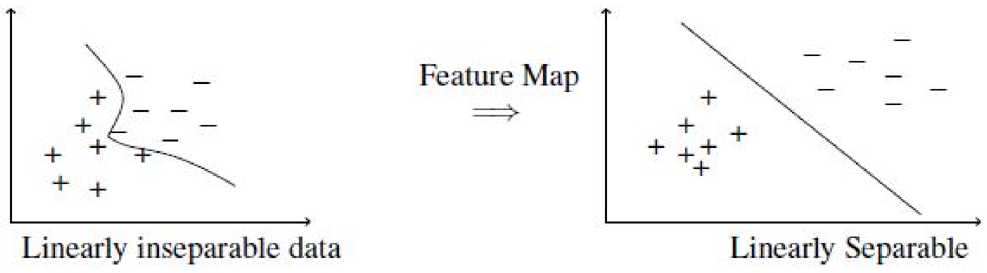

Figure 3: Illustration of the nonlinear SVM concept

### 3.3 Advantages and Disadvantages of Support Vector Machine

The major advantage of the Support Vector Machine is that the training is relatively easy. It scales relatively well to the high dimensional data. The trade-off between the model complexity and the error can be controlled easily. It can deal with both continuous and categorical data. It captures the nonlinear relationships in the data. No assumptions are required regarding to the data structure because it is a non-parametric technique. The prediction accuracy is very high and it provides a good generalization performance. It delivers a unique solution because the optimization problem is convex, which means that it has a unique minimum value. It is robust and able to deal with data that contains errors. One of the major disadvantages of the Support Vector Machine is the difficulty to interpret unless the features are interpretable. It can be computationally expensive and it needs a good kernel function. Its lack of transparency in results because it is a non-parametric method (Jakkula, 2006) (Auria & Moro, 2008).

## *4 Artificial Neural Network*

The artificial neural network is a mathematical model that tries to simulate the functionality of the biological nervous system. This mathematical model has three basic rules: multiplication, summation and activation. They basically involve the inputs that are weighted, which means that every input value is multiplied by a specific weight. Then all the weighted inputs will be added with a bias term. At the end, the sum of all the weighted inputs and the bias term will be transformed by an activation function to compute the output. The weights that are associated with each input provide the synapse strength. The higher the weight which is associated with a specific input, the stronger the input. These weights can be positive or negative, when the weight is positive ($w_i > 0$), it indicates an excitatory connection, while a negative weight inhibits the neuron activity. The basic processing element is called the perceptron. The perceptron has inputs $x_i$that may come from the environment (External input) or may be the outcome of other perceptrons. The output of this perceptron can be derived by:

$$y = \sum_{i=1}^{N} w_i x_i + b \qquad (10)$$

Where $b$is the bias term and it is also called the neuron's threshold, where it may be considered as an additional input, it is always unity and its weight is equal to $b$. In this case the perceptron output can be written as a dot product: $y = \boldsymbol{w}^T \boldsymbol{x}$ (11)

Where $\boldsymbol{w}$and $\boldsymbol{x}$are two vectors. The activation function or the transfer function defines the properties of the artificial neuron and it could be any activation function $\phi(.)$and the output will be as the following form:

$$y = \phi(\sum_{i=1}^{n} w_i x_i + b) \qquad (12)$$

The activation function can be determined depending on the problem that the artificial neuron needs to solve. It acts as a transformation entity so that the output of a neuron takes a value between certain range such as $[0,1]$or $[-1,1]$ according to the chosen function. The most popular activation functions are:Threshold function: It is also called a step function which has only two possible outcomes (zero or one), it takes zero if the summation of the input is less than a specific threshold and it takes 1 if the summation of the input is greater than or equal to that specific threshold. It takes the following form (Krenker, Kos, & Bešter, 2011) (Haykin, Haykin, Haykin, & Haykin, 2009) :

$$y = \begin{cases} 1, & \text{if} w^T x \geq b \\ 0, & \text{if} w^T x < b \end{cases} \qquad (13)$$

Sigmoid function is the most commonly used when we use nonlinear function. The sigmoid function is defined to be a strictly increasing function that exhibits balance between the linear and the nonlinear case whose graph is $S$shaped. The sigmoid function takes the following form (Krenker et al., 2011), (Haykin et al., 2009):

$$y = sigmoid\ (\boldsymbol{w^T x}) = \frac{1}{1 + exp(-\boldsymbol{w^T x})} \qquad (14)$$

The sigmoid function takes values between the range zero and one, but in some models, it is beneficial to use this interval $[-1, 1]$. In the latter case, the threshold can be defined as:

$$y = \begin{cases} 1, & \text{if} w^T x > b \\ 0, & \text{if} w^T x = b \\ -1, & \text{if} w^T x < b \end{cases} \qquad (15)$$

So, the whole mathematical framework for how the information flows from the input to yield the output can be represented as figure 4 shows.

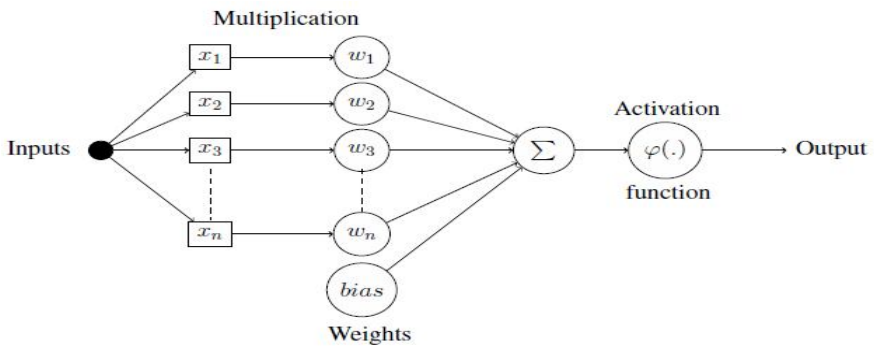

Figure 4: Mathematical framework of the Artificial Neural Network

### 4.1 Strengths and weakness of Artificial Neural Network

The strengths and weakness of the artificial neural network can be summarized in the following points (Basheer & Hajmeer, 2000):

1. **Strengths:**

- It can be used to solve linear and nonlinear programming problems.
- No prior knowledge of the process that generating the data is required for the artificial neural network to be applied.
- The ability to learn from provided examples makes them powerful and flexible; it means that the neural network learns, it does not have to be re-programmed.
- It has been successful for solving many classification, clustering and regression problems.

2. **Weakness:**

- It should never be viewed as a panacea to all the real-world problem, because other techniques are powerful in their own.
- The success of the model depends on the quantity of the data.
- Lack of clear guidelines for which artificial network architecture is optimal because the process is involving trial and error.

## *5 Decision Tree*

Decision tree is one of the most widely used classifiers in statistics and machine learning. Decision tree is a hierarchical design that implements the divide-and-conquer approach. It is a nonparametric technique used for both classification and regression. It can be directly converted to a set of simple ***if-then*** rules. It's straightforward representation makes the reader able to interpret the result and easy to understand. This section presents the basic features of the decision tree method for classification (Alpaydin, 2014)(Mitchell,1997)(Myles, Feudale, Liu, Woody, & Brown, 2004).

### 5.1 The graphical representation of the Decision Tree

In 1986, an algorithm for inducing decision trees ID3 was introduced by Quinlan (Quinlan, 1986). The ID3 algorithm was improved in 1993 and upgraded to C4.5 (Salzberg, 1994). Decision tree is a top-down greedy algorithm. It consists of a recursive splits in simple steps. It is made up of internal nodes and terminal leaves. Typically, the input data are represented as attribute-value. At each node, specific test function is implemented to decide the branch or the leaf for each instance, which means that the decision of the class that the instance belongs to. This process starts from the root of the tree, apply the test function of this specific node and moving down to the tree branch that corresponds to the value of the attribute in the given instance. This procedure is performed at each node until a leaf is encountered, then the predicted class of the given example will be its label (Alpaydin, 2014) (Mitchell, 1997)(Podgorelec, Kokol, Stiglic, & Rozman, 2002). The following graph illustrates the process for predicting a new job applicant according to his/her educational level and their years of experience.

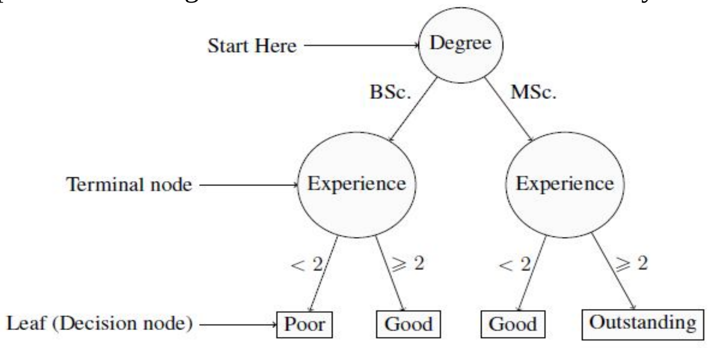

Figure 5: Typical representation of a decision tree

One of the rules that can be extracted from the graph above is, IF the applicant has a bachelor degree and has less than two years of experience, THEN his/her performance will be classified as poor. The graph shows that how decision tree works with the binary classes, but it can be extended to deal with multi-class cases. It is a greedy algorithm because the best attribute is determined at each node from the root to the last terminal node. The idea is how to determine the best attribute that splits the data efficiently at each stage starting from the root. We wish to pick the attribute that is most useful for classifying the data, which means that it gives the maximum degree of discrimination. The information gain is a statistical technique which measures how well the attribute splits the data. Before defining the information gain criterion (Quinlan, 2014), let us start by introducing the entropy. Entropy as it is defined in the information theory, is the minimum number of bits that is needed to encode the class of an instance. It is also called the impurity measure. Assume that we have a collection of examples S, and $S_m$is the number of examples that reach node $m$, $S_m^j$is the number of examples that belongs to class$j$in node $m$with$\sum_{j=1}^{k} S_m^j = S_m$. The probability of class $j$, given an instance $\boldsymbol{x}$that reaches node $m$is

$$P(c_j|\boldsymbol{x}, m) = P_m^j = \frac{S_m^j}{S_m} \qquad (16)$$

Then the Entropy can be computed as:

$$Entropy\ (S) = -\sum_{j=1}^{k} P_m^j log_2 P_m^j \qquad (17)$$

The entropy function can be represented as follows:

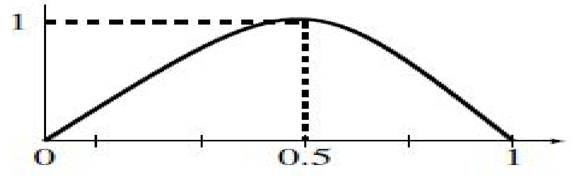

Figure 13: Entropy function

In the illustration of figure 6, if all the members of the training examples belong to the same class, so one of the class probabilities is equal to 1 and the other is equal to 0, in this case the entropy will be equal to zero. On the other hand, if the instances be split equally to the two classes, it means that either of the class probabilities are equal to 0.5 and the entropy is equal to 1. The smaller the entropy, the more informative the attribute (Alpaydin, 2014; Mitchell, 1997).Given the entropy, the effectiveness of a specific attribute can be measured by the information gain criterion as follows:

$$Gain(S,A) = Entropy\ (S) - \sum_{v\, \in values(A)} \frac{|S_v|}{|S|}\ Entropy\ (S_v) \qquad (18)$$

Where $values\ (A)$ indicates all the possible values that attribute $A$can take, and $S_v$is the subset of the whole collection sample $S$for which attribute $A$has value $v$. The first term in the equation is the entire entropy before partitioning the dataset and the second term of the equation is the entropy after splitting the instances using attribute $A$. This means that the information gain $Gain(S,A)$ is the expected reduction of entropy after knowing the value of attribute $A$(Mitchell, 1997).

### 5.2 When to stop growing the tree?

If we continue growing the tree until each node corresponds to the smallest impurity, this may cause over fitting which typically affects the generalization and increases the test set error. One possibility to avoid over fitting is to prune the tree. There are two types of pruning the tree:

- Post-pruning: this method allows the tree to be fully grown and allows over fit to occur, then start pruning the sub tree that caused over fitting. That is why it is called post-prune.
- Pre-pruning: in this method stop growing the tree before over fitting occurs.

Although Pre-pruning is the faster, but post-pruning is the most efficient way in practice, because we do not know accurately beforehand when to stop growing the tree. In post-pruning, pruning set is used for pruning the subtrees that cause over fitting. This pruning set is used to evaluate the performance of each subtree and it is entirely different from the set that is already used during the learning phase. Each subtree is replaced by a leaf node labelled by the training instances covered by the subtree. If the leaf node performs worse than the subtree on the pruning set, subtree is kept without pruning; otherwise, prune the subtree and replace it by the leaf node (Alpaydin, 2014; Mitchell, 1997).

### 5.3 Strengths and Limitations of Decision Tree

1. **Strengths:** (Rokach & Maimon, 2005; Moore, 2001; Leung, 2007; Timofeev, 2004).
   - Decision Tree is a self-explanatory tool since it has a simple schematically representation that can even be followed by the non-professionals.
   - Decision Tree can easily be converted to a set of rules which are often comprehensible for the reader.
   - Decision Tree is a non-parametric tool, therefore it does not require any functional form specification.
   - Decision Tree can easily handle outliers and missing values.
2. **Limitations:**
   - Decision Tree can be computationally expensive.
   - Decision Tree can easily overfit the data, but in practice there are several tools to avoid overfitting, such as post-prune and pre-prune.
   - In practice, decision tree is widely used for classification and less appropriate for estimation tasks in regression.

## *6 K-Nearest-Neighbours*

K- Nearest-Neighbor is an example of instance-based leaning and it is often used for classification where the task is to classify the unseen examples based on the database stored. The observations are presented in a $d$-dimensional space, where $d$is the number of attributes or characteristics which the observation has. Given a new point, it is classified according to its similarity to the rest of the data points stored in the model by some similarity measures. The algorithm decides the class of the new point by picking theK closest points to the new example and takes the most common class among them by the majority vote to be the class of the new point. If $K = 1$, the new point will be classified according to the 1-nearest neighbor data point to the new example. If $k = 2$, the class of the new point will be chosen among the 2-nearest neighbors and voting would not help. Voting helps starting from $K = 3$, classifying the new point based on the most common class among these nearest neighbors (Sutton, 2012; Larose, 2014; Tan, Steinbach, & Kumar, 2005). The intuition behind using the nearest neighbors can be clarified by the following saying “If it walks like a duck, quacks like a duck, and looks like a duck, then it’s probably a duck.”(Tan et al., 2005).

### 6.1 What is the optimal $K$?

In practice, there is no best solution for choosing $K$, it depends on the problem in hand. If $K$is too large, so the algorithm may misclassify the new point because the nearest neighbors may be located far away from its neighborhood. If $K$is too small, the algorithm is prone to over fit the data because of the noise in the training data. This will affect the generalization ability. The optimal size of $K$is the one that minimizes the classification error (Larose, 2014; Tan et al., 2005).

### 6.2 Distance Function

We have seen that how the algorithm assigns the unseen example to a specific class according to its similar neighbours. The similarity between two objects is defined by a distance function(Larose, 2014). A distance function $d$for the coordinates $x, y$and $z$has the following properties:

- $d(x,y) \geq 0 \ if \ and \ only \ if \ x = y$
- $d(x,y) = d(y,x)$
- $d(x,z) \leq d(x,y) + d(y,z)$

The first property indicates that the distance is always non-negative, the second property is the commutative property, and the last property is the triangle inequality (Larose, 2014). The most common distance function that is used to measure similarity is the Euclidian distance and it is defined by:

$$\boldsymbol{d_{Euclidian}}(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{\boldsymbol{i}}(x_{\ \boldsymbol{i}} - y_{\boldsymbol{i}})^{\boldsymbol{2}}} \qquad (19)$$

Where $\boldsymbol{x} = x_1, \dots, x_m$and $\boldsymbol{y} = y_1, \dots, y_m$and $m$is the attribute values of two points $x$and $y$. The more shorter the distance between $x$and $y$, the more similar $x$and $y$are.

### 6.3 Advantages and Disadvantages of K-Nearest-Neighbour Algorithm

K-Nearest-Neighbour algorithm is like the rest of the techniques, it has its advantages and disadvantages (Bhatia et al., 2010; Cunningham & Delany, 2007).

1.Advantages

- The training phase is very fast and its cost is zero.
- Simple and easy for implementation.
- It can deal with the noisy data.

2.Disadvantages

- It is computationally very expensive.
- It is very sensitive to the irrelevant features.
- It is a lazy algorithm (it takes more time to run).
- It needs huge memory to store all the training examples.

## 7 Methodology

### 7.1 Data sets

Data has been obtained from the UCI Repository of Machine Learning Databases (Lichman, 2013). German Credit data contains data used to evaluate credit applications in Germany. The dataset contains 1000 instances, 24 attributes and 2classes, "1" and"2", where 1indicates *Good*and 2indicates *Bad*. Weka software used for the data analysis(Hall et al., 2009).

### 7.2 Classifier Evaluation Measures

There are various measures for evaluating the classifier performance. No single measure can give us the whole story about the classifier performance. Some specific measures were used for evaluating the performance of the classifier.

**1. Accuracy:**

The accuracy of an algorithm is the measure of how correctly the algorithm classifies the unseen instances, and it can be computed by the following formula:

$$\frac{\text{Number of correctly classified instances}}{\text{Total number of instances}} \times 100$$

**2. Confusion matrix:**

Accuracy is not the only way for evaluating the performance and sometimes we may want a more detailed picture of the performance of the classifier. one such detailed is a table called Confusion Matrix and it is shown in the following table (Alpaydin, 2014).

**Table 1: Confusion matrix for two classes**

| | Predicted Class | | |
|---|---|---|---|
| Actual Class | Positive | Negative | Total |
| Positive | True Positive $(tp)$ | False Negative$(fn)$ | $p$ |
| Negative | False Positive$(fp)$ | True Negative$(tn)$ | $n$ |
| Total | $p'$ | $n'$ | $N$ |

The following rules can be extracted from the table above:

$$Accuracy = \frac{tn + tp}{tn + fn + fp + tp}$$

$$Error\ rate = 1 - Accuracy = \frac{fn + fp}{tn + fn + fp + tp}$$

$$Sensitivity = recall = \frac{tp}{tp + fn}$$

$$Specificity = \frac{tn}{fp + tn}$$

Sensitivity is the ratio of the positive examples that correctly classified. Specificity is the ratio of the negative examples that correctly classified (Alpaydin, 2014). The higher is the better for accuracy, specificity and sensitivity, but for the error rate, the lower is the better. A good classifier should be sensitive and specific with a higher degree (Alpaydin, 2014).

**3. Receiver Operating Characteristics (ROC) curve:**

Receiver Operating Characteristics (ROC) curve is a visual analysis for displaying the trade-off between false positive rate $(fb - rate)$ = 1− specificity and true positive rate $(tp - rate)$ = sensitivity. The false positive rate is shown on the $x$-axis and the $tp - rate$ is plotted along the $y$-axis. The ideal classifier should has$fp - rate = 0$and $tp - rate$= 1.

The more closely the curve to the upper-left corner, the better the classifier. The main diagonal (dashed-line) represents the random guessing of the classifier. It is clear from figure (7.a) that the classifier corresponds to the highest curve is the better.

But when they intersect like in figure (7.b), we cannot say that one is strictly better than the other. In this case, the two classifiers are better under different thresholds. The overall performance of a classifier given by the Area Under the Curve (AUC). The perfect classifier has the AUC = 1 (Alpaydin, 2014).

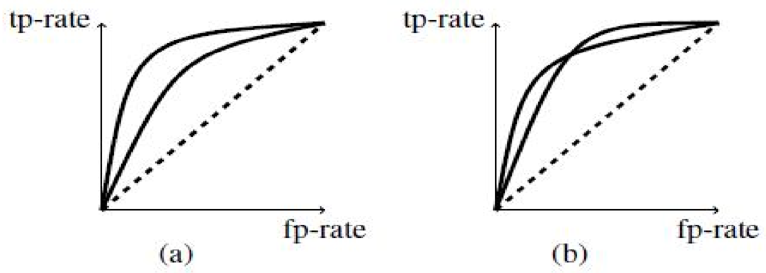

Figure 7: ROC curve and AUC

**4.Kappa Statistic:**

The Kappa Statistic measures the performance of a classifier comparing with the classifier that makes predictions based only on the random guessing. The more closely the value of the kappa statistic to 1, the better the classifier (Viera, Garrett, et al., 2005).

## *8. Results and Conclusion*

In this section, a comparison of the four algorithms has been done depending on the output of Weka software for each classifier. The following table summarizes the output that obtained from applying the classifiers on the dataset using cross validation with 10-folds. Support Vector Machine was trained using the linear function and K-Nearest Neighbor was used with K = 10.

**Table 2: Results for the four classifiers.**

| Algorithms | Accuracy | Specificity | Sensitivity | Kappa |
|---|---|---|---|---|
| Support Vector Machine | 0.763 | 0.487 | 0.881 | 0.3948 |
| Artificial Neural Network | 0.709 | 0.507 | 0.796 | 0.3038 |
| Decision Tree | 0.739 | 0.487 | 0.847 | 0.3495 |
| K-Nearest-Neighbour | 0.713 | 0.253 | 0.91 | 0.1929 |

Table 2 shows that SVM has the highest overall accuracy and the highest kappa statistic amongst all the classifiers, and it is one of the highest classifiers for specificity and sensitivity measures. In contrast, K-N-N is the lowest amongst all the classifiers for kappa statistics and specificity, whereas it is the highest in sensitivity with 91%. It means that K-N-N is the most sensitive classifier in this specific classification problem, while A-N-N is the most specific. K-N-N has roughly the same accuracy as A-N-N, but it is the lowest comparing with SVM and decision tree.Decision tree is better than A-N-N in accordance with the accuracy, the sensitivity, and the kappa statistic with 0.94, 0.847 and 0.3495 respectively. However, they have almost the same specificity. For the accuracy of the decision tree , it is better than the K-N-N, but it is the lowest comparing with the SVM. Overall, according to the previous measures, SVM seems to be one of the best classifiers for this problem. But it does not mean that SVM can outperform all the classifiers in all cases. It totally depends on the problem in hand, and the method that you are interested in and take it into consideration for evaluating the classifier performance.

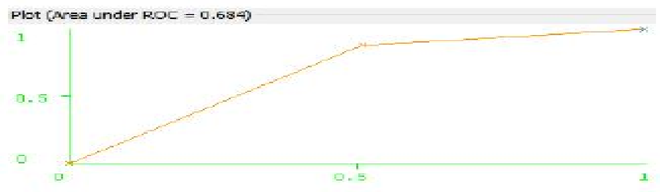

Figure 8: ROC curve for SVM

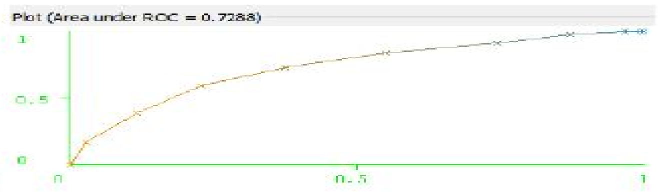

Figure 9: ROC curve for K-N-N

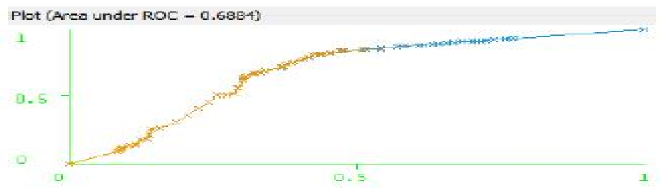

Figure 10: ROC curve for Decision Tree

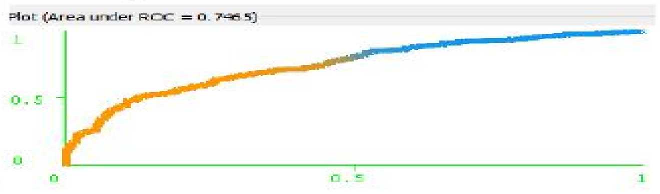

Figure 11: ROC curve for A-N-N

Figure 12: ROC curves

The different ROC curves for the four classifiers are presented in figure 12. The area under the curve measures the overall ability for the classifier to discriminate between the two classes. It is clear from the figure that A-N-N and K-N-N has the largest AUC with0.7465 and 0.7288 respectively. On the other hand, SVM and decision tree have the lowest AUC with 0.684 and 0.6884 respectively. Ideally, the classifier has an AUC = 1 .For the comparison purpose, according to the ROC curve and the AUC measures, we would say that A-N-N and K-N-N are the best classifiers for this problem. This does not contrast with the interpretation of the previous measures, because there is no one measure can summarize the whole story for the classifier performance. As mentioned above, the more closer the curve to the upper-left corner, the better the classifier. For more than one classifier in one graph, the better classifier is the one above the other, since it gives the greater AUC. If the ROC curves for different classifiers intersect like in figure 13, it is very difficult to decide which one has the better performance. In this case, we can say that the four classifiers are good under different thresholds. An optimal threshold is the one that maximize both the sensitivity and specificity so that the classifier gives the trade-off between the cost of failing to classify the positive examples as positives and the cost of raising the false positive rate.

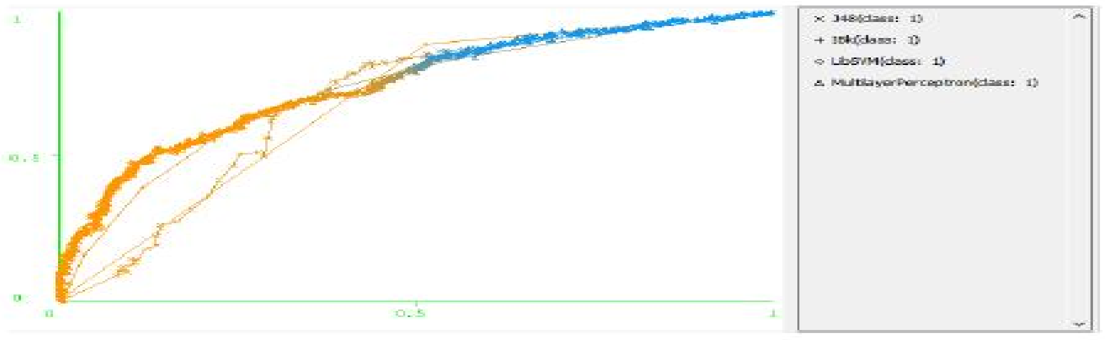

Figure 13: ROC curves for the four classifiers

## 8.1 Conclusion

This research has presented results of a comparative study carried out to explore four of the most well-known techniques of machine learning and data mining used for classification like Decision Tree, Artificial-Neural-Network, K-Nearest-Neighbor, and Support Vector Machine. Classification is very important for organizing the data so it can be easily accessed. These techniques have been gained a lot of importance in many different application areas, from finance to medicine, from astronomy to biology.

The study had shown that each technique has implemented in different areas with different data sets, each technique has its own advantages and disadvantages, and it is very difficult to find one classifier can classify all the data sets with the same accuracy. For each technique, there is a dataset where it is very accurate and another dataset where it is not. These techniques were applied to classify credit card applications. Support Vector Machine has the largest overall accuracy with 76.3% amongst all the learning algorithms in this specific dataset. The other methods also performed well and they can be a reasonable choice since they all have accuracy above 70%. The performance of the learning algorithm is strongly depending on the nature of the dataset. There is no only single measure can evaluate the classifier performance. Accuracy is not the only criterion for evaluation, but there are a lot of measures that affect our decision and none of the classifiers can satisfy all the criteria.

## *References*

Alpaydin, E. (2014). Introduction to machine learning.

Auria, L., & Moro, R. A. (2008). Support vector machines (svm) as a technique forsolvency analysis.

Basheer, I., &Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, 43(1), 3–31.

Bhatia, N., et al.(2010). Survey of nearest neighbor techniques. arXiv preprint arXiv:1007.0085.

Blanz, V., Schölkopf, B., Bülthoff, H., Burges, C., Vapnik, V., & Vetter, T. (1996). Comparison of view-based object recognition algorithms using realistic 3d models. In Artificial neural networks—icann 96 (pp. 251–256). Springer.

Boser, B. E., Guyon, I. M., &Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on computational learning theory (pp. 144–152).

Brehmer, B. (1992). Dynamic decision making: Human control of complex systems. Acta psychologica, 81(3), 211–241.

Breiman, L., Friedman, J., Stone, C. J., &Olshen, R. A. (1984). Classification and regression trees. CRC press.

Burbidge, R., & Buxton, B. (2001). An introduction to support vector machines for data mining. Keynote papers, young OR12, 3–15.

Cao, L., & Tay, F. E. (2001). Financial forecasting using support vector machines. NeuralComputing & Applications, 10(2), 184–192.

Cortes, C., &Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3),273–297.Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. InformationTheory, IEEE Transactions on, 13(1), 21–27.

Cunningham, P., & Delany, S. J. (2007). k-nearest neighbour classifiers. Multiple Classifier Systems, 1–17.

Dietterich, T. G., & Kong, E. B. (1995). Machine learning bias, statistical bias, and statistical variance of decision tree algorithms (Tech. Rep.). Technical report, Department of Computer Science, Oregon State University.

Gershenson, C. (2003). Artificial neural networks for beginners. arXiv preprint cs/0308031.

Gonzalez, C., Lerch, J. F., &Lebiere, C. (2003). Instance-based learning in dynamic decision making. Cognitive Science, 27(4), 591–635.

Goodacre, R., Howell, S. A., Noble, W. C., & Neal, M. J. (1996). Sub-species discrimination, using pyrolysis mass spectrometry and self-organising neural networks, of propionibacterium acnes isolated from normal human skin. ZentralblattfürBakteriologie, 284(4), 501–515.

Guajardo, J., Weber, R., & Miranda, J. (2006). A forecasting methodology using support vector regression and dynamic feature selection. Journal of Information & Knowledge Management, 5(04), 329–335.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009, November). The weka data mining software: An update. SIGKDD Explor. Newsl.,
11(1), 10–18. Retrieved from http://doi.acm.org/10.1145/1656274 .1656278
doi: 10.1145/1656274.1656278Haykin, S. S., Haykin, S. S., Haykin, S. S., &Haykin, S. S. (2009). Neural networks and learning machines (Vol. 3). Pearson Education Upper Saddle River.

Ibric, S., Djuriš, J., Paroj´ciˇ c, J., &Djuri´c, Z. (2012). Artificial neural networks in evaluation and optimization of modified release solid dosage forms. Pharmaceutics, 4(4), 531–550.

Jakkula, V. (2006). Tutorial on support vector machine (svm). School of EECS, Washington State University.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Springer.

Kharat, K. D., Kulkarni, P. P., &Nagori, M. (2012). Brain tumor classification using neural network based methods. International Journal of Computer Science and Informatics, 1(4).

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1), 307–319.

Kolodner, J. L. (1992). An introduction to case-based reasoning. Artificial Intelligence Review, 6(1), 3–34.

Kolter, Z. (2008). Linear algebra review and reference. Available online: http.

Krenker, A., Kos, A., &Bešter, J. (2011). Introduction to the artificial neural networks.INTECH Open Access Publisher.

Larose, D. T. (2014). Discovering knowledge in data: an introduction to data mining.John Wiley & Sons.

Lee, Y.-J., Yeh, Y.-R., &Pao, H.-K. (n.d.). An introduction to support vector machines.

National Taiwan University of Science and Technology.

Leung, K. M. (2007). Decision trees and decision rules. presentation slides, http://cis. poly. edu/˜ mleung/FRE7851/f07/decisionTrees. pdf.

Li, Y., & Cheng, B. (2009). An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification. In Geoinformatics, 2009 17th international conference on (pp. 1–4).

Lichman, M. (2013). UCI machine learning repository.Retrieved from http:// archive.ics.uci.edu/mlLiu, L., & Wang, W. (2008). Exchange rates forecasting with least squares support vector machine. In Computer science and software engineering, 2008 international conference on (Vol. 5, pp. 1017–1019).

Maciel, L. S., &Ballini, R. (2010). Neural networks applied to stock market forecasting: an empirical analysis. Journal of the Brazilian Neural Network Society, 8(1), 3–22.

Mandala, I. G. N. N., Nawangpalupi, C. B., &Praktikto, F. R. (2012). Assessing credit risk: An application of data mining in a rural bank. Procedia Economics and Finance, 4, 406–412.

Mitchell, T. M. (1997). Machine learning.

Moore, A. W. (2001). Decision trees. Presentation Carnegie Mellon University.

Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., &Vapnik, V. (1997). Predicting time series with support vector machines. In Artificial neural networks—icann'97 (pp. 999–1004). Springer.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics, 18(6), 275–285.

Osuna, E., Freund, R., &Girosi, F. (1997a). An improved training algorithm for support vector machines. In Neural networks for signal processing [1997] vii. proceedings of the 1997 ieee workshop (pp. 276–285).

Osuna, E., Freund, R., &Girosi, F. (1997b). Training support vector machines: an application to face detection. In Computer vision and pattern recognition, 1997. proceedings., 1997 ieee computer society conference on (pp. 130–136).

Podgorelec, V., Kokol, P., Stiglic, B., &Rozman, I. (2002). Decision trees: an overview and their use in medicine. Journal of medical systems, 26(5), 445–463.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81–106.

Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier.

Rachlin, J., Kasif, S., Salzberg, S., & Aha, D. W. (1994). Towards a better understanding of memory-based reasoning systems. In Icml(pp. 242–250).

Rokach, L., &Maimon, O. (2005). Decision trees. In Data mining and knowledge discovery handbook (pp. 165–192). Springer.

Salzberg, S. L. (1994). C4. 5: Programs for machine learning by j. ross quinlan. morgankaufmann publishers, inc., 1993. Machine Learning, 16(3), 235–240.

Sarkar, M., & Leong, T.-Y. (2000). Application of k-nearest neighbors algorithm on breast cancer diagnosis problem. In Proceedings of the amia symposium (p. 759).

Schalkoff, R. J. (1997). Artificial neural networks. McGraw-Hill Higher Education.

SchiilkopP, B., Burgest, C., &Vapnik, V. (1995). Extracting support data for a given task. no. x.

Schmidt, M. S. (1997). Identifying speakers with support vector networks. ComputingScience and Statistics, 305–316.

Schölkopf, B., Sung, K.-K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., &Vapnik, V. (1997). Comparing support vector machines with gaussian kernels to radial basis function classifiers. Signal Processing, IEEE Transactions on, 45(11), 2758–2765.

Sims, C. J., Meyn, L., Caruana, R., Rao, R. B., Mitchell, T., &Krohn, M. (2000). Predicting cesarean delivery with decision tree models. American journal of obstetrics and gynecology, 183(5), 1198–1206.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. Communications of the ACM, 29(12), 1213–1228.

Sutton, O.(2012).Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction. University lectures, University of Leicester.

Takagi, N. (2006). An application of binary decision trees to pattern recognition. JACIII,10(5), 682–687.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to data mining, (first edition). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. Omega, 29(4), 309–317.

Timofeev, R. (2004). Classification and regression trees (cart) theory and applications.

Viera, A. J., Garrett, J. M., et al. (2005). Understanding interobserver agreement: the kappa statistic. Fam Med, 37(5), 360–363.

Watson, I., &Marir, F. (1994). Case-based reasoning: A review. The knowledge engineering review, 9(04), 327–354.

Yavuz, T., &Guvenir, A. (1998). Application of k-nearest neighbor on feature projections classifier to text categorization. In Proceedings of the 13th international symposium on computer and information sciences–iscis(Vol. 98, pp. 135–142).