# Machine Learning Final Study Guide

# Introduction

## Intro to ML, ML algorithms

The field of study interested in the development of computer algorithms to transform data into intelligent action. A computer may be more capable than a human of finding patterns in large data or perform high speed computing, but still it needs a human to motivate the analysis and turn the result into meaningful action.

## ML five-step process

1. Data Collection: It is about gathering the learning material. Data will need to be combined into a single source like a text file, spreadsheet, or database.
2. Data exploration and preparation:The goal is to learn more about the data. This step involves fixing or cleaning the data, eliminating unnecessary data, and recoding the data to conform to the learner's expected inputs.
3. Model training: At this point, you will be able to sense what to learn from the data. The specific machine learning task chosen will inform the selection of the appropriate algorithm that will represent the data in the form of a model.
4. Model evaluation(testing): It is important to evaluate how well the model learns from its experience by evaluate the accuracy of the model using a test dataset.
5. Model improvement: If better performance is needed, it becomes necessary to utilize more advanced strategies to augment performance or sometimes switch to a different type of model.

## Data in ML

Datasets that store the unit of observation and their properties can be presented as collections of data consisting of:

1. Examples(inputs, cases): Instances of the unit of observation for which properties have been recorded.
2. Features (variables): Recorded properties or attributes of examples that may be useful for learning.

## ML algorithms

- Supervised machine learning: It is the process of training a predictive model which is the attempts to discover and model the relationship between the target feature and the other feature. Predictive model is used for tasks that involve the prediction of one value using other values in the dataset.
    - Classification: Learn a model that predicts class label as a function of the values of the attributes.
    - Regression: Learn a model that predicts response variable as a function of the values of the attributes.

- Unsupervised machine learning: It is the process of training a descriptive model. As opposed to predictive model that predict a target of interest, in descriptive model, no single feature is more important than any other.Descriptive models used for tasks that would benefit from the insight gained from summarizing data in new and interesting ways.

- Pattern discovery: used to identify useful associations within data. This method is used often for Market basket analysis.
- Clustering: used to dividing a dataset into homogeneous groups. This method is used sometimes for Segmentation analysis.

## Exploring and understanding data

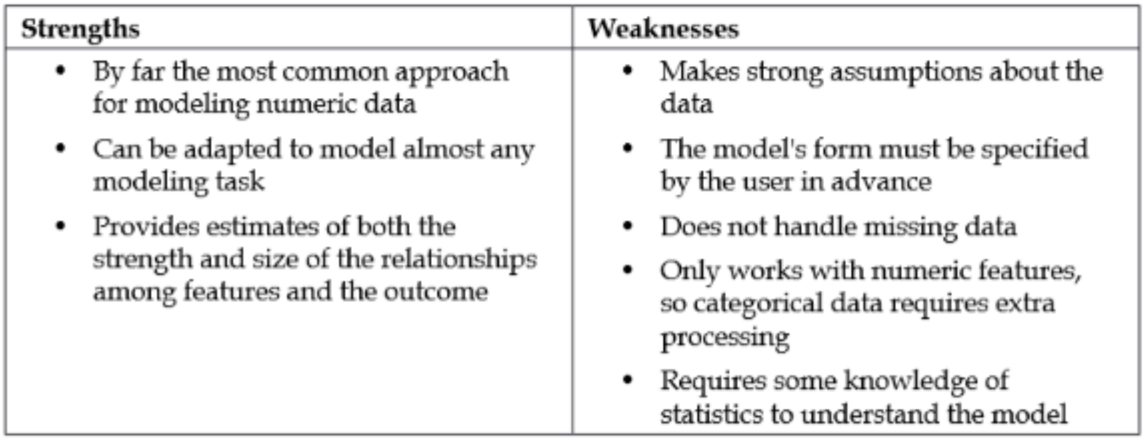
???

## Tuning parameter using caret, Understanding ensembles, Bagging, Random Forests

See #heading=h.icku5rk03ztr

# Regression/ Linear Models

1. **R package and function**
   Package: stats
   Function: lm(target_value~., data =train_data)
   predict(model, test_data)
2. **1 strength and 1 weakness**



| Strengths | Weaknesses |
|---|---|
| • By far the most common approach for modeling numeric data<br>• Can be adapted to model almost any modeling task<br>• Provides estimates of both the strength and size of the relationships among features and the outcome | • Makes strong assumptions about the data<br>• The model's form must be specified by the user in advance<br>• Does not handle missing data<br>• Only works with numeric features, so categorical data requires extra processing<br>• Requires some knowledge of statistics to understand the model |

3. **What Kind of data and Machine learning work best with this method**
   Works best with numerical data, categorical data needs extra processing. This entails dummy coding.
   Uses:
   - Examining how populations and individuals vary by their measured characteristics, for use in scientific research across fields as diverse as economics, sociology, psychology, physics, and ecology
   - Quantifying the causal relationship between an event and the response, such as those in clinical drug trials, engineering safety tests, or marketing research
   - Identifying patterns that can be used to forecast future behavior given known criteria, such as predicting insurance claims, natural disaster damage, election results, and crime rates.

4. **If there is a specific parameter used for this method, how to pick the parameter**
   N/A
5. **Do you need to adjust/change something in the data before using the method**
   If we wanna use regression on non-numerical stuff, don't we have to do the thing that turns the category into numbers? Like 0 for No and 1 for Yes or smth. Yes? Dummy coding.
6. **1 or 2 ways to improve the method**
   a. Adding Non-linear relationships
      i. When dealing with numeric data, most values are gonna be treated as a linear relationship between the output variable and the input variable. To

allow the input variable to have a stronger effect on the output, we can raise the variable to a higher order treating it as a polynomial.

b. Convert Numeric values into binary indicator
   i. Sometimes the values of a feature is only useful if they reach a certain threshold. For example if we were to use BMI, it would useful to know if a person is, or is not (binary), overweight. This would be a good case to convert that numeric value into a binary one.

c. Adding Interaction Effects
   i. Sometimes the combination of two features results in a larger effect on the output. If someone is overweight and has a existing diseases, this may result in a larger chance of mortality. To represent this in R, we use the "*" symbol. For example mortality ~ overweight*disease

# K-NN

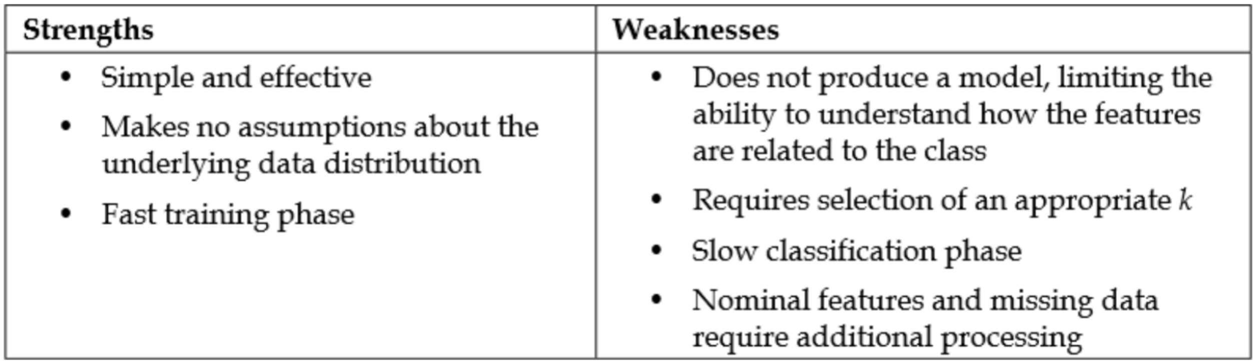
1. **R package and function**

   Package: class

   function : knn(train_data, test_data, train_class_labels, k)

   - train_data : the training data, class labels removed
   - Test_data: the testing data, class labels removed
   - Train_class_labels: the class labels for each row of the training data
   - K: int that represents the number of nearest neighbors

2. **1 strength and 1 weakness**



| Strengths | Weaknesses |
|---|---|
| • Simple and effective<br>• Makes no assumptions about the underlying data distribution<br>• Fast training phase | • Does not produce a model, limiting the ability to understand how the features are related to the class<br>• Requires selection of an appropriate *k*<br>• Slow classification phase<br>• Nominal features and missing data require additional processing |

3. **What Kind of data and Machine learning work best with this method**

   Works with both categorical data and numerical data. It is best suited to data that has many relationships among features and target classes are numerous.

4. **If there is a specific parameter used for this method, how to pick the parameter**

   The parameter that needs to be chosen with care is K. Because K is the number of neighbors that will be used to calculate what class an example is, it can change the results dramastically.

   Some ways to choose K involve:

   a. K = sqrt(n) where n is the number of observations in the training data.
   b. Test several Ks
   c. Use a larger K with weighted votes. A higher weight is given to examples close to the example to be classified.

5. **Do you need to adjust/change something in the data before using the method**

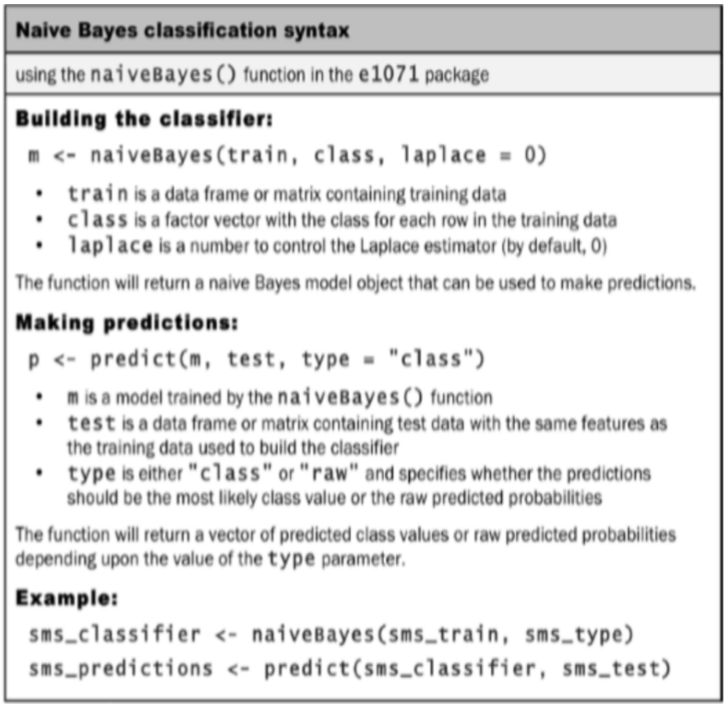
   The data needs to be rescaled. This is because this function uses a distance measurement that would be dominated by by a value from a larger range. You can use min-max or z-score re-scaling. According to the professor, z-score is better. Use dummy coding on nominal data.

6. **1 or 2 ways to improve the method**

   If the accuracy of the model is not satisfactory, the best method for improving it is changing K. To change K, see question 4.

# Problistic Learning (Naive Bayes)

1. **R package and function**



**Naive Bayes classification syntax**

using the `naiveBayes()` function in the `e1071` package

**Building the classifier:**

```
m <- naiveBayes(train, class, laplace = 0)
```

- `train` is a data frame or matrix containing training data
- `class` is a factor vector with the class for each row in the training data
- `laplace` is a number to control the Laplace estimator (by default, 0)

The function will return a naive Bayes model object that can be used to make predictions.

**Making predictions:**

```
p <- predict(m, test, type = "class")
```

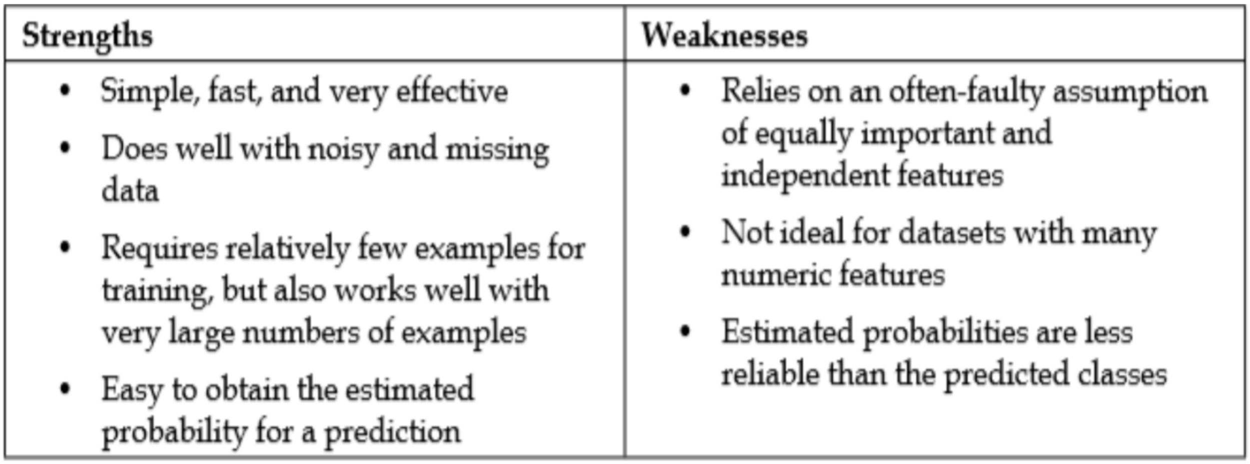
- `m` is a model trained by the `naiveBayes()` function
- `test` is a data frame or matrix containing test data with the same features as the training data used to build the classifier
- `type` is either `"class"` or `"raw"` and specifies whether the predictions should be the most likely class value or the raw predicted probabilities

The function will return a vector of predicted class values or raw predicted probabilities depending upon the value of the `type` parameter.

**Example:**

```
sms_classifier <- naiveBayes(sms_train, sms_type)
sms_predictions <- predict(sms_classifier, sms_test)
```

2. **1 strength and 1 weakness**



| Strengths | Weaknesses |
|---|---|
| • Simple, fast, and very effective<br>• Does well with noisy and missing data<br>• Requires relatively few examples for training, but also works well with very large numbers of examples<br>• Easy to obtain the estimated probability for a prediction | • Relies on an often-faulty assumption of equally important and independent features<br>• Not ideal for datasets with many numeric features<br>• Estimated probabilities are less reliable than the predicted classes |

3. **What Kind of data and Machine learning work best with this method**
   This machine learning works best with categorical data. If you choose to use numeric data, you will need to discretize the features. This is done by putting numbers into bins. How the data is separated into bins is dependant on the feature that is being binned. If the feature had to do with a 24 hour scale, it may be useful to break it into four 6 hour bins that represented, morning, afternoon, mid afternoon, and night.
4. **If there is a specific parameter used for this method, how to pick the parameter**
   The only parameter that may be useful for changing is the laplace. The laplace essentially adds 1 value to add the features ensure that no one class-feature has a value

of 0. This is important because if the value of a class-feature is 0, the formula for calculating the likelihood is 0.

5. **Do you need to adjust/change something in the data before using the method**
   You need to ensure that numeric data is discretized. See question 3.
6. **1 or 2 ways to improve the method**
   Changing the laplace. See question 4. Note that this does not always improve the model, but it has the best chance to.

   **Algorithm steps**
   1. Begin by building a frequency table
   2. Use the frequency table to build a likelihood table
   3. Multiply the conditional probabilities according to the Naive Bayes' rule.
   4. Finally, divide by the total likelihood to transform each class likelihood into a probability.

# Decision Tree

1. **R package and function**

   install.packages("C50");
   library(C50);
   m <- c5.0 (train, class, trail = 1, cost = NULL)

2. **1 strength and 1 weakness**

- Strengths
  - Trees have a very important role in informing decision process
  - It is the most widely used machine learning technique
  - It can be applied to model almost any type of data
  - An all-purpose classifier that does well on most problems
  - Highly automatic learning process which can handle numeric or nominal features, as well as missing data
  - Excludes unimportant features
  - Can be used on both small and large datasets
  - Results in a model that can be interpreted without a mathematical background
- Weaknesses
  - Trees may not be an ideal fit in task where the data has:
  - A large number of nominal features with many levels
  - A large number of numeric features
  - These cases may result in a very large number of decisions and an overly complex tree. They may contribute to decision trees to overfit data( but this weakness can be overcome by adjusting some simple parameters)
  - Decision tree models are often biased towards splits on features having a large number of levels
  - It is easy over or unfit the model
  - Can have trouble modeling some relationships die to reliance on axis-parallel splits
  - Small changes in the training data can result in large changes to decision logic
  - Large trees can be difficult to interpret and the decisions they make may seem counterintuitive

3. **What Kind of data and Machine learning work best with this method**
   - Credit scoring models in which the criteria that causes an applicant to be rejected need to be clearly documented and free from bias
   - Marketing studies of customer behavior such as satisfaction or churn, which will be shared with management or advertising agencies
   - Diagnosis of medical conditions based on laboratory measurements, symptoms, or the rate of disease progression

4. **If there is a specific parameter used for this method, how to pick the parameter**

   The 3rd parameter ‘trial’ which enables a boosting procedure. This method is model similar to AdaBoost than to more statistical approaches such as stochastic gradient boosting.

The 4th parameter ‘cost’ denotes a cost-matrix that can also be used to emphasize certain classes over others.

## 5. Do you need to adjust/change something in the data before using the method

The training and test datasets will be split into a portion of 90% to 10%, leaving 900 values for training and 100 for test. However, as the dataset is not sorted in random order, this could cause bias if for example the data is sorted by loan amounts ascending. The model will train on small loans and test on big loans. Hence, random sampling is required.

## 6. 1 or 2 ways to improve the method

1. Boosting the accuracy of decision trees
    a. Adaptive boosting: This is a process in which many decision trees are built and the trees vote on the best class for each example.
    b. to add boosting to our C5.0 decision tree, we need to add an additional trials parameter indicating the number of separate decision trees to use in the boosted team.
2. Making mistakes more costlier than others
    a. Giving a loan out to an applicant who is likely to default can be an expensive mistake. One solution to reduce the number of false negatives may be to reject a larger number of borderline applicants.
    b. C5.0 algorithm allows us to assign a penalty to different types of errors, in order to discourage a tree from making more costly mistakes. The penalties are designated in a cost matrix, which specifies how much costlier each error is, relative to any other prediction.

# Ensemble Methods

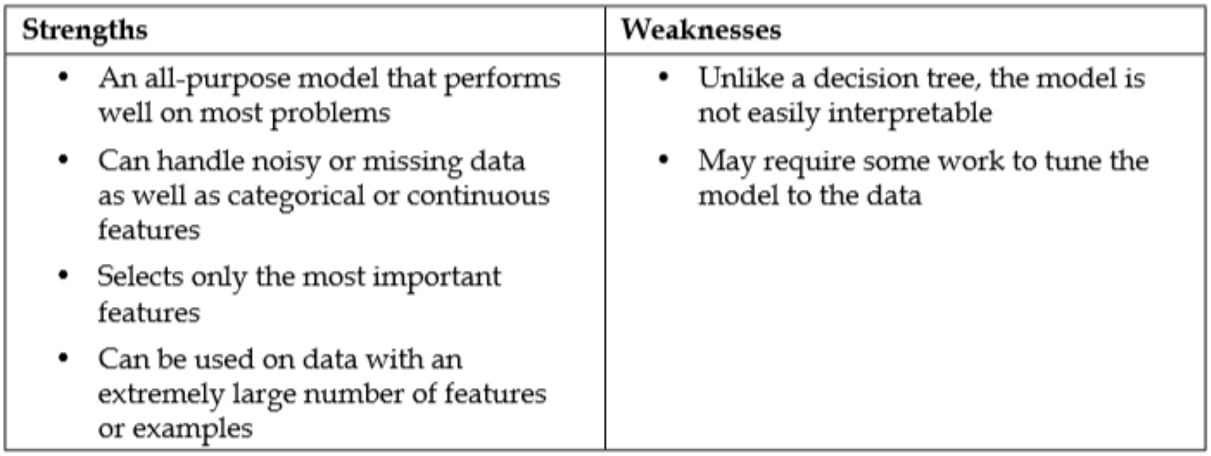
**Tuning parameter using caret, Understanding ensembles, Bagging, Random Forests**

1. **R package and function**

```
install.packages (“Caret”)
library(caret)
ctrl <-trainControl(method = "cv", number = 10, selectionFunction= "oneSE")
grid <-expand.grid(.model = "tree", .trials = c(1, 5, 10, 15, 20, 25, 30, 35), .winnow = "FALSE")
m <-train(default ~ ., data = credit, method = "C5.0", metric = "Kappa", trControl= ctrl, tuneGrid= grid)

install.packages("randomForest");
library(randomForest);
m <- randomForest (train, class, ntree = 500, mrty = sqrt(p))
```

2. **1 strength and 1 weakness**



| Strengths | Weaknesses |
|---|---|
| • An all-purpose model that performs well on most problems<br>• Can handle noisy or missing data as well as categorical or continuous features<br>• Selects only the most important features<br>• Can be used on data with an extremely large number of features or examples | • Unlike a decision tree, the model is not easily interpretable<br>• May require some work to tune the model to the data |

3. **What Kind of data and Machine learning work best with this method**

   ?

4. **If there is a specific parameter used for this method, how to pick the parameter**

   The 3rd parameter ntree is an integer specifying the number of trees to grow.
   The 4th parameter mtry is an optional integer specifying the number of features to randomly select at each split (use sqrt(p) by default, where p is the number of features in the data)

5. **Do you need to adjust/change something in the data before using the method**

   The training and test datasets will be split into a portion of 90% to 10%, leaving 900 values for training and 100 for test. However, as the dataset is not sorted in random order, this could cause bias if for example the data is sorted by loan amounts ascending. The model will train on small loans and test on big loans. Hence, random sampling is required.

## 6. 1 or 2 ways to improve the method

? The core functionality is provided by a **train( )**function that serves as a standardized interface for over 175 different machine learning models for both classification and regression tasks. By using this function, it is possible to automate the search for optimal models using a choice of evaluation methods and metrics. ?

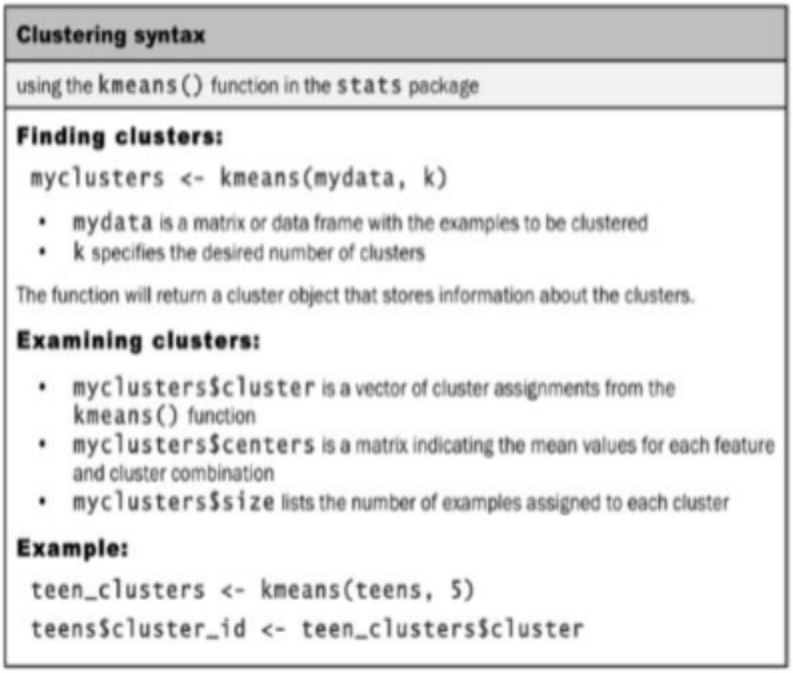
Increase number or trees made.

# Clustering

1. **R package and function**

   Package: stats

   Functions:



**Clustering syntax**

using the `kmeans()` function in the `stats` package

**Finding clusters:**

```
myclusters <- kmeans(mydata, k)
```

- `mydata` is a matrix or data frame with the examples to be clustered
- `k` specifies the desired number of clusters

The function will return a cluster object that stores information about the clusters.

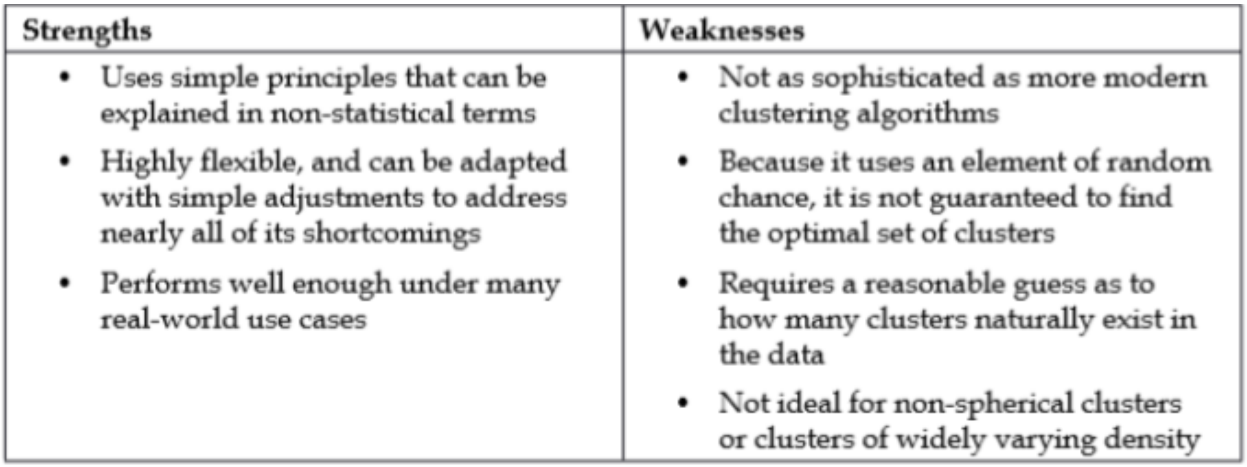
**Examining clusters:**

- `myclusters$cluster` is a vector of cluster assignments from the `kmeans()` function
- `myclusters$centers` is a matrix indicating the mean values for each feature and cluster combination
- `myclusters$size` lists the number of examples assigned to each cluster

**Example:**

```
teen_clusters <- kmeans(teens, 5)
teens$cluster_id <- teen_clusters$cluster
```

2. **1 strength and 1 weakness**



| Strengths | Weaknesses |
|---|---|
| • Uses simple principles that can be explained in non-statistical terms<br>• Highly flexible, and can be adapted with simple adjustments to address nearly all of its shortcomings<br>• Performs well enough under many real-world use cases | • Not as sophisticated as more modern clustering algorithms<br>• Because it uses an element of random chance, it is not guaranteed to find the optimal set of clusters<br>• Requires a reasonable guess as to how many clusters naturally exist in the data<br>• Not ideal for non-spherical clusters or clusters of widely varying density |

3. **What Kind of data and Machine learning work best with this method**

Numeric data works best with clustering.

Uses:

- Segmenting customers into groups with similar demographics or buying patterns for targeted marketing campaigns
- Detecting unusual behavior, such as unauthorized network intrusions, by identifying patterns of use falling outside the known clusters

- Simplifying extremely large datasets by grouping features with similar values into a smaller number of homogeneous categories

4. **If there is a specific parameter used for this method, how to pick the parameter**
   The specific parameter you want to change is the K value. One rule for choosing K is the square root of (n/2) where n is the number of observations in the data. Option 2 for choosing K is to use the elbow method.
   Long Explanation:
   https://learning.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml
5. **Do you need to adjust/change something in the data before using the method**
   You need to rescale the data because the k-means function using Euclidean distance to calculate which examples belong in each cluster.
6. **1 or 2 ways to improve the method**
   a. Changing K
      i. When choosing a K, you want to find a value that is not too large or too small. This is usually a trial and error search, even with the methods mentioned above. Choosing a K that is too large creates an excessive amount of groups that many not have any meaningful similarities between the groups. Choosing a K that is too small however increases the diversity of the observations in the groups making them not useful for analysis.

# Neural Network

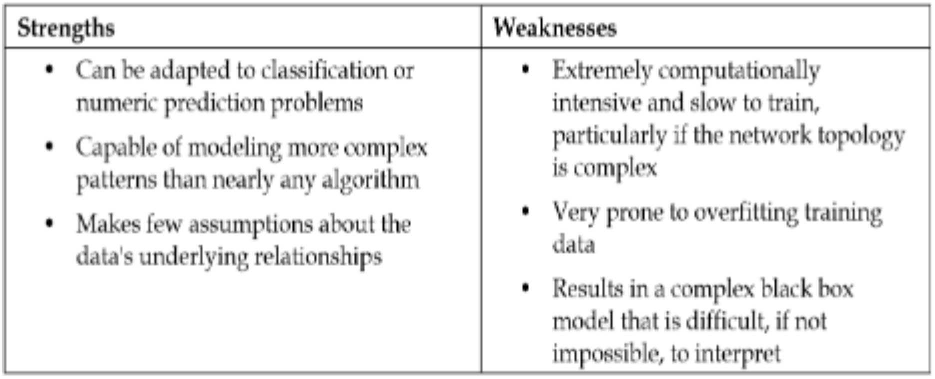
## 1. R package and function

Package: neuralnet

Function:

- Building a model: neuralnet(target ~ predictors, data = train_data, hidden = #)
  - target : formula to find the target value
  - Data: …...the training data
  - Hidden: Number of hidden layers
- Making a prediction: compute(model, test_data )
  - Model: From the previous function
  - test_data : the test data...

## 2. 1 strength and 1 weakness



| Strengths | Weaknesses |
|---|---|
| • Can be adapted to classification or numeric prediction problems<br>• Capable of modeling more complex patterns than nearly any algorithm<br>• Makes few assumptions about the data's underlying relationships | • Extremely computationally intensive and slow to train, particularly if the network topology is complex<br>• Very prone to overfitting training data<br>• Results in a complex black box model that is difficult, if not impossible, to interpret |

## 3. What Kind of data and Machine learning work best with this method

This machine learning technique best works with all data that is fairly simple. If the data is numeric, you should rescale the data to be in the same range.

Uses:

- Speech and handwriting recognition programs like those used by voicemail transcription services and postal mail sorting machines
- The automation of smart devices like an office building's environmental controls or self driving cars and self piloting drones
- Sophisticated models of weather and climate patterns, tensile strength, fluid dynamics, and many other scientific, social, or economic phenomena

## 4. If there is a specific parameter used for this method, how to pick the parameter

The the specific parameter used for this function is the Hidden parameter. This parameter changes the number of hidden layer. Hidden layers are used to transform the inputs into something that the output layer can use. If more than 1 hidden layer, you are now doing deep learning.

Long explanation: [https://stats.stackexchange.com/questions/63152/what-does-the-hidden-layer-in-a-neural-network-compute](https://stats.stackexchange.com/questions/63152/what-does-the-hidden-layer-in-a-neural-network-compute)

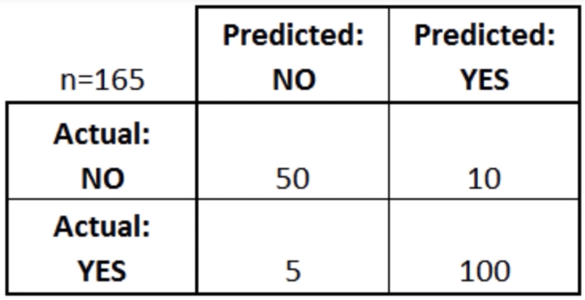
5. **Do you need to adjust/change something in the data before using the method**
   Yes, you need to rescale the data. This ensures that no single value has a advantage of affecting the neurons.

6. **1 or 2 ways to improve the method**

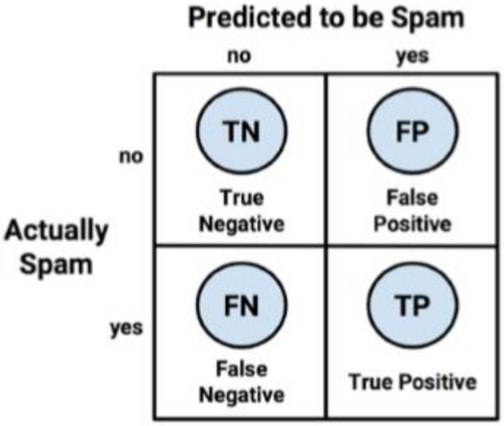
To improve this method you can change the number of hidden layer. This will make the model build slower, however it decreases the error by increasing the number of steps through the network.

# Confusion Matrix Measures to Evaluate a Model

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data from which the true values are known.



| n=165 | Predicted:<br>NO | Predicted:<br>YES |
|---|---|---|
| Actual:<br>NO | 50 | 10 |
| Actual:<br>YES | 5 | 100 |



There are two possible prediction classes: “yes” and “no”. Out of those 165 cases the classifier predicted “yes” 110 times and “no 55 times. 60 cases where actually no and 105 cases where actually yes.

False positives are type 1 errors. They are the worst type of errors.