Intrusion Detection System

By Kirushikesh

Introduction

THREAT TO NETWORK SECURITY

A significant security problem for networked system is, or atleast unwanted, trespass by users or software.

- User trespass can take form of unauthorized logon to a macchine or, in case of an authorized user, acquisition of privileges or performance of actions beyond those that have been authorized.
- Software trespass can take form of a virus, worm or Trojan horse.

What is an intrusion?

Any set of actions that attempt to compromise the confidentiality, integrity, or availability of a computer resource.

Types of Intruders

- Masqueraders : A individual who is not authorized to use the computer and who penetrates a system’s access controls to exploit a legitimate user’s account.

- Misfeasor : A legitimate user who accesses data, programs or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges.

Consequences of Intrusion

Intruder attacks range from benign to the serious. At the benign end of the scale, there are many people who simply wish to explore internet and what is out there. At the serious end, intruder may attempt

- Read privileged data.
- Perform unauthorized modification of data.
- Disrupt the system settings.

Intrusion Detection Systems

- Intrusion detection is the process of identifying and responding to malicious activity targeted at resources.

- IDS is a system designed to analyze network system traffic against a set of parameters and alert when these thresholds are met.

- IDS uses collected information and predefined knowledge-based system to reason about the possibility of an intrusion.

Types of IDS

- A passive IDS simply detects and alerts. When suspicious or malicious traffic is detected an alert is generated and sent to the administrator and it is up to them to take action to block the activity or respond in some way.

- A reactive IDS will not only detect suspicious or malicious traffic and alert the administrator, but will take predefined proactive actions to respond to the threat.

IDS Detection Approaches

1. Signature-based IDS
2. Statistical anomaly based IDS

Signature Detection

It is a technique often used in the IDS and many anti-malware systems such as anti-virus and anti-spyware. In the signature detection process, network or system information is scanned against a known attack or malware signature database. If match found, an alert takes place for further actions.

Detecting new attacks is difficult and suffer from false alarms.

Have to programmed again for every new pattern to be detected.

Anomaly based IDS

It involves the collection of data relating to the behavior of legitimate users over a period of time. Then tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

These too generate many false alarms and hence compromise the effectiveness of the IDS.

Machine Learning in IDS

Unsupervised learning algorithms can “learn” the typical patterns of the network and can report anomalies without any labelled dataset. It can detect new types of intrusion but is very prone to false positive alarms.

To reduce the false positives, we can introduce a labelled dataset and build a supervised machine learning model by teaching it the difference between a normal and an attack packet in the network. The supervised model can handle the known attack deftly and can also recognise variations of those attacks.

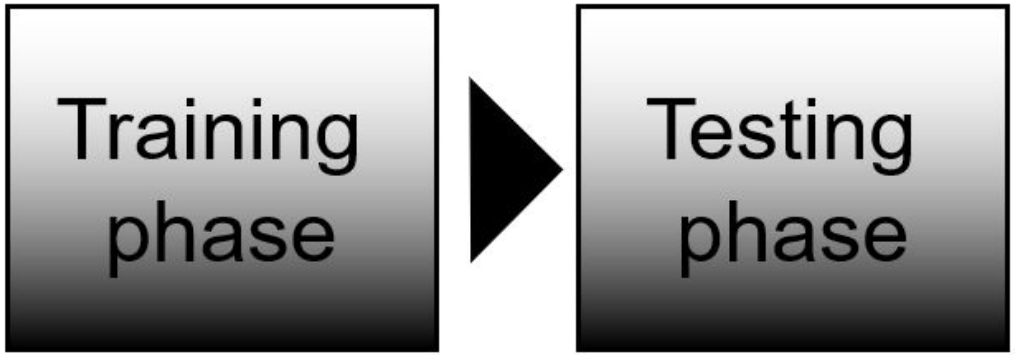
Classification

A machine learning task that deals with identifying the class to which an instance belongs.

Types of classification :

1. Binary Classification.
2. Multi-class Classification.
3. Multi-Label Classification.
4. Multi-Output Classification.

Classification Learning



Learning the classifier from the available data

'Training set'

(Labeled)

Testing how well the classifier performs

'Testing set'

Generating the Dataset

Usually we split the dataset into 3 parts mainly,

1. Training set 80%
2. Validation set 10%
3. Test set 10%

Validation set and Test set are known as Holdout set(since we are not showing this data to the model while training).

Validation set is used for

1. Selecting the best algorithm for the task.
2. Selecting the optimal value of hyperparameters for an algorithm.

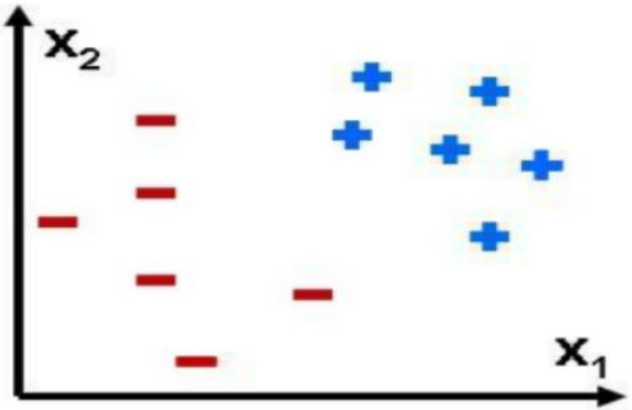
Classification Algorithms

Binary Classification

$y \in \{0, 1\}$, where 0 : "Negative class" (*e.g.*, benign tumor), 1 : "Positive class" (*e.g.*, malignant tumor)

Some more examples:

- Email: Spam/ Not Spam?
- Video: Viral/Not Viral?
- Tremor: Earthquake/Nuclear explosion?



Linear Classifier with Hard Threshold

Linear functions can be used to do classification as well as regression. For example,

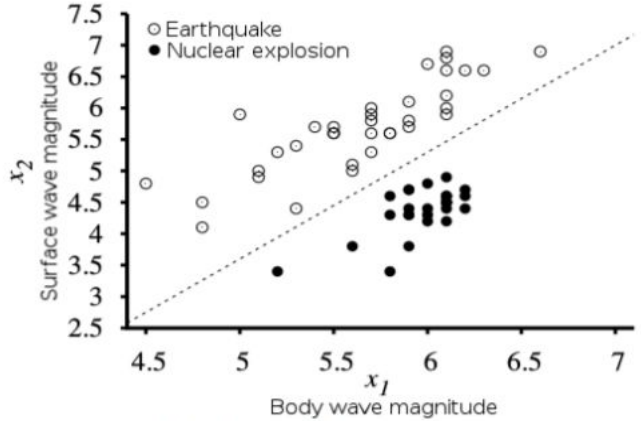


Figure credit: *AIMA: Russell, Norvig*

A **decision boundary** is a line (or a surface, in higher dimensions) that separates the two classes.

A linear function gives rise to a **linear separator** and the data that adit such a separator are called **linearly separable**.

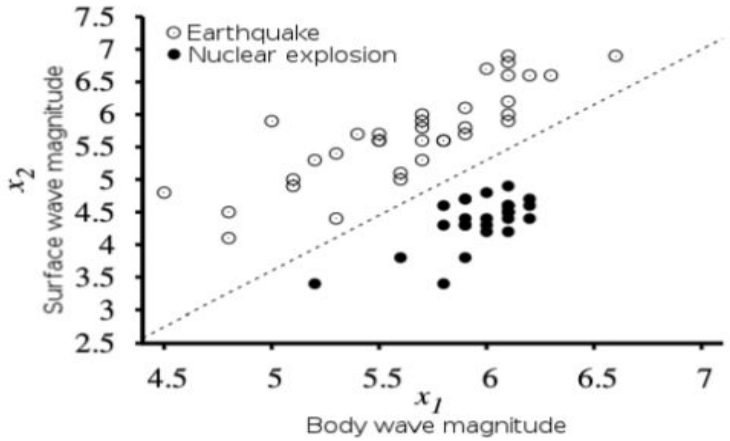
The linear separator in the associated fig is given by,

$$x_2 = 1.7x_1 - 4.9$$

$$\implies -4.9 + 1.7x_1 - x_2 = 0$$

$$\implies [-4.9, 1.7, 4.9] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = 0$$

$$\boldsymbol{\theta}^T \mathbf{x} = 0$$



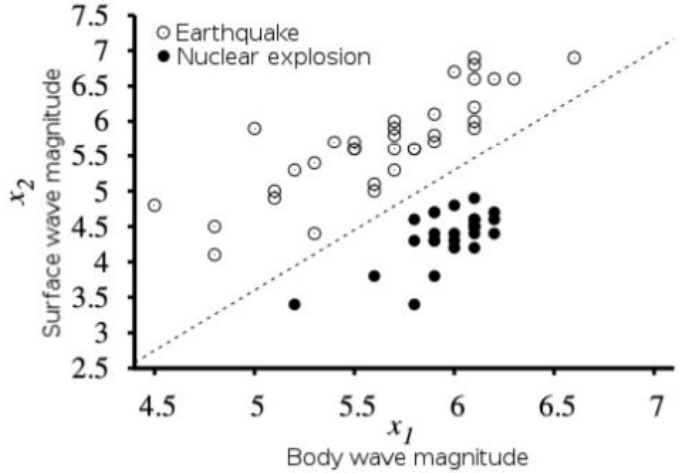
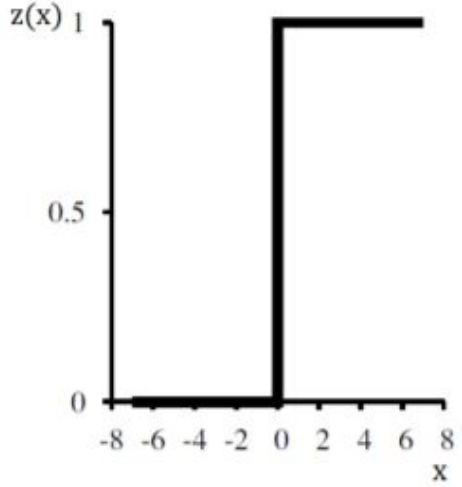


Figure credit: *AIMA: Russell, Norvig*

The explosions ($y = 1$) are to the right of this line with higher values of x_1 and lower values of x_2. So, they are points for which $\boldsymbol{\theta}^T \mathbf{x} \geq 0$
Similarly earthquakes ($y = 0$) are to the left of this line. So, they are points for which $\boldsymbol{\theta}^T \mathbf{x} < 0$
The classification rule is then,

$$y(\mathbf{x}) = \begin{cases} 1 & \text{if } \boldsymbol{\theta}^T \mathbf{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Alternatively, we can think y as the result of passing the linear function $\boldsymbol{\theta}^T \mathbf{x}$ through a threshold function.

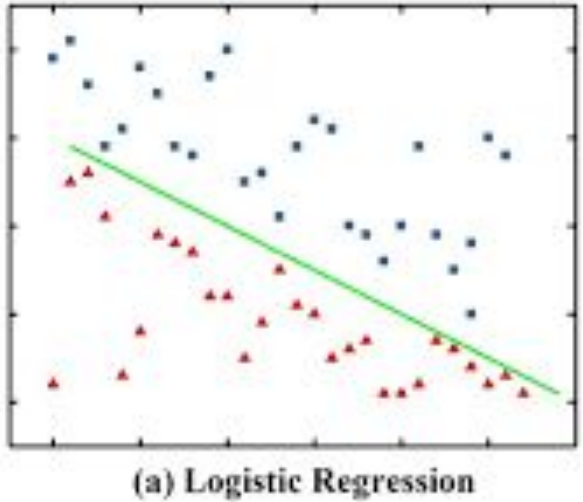


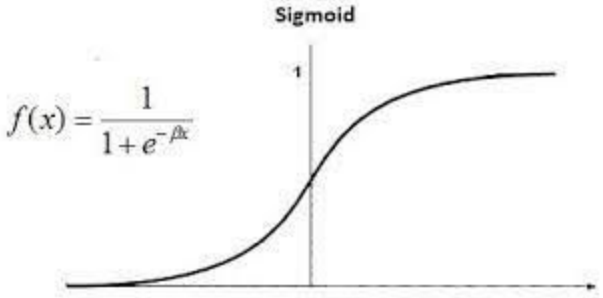
To get the linear separator we have find the θ which minimizes classification error on the training set.

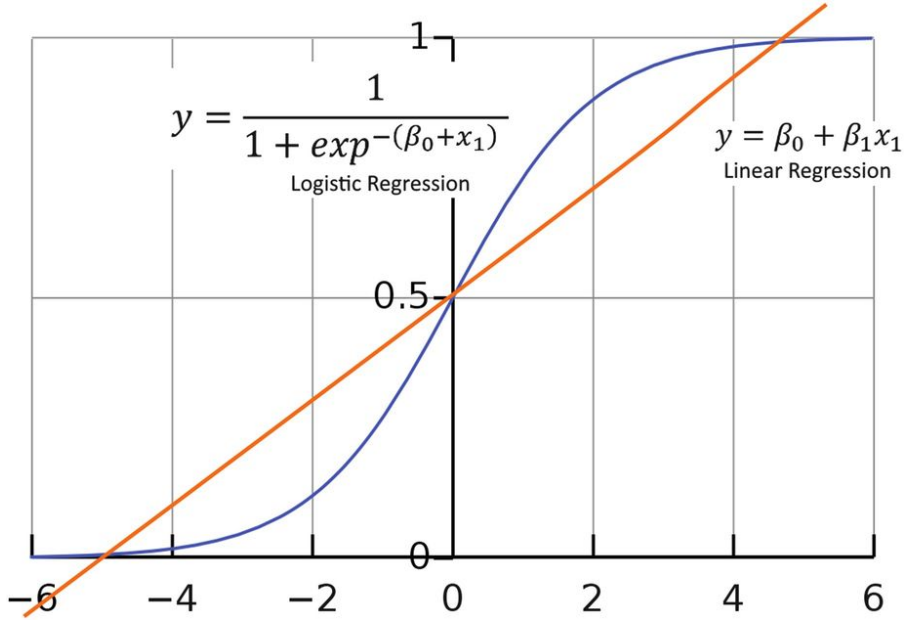
Intuition to soft threshold

What if we can get a probabilistic output. Where the instance that are farther from the decision boundary has higher confidence than the instance near decision boundary.

That is achieved using logistic function.


(a) Logistic Regression



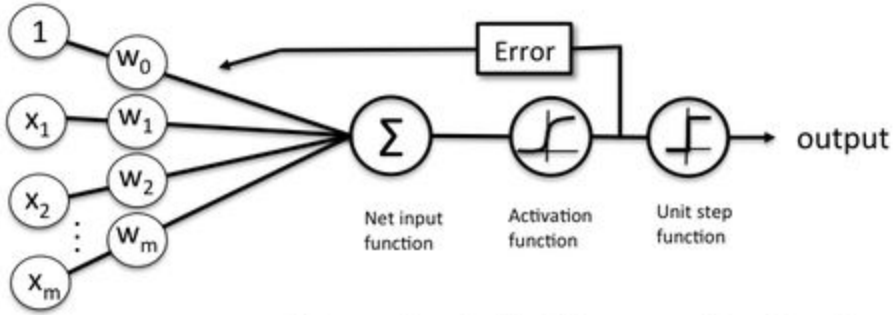


$$\sigma(u) = \frac{1}{1 + e^{-u}}$$

$$y = \sigma(h_\theta(x)) = \sigma(\boldsymbol{\theta}^T \mathbf{x})$$

The output of the logistic regression is interpreted as the probability a particular instance belongs to +ve class.

If u is larger than the output of the logistic regression is near 1 and vise versa.



Schematic of a logistic regression classifier.

Logistic regression algorithm can be only applied to a binary classification problem. To handle multi-class classification we will a extension of logistic regression known as softmax regression.

Softmax Regression

Softmax regression (or multinomial logistic regression) is a generalization of logistic regression to the case where we want to handle multiple classes. In logistic regression we assumed that the labels were binary: $y^i \in \{0,1\}$.

Softmax regression allows us to handle $y^i \in \{1,\ldots,K\}$ where K is the number of classes.

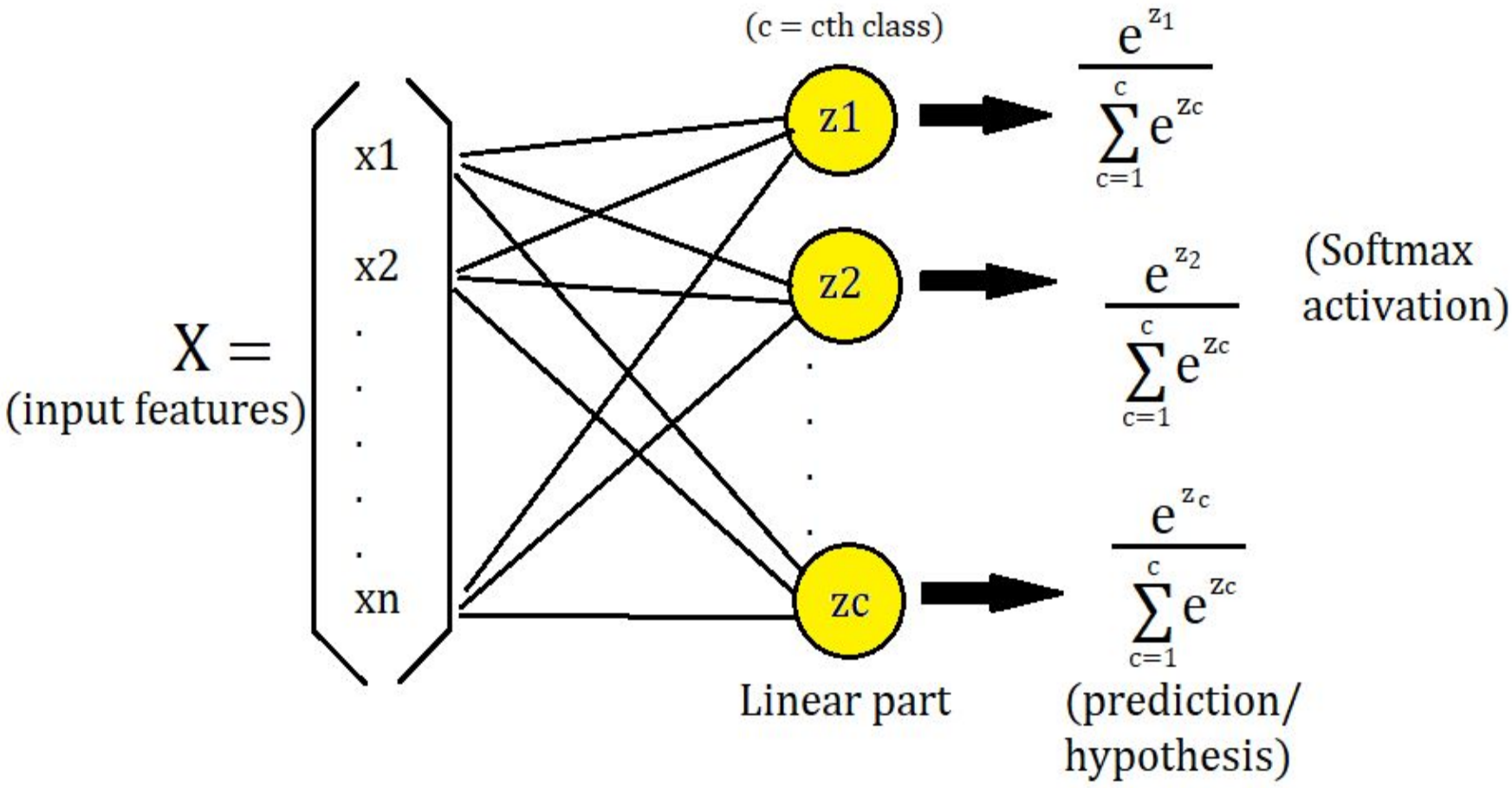
In our training set $\{(x(1),y(1)),\ldots,(x(m),y(m))\}$, we now have that $y(i) \in \{1,2,\ldots,K\}$. In our case K=5.

Given a test input x, we want our hypothesis to estimate the probability that P(y=k|x) for each value of k=1,…,K. I.e., we want to estimate the probability of the class label taking on each of the K different possible values. Thus our output will be K-dimensional.

h(x) =

$$\begin{bmatrix} P(y=1|x;\theta) \\ P(y=2|x;\theta) \\ \vdots \\ P(y=K|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x)} \begin{bmatrix} \exp(\theta^{(1)\top} x) \\ \exp(\theta^{(2)\top} x) \\ \vdots \\ \exp(\theta^{(K)\top} x) \end{bmatrix}$$

Here $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)} \in \Re^n$ are the parameters of our model. Notice that the term $\frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x)}$ normalizes the distribution, so that it sums to one.


(c = cth class)
X =
(input features)
x1
x2
xn
z1
z2
zc
$\frac{e^{z_1}}{\sum_{c=1}^{c} e^{z_c}}$
$\frac{e^{z_2}}{\sum_{c=1}^{c} e^{z_c}}$
$\frac{e^{z_c}}{\sum_{c=1}^{c} e^{z_c}}$
(Softmax activation)
Linear part
(prediction/ hypothesis)

K Nearest Neighbours Algorithm

Simple Analogy..

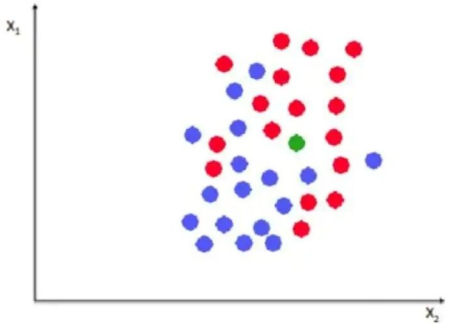
- Tell me about your friends(*who your neighbors are*) and *I will tell you who you are*.

KNN – Different names

- K-Nearest Neighbors
- Memory-Based Reasoning
- Example-Based Reasoning
- Instance-Based Learning
- Lazy Learning

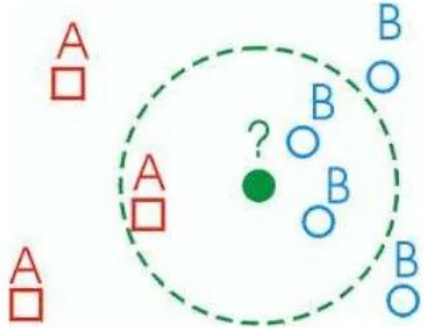
What is KNN?

- A powerful classification algorithm used in pattern recognition.
- K nearest neighbors stores all available cases and classifies new cases based on a *similarity measure*(e.g **distance function**)
- One of the top data mining algorithms used today.
- A non-parametric lazy learning algorithm (An Instance-based Learning method).



KNN: Classification Approach

- An object (a new instance) is classified by a majority votes for its neighbor classes.
- The object is assigned to the most common class amongst its K nearest neighbors.(*measured by a distant function*)



Distance measure for Continuous Variables

Distance functions

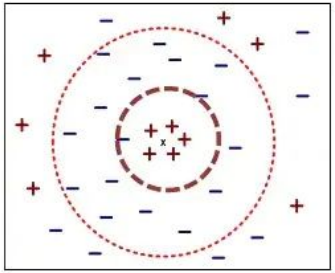
Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

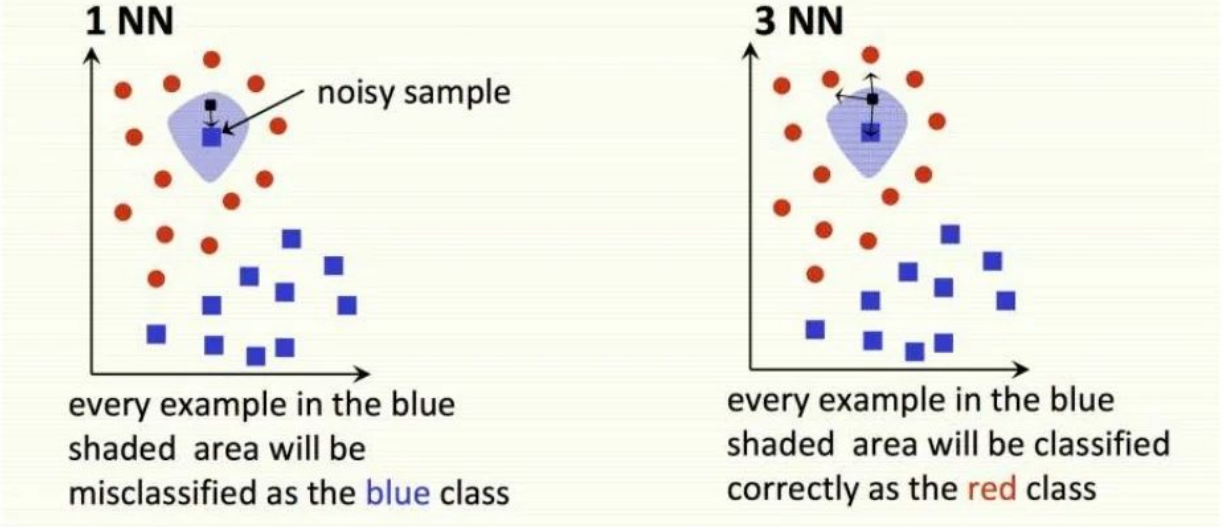
Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

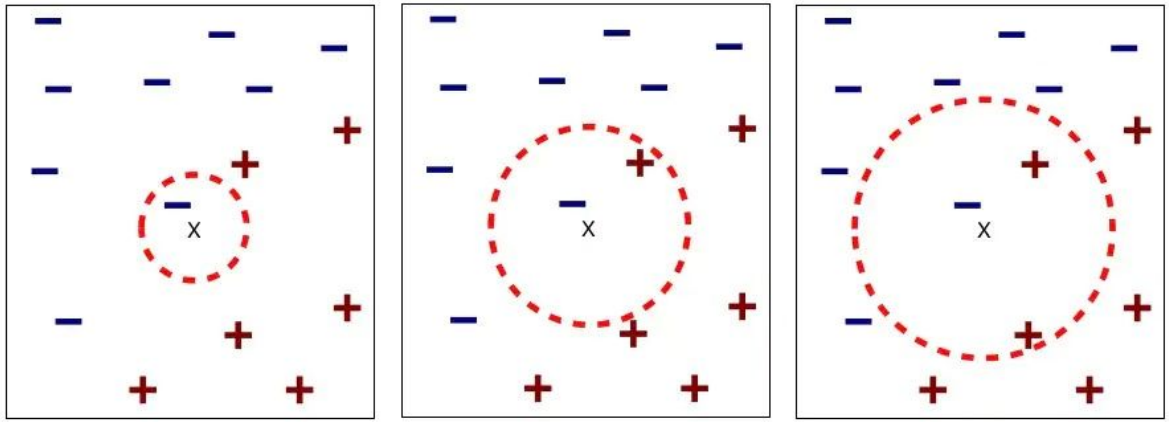
How to choose K?

- If K is too small it is sensitive to noise points.
- Larger K works well. But too large K may include majority points from other classes.



- Rule of thumb is K < sqrt(n), n is number of examples.


1 NN
noisy sample
every example in the blue shaded area will be misclassified as the blue class
3 NN
every example in the blue shaded area will be classified correctly as the red class



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Strengths of KNN

- Very simple and intuitive.
- Can be applied to the data from any distribution.
- Good classification if the number of samples is large enough.

Weaknesses of KNN

- Takes more time to classify a new example.
  - need to calculate and compare distance from new example to all other examples.
- Choosing k may be tricky.
- Need large number of samples for accuracy.

Conclusion and References

1. Andrew NG Coursera
2. Sudeshna Sarkar IITM NPTEL Course
3. Introduction to Artificial Intelligence by Stanford University
4. Scikit-learn Library for Documentation
5. Medium Blogs
6. The Hundred page machine learning book
7. Programming Collective Intelligence Book
8. Machine learning crash course by Google