What is Learning?
"Learning denotes changes in a system that enable a system to do the same task more efficiently the next time." or we can say "Learning is constructing or modifying representations of what is being experienced."

What is Machine Learning?
Machine learning is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables.

Why do Machine Learning?
Discover new things or structure that is unknown to humans eg. data mining. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data.

Types of Machine Learning
- **Supervised learning**
  - Tasks, where the desired output for each object is given, are called supervised and the desired outputs are called targets.
  - generates a function that maps inputs to desired outputs
- **Unsupervised learning**
  - If data has to be processed by machine learning methods, where the desired output is not given, then the learning task is called unsupervised.
  - models a set of inputs, like clustering.
- **Semi-supervised learning**
  - combines both labeled and unlabeled examples to generate an appropriate function or classifier.

Machine Learning in Bioinformatics
- Protein secondary structure prediction (neural networks, support vector machines)
- Gene recognition (hidden Markov models)
- Multiple alignment (hidden Markov models, clustering)
- Splice site recognition (neural networks)
- Microarray data: normalization (factor analysis)
- Microarray data: gene selection (feature selection)
- Microarray data: Prediction of therapy outcome (neural networks, support vector machines)
- Microarray data: dependencies between genes (independent component analysis, clustering)
Methods of Unsupervised Learning

1. Hierarchical Clustering-
Hierarchical clustering algorithms partition the objects into a tree of nodes, where each node represents a cluster. Each node in a tree has zero or more child nodes, which are below it in the tree; by convention, tree grow down, not up as they do in nature. Hierarchical methods include:

**Agglomerative:** Initially, many small clusters are formed, which are merged based on their similarity. Finally one cluster contains all the objects.

a) **Divisive:** Initially, all objects form one cluster, which is decomposed into smaller clusters.

b) **Steps in Agglomerative Clustering**
- Construct the similarity matrix
- Find largest value in similarity matrix
- Join clusters together, which are having largest value
- Recompute matrix and iterate the process till all the clusters join together

Distances between two clusters are determined by following linkage algorithms.

**Single linkage:** the distance between two clusters is their minimum distance.

\[
\min_{x \in A, y \in B} d(x, y)
\]

**Complete linkage:** the distance between two clusters is their maximum distance.

\[
\max_{x \in A, y \in B} d(x, y)
\]

**Average linkage:** takes the mean distance between all pairs of objects of two clusters.

\[
\frac{1}{\text{card}(A) \times \text{card}(B)} \sum_{x \in A} \sum_{y \in B} d(x, y)
\]

Types of hierarchical clustering on the basis of data type i.e. numerical and discrete

Hierarchical clustering for numerical data-

1. **BIRCH**- It is suitable for large databases, improving the slow runtimes of other hierarchical algorithms.

2. **CURE**- It improves upon BIRCH by ability to discover clusters of arbitrary shapes and also more robust with respect to outliers.

Hierarchical clustering for discrete data-

1. **ROCK**- It is an agglomerative algorithm and assumes a similarity measure between objects and defines a ‘link’ between two objects whose similarity exceeds a threshold. It is not suitable for large datasets.
2. **Chameleon**- It improves upon CURE and ROCK by considering the internal interconnectivity and closeness of the objects both between and within two clusters to be merged.

3. **LIMBO**- It improves on the scalability of other hierarchical clustering algorithms. It builds on the information bottleneck (IB) framework for quantifying the relevant information preserved when clustering and able to handle large datasets.

2. **Partition Clustering**-
   In this approach, objects are partitioned and may change clusters based on dissimilarity. Partitioning methods are useful for bioinformatics applications where a fixed number of clusters is desired, such as small gene expression datasets.

1. **K-Means**- In this, the user specifies the number of clusters k. Clusters have a representative, which is the mean vector, for finding the closest cluster to an object, which minimizes mean-object dissimilarity with a metric such as Euclidean distance. K-Means iteratively assigns objects to the closest of k clusters, and the means get updated. The iteration continues until the number of objects changing clusters is below a user-specified threshold. It deals with numerical attribute values but it is also applicable to binary datasets. It is unsuitable for discovering arbitrary shaped clusters and handling noisy datasets.

   **Steps in K-Means Clustering**
   - Randomly generate k clusters and determine the cluster centers or directly generate k seed points as cluster centers
   - Assign each point to the nearest cluster center.
   - Recompute the new cluster centers.
   - Repeat until some convergence criterion is met (usually that the assignment hasn't changed).

2. **Farthest First Traversal k-centre (FFT) algorithm**- It improves the k-means complexity to terminate at a local optimum. It deals with k-means sensitivity to initial cluster means, by ensuring means represent the dataset variance. But like k-Means, it is also not suitable for noisy datasets.

3. **k-Medoids or PAM**- It deals with k-means problem of outliers by setting a cluster’s median to the object that is nearest to the ‘centre’ of the cluster. k-Medoids involves reducing the distance between all objects in a cluster and the central object. It does not scale well to large datasets.

4. **CLARA** (Clustering Large Applications)- It improves the k-Medoids by extending it with focus on scalability. It selects a representative sample of the entire dataset. Medoids are then chosen from this sample, similar to k-medoids. If the sampling is done properly, the medoids chosen from the sample are similar to the ones that would have been chosen from the whole dataset.

5. **Fuzzy k-means**- The k-means clusters are ‘hard’, since an object either is or is not a member of a cluster. Fuzzy k-means produces ‘soft’ or ‘fuzzy’ clusters, where an object has a degree of membership in each cluster. Fuzzy k-means was applied to gene expression to study overlapping gene sets.
6. **K-modes** - It is a discrete adaptation of k-means, with similar runtime, benefits and drawbacks.

7. **COOLCAT** - It deals with k-modes sensitivity to the initial cluster modes but sensitive to the order of object selection.

3. **Model Based Clustering**
Model based clustering assumes that objects match a model, which is often a statistical distribution. Then, the process aims to cluster objects such that they match the distribution. The model may be user specified as a parameter and the model may change during the process. In bioinformatics, model-based clustering methods integrate background knowledge into gene expression and sequences.

1. **Self Organizing Maps** - It involves neural networks, which resemble processing that occurs in the brain. SOM clustering involves several layers of units that pass information between one another in weighed manner.

   **Steps in Self Organizing Maps**
   - Initialize the map by assigning weight \( t=0 \)
   - Randomly select a sample
   - Get best matching unit
   - Determining neighbours
   - Increase \( t \) by a small amount
   - Repeat the process until \( t=1 \)

2. **COBWEB** - It is a conceptual clustering method which creates a hierarchical clustering in the form of classification tree. It integrates observations incrementally into an existing classification tree by classifying the observation along a path of best matching nodes. A benefit of COBWEB is that it can adjust the number of clusters in a partition, without the user specifying this input parameter.

3. **Consensus Clustering**
Different clustering on the same data can be obtained either from different experimental sources or from multiple runs of non-deterministic clustering algorithms. Consensus clustering formalizes the idea that combining different clusterings into a single representative, or consensus, would emphasize the common organization in the different data sets and reveal the significant differences between them. The goal of consensus clustering is to find a consensus which would be representative of the given clusterings of the same data set.

   **Procedure of Consensus Clustering**
   **Input**: a set of items \( D = \{e_1, e_2, ..., e_N\} \)
   a clustering algorithm \( \text{Cluster} \)
   a resampling scheme \( \text{Resample} \)
   number of resampling iterations \( H \)
   set of cluster numbers to try, \( K = \{K_1, ..., K_{max}\} \)
   for \( K \in K \) do
     \( M \leftarrow \emptyset \) \{set of connectivity matrices, initially empty\}
     for \( h=1, 2, ..., H \) do
       \( D(h) \leftarrow \text{Resample}(D) \) \{generate perturbed version of \( D \)\}
       \( M(h) \leftarrow \text{Cluster}(D(h), K) \) \{cluster \( D(h) \) into \( K \) clusters\}
       \( M \leftarrow \bigcup M(h) \)
Unsupervised Machine Learning Techniques in Genomics

\[
\text{end} \{ \text{for } h \} \\
M(K) \leftarrow \text{compute consensus matrix from } M = \{ M(1), \ldots, M(H) \} \\
\text{end} \{ \text{for } K \} \\
\hat{K} \leftarrow \text{best } K \in K \text{ based on consensus distribution of } M(K)'s \\
P \leftarrow \text{Partition } D \text{ into } \hat{K} \text{ clusters based on } M(\hat{K}) \\
\text{Return } P \text{ and } \{ M(K) : K \in K \}
\]

Practical Exercise - HIERARCHICAL CLUSTERING

Download the dataset of 'drosophilaaging_wide.sas7bdat' (example dataset located in Sample Data\ Microarray\Scananalyse Drosophila directory included with JMP Genomics). Details of the experiment: The experiment consists of 24 two color cDNA microarrays, 6 for each experimental combination of 2 lines (Oregon and Samarkand), 2 sexes (Male and Female) and 2 ages (1 week and 6 weeks). The Cy3 and Cy5 dyes were flipped for two of the six replicates for each genotype and sex combination. The design is a split plot design with Age and Dye as subplot factors and Line and Sex as whole plot factors. A total of 4256 clones were spotted on the arrays but for this example, we use a subset containing 100 randomly selected genes.

STEPS:

- Open JMP Genomics → Genomics → Pattern Discovery → Hierarchical Clustering → Choose → Open Data File → Navigate into the Sample Data\ Microarray\Scananalyse Drosophila directory → select "drosophilaaging_wide.sas7bdat" → Open.

- Examine the list of available variables using List-Style Specification option.

- Leave the Variables Whose Rows are to be Clustered and the Ordering Variable fields blank.
- Highlight Array in the Available Variables field.
- Click-> to add Array to the Label Variable field.
- Hold the <Ctrl> key as you highlight age, sex and line in the Available field.
- Click -> to add age, sex and line to the Compare Variables field.
For our example dataset, we do not need to subset the data into different groups. Leave the **By Variables** field blank.

To specify those variables, all of which begin with the string log2in, whose rows are to be clustered, use the **List-Style Specification** field.

Type **log2in:** in the **List-Style Specification of Variables Whose Rows are to be Clustered** field.

To specify an output folder, complete the following steps:
- Click **Choose**
- Navigate to the folder where you wish to place the output or create new folder.
- Click **OK**.

Completed General tab is shown as follows:

- **Click Options**
- Examine **Option** tab.
- Use default setting for the given example as follows:

![](image)

- **Click Run.**
- **RESULTS:**
The output data is essentially identical to the input data set except for the addition of 7 columns. Output data has 112 columns and 48 rows.

The data listed in the last seven columns of the output data set are used to generate a dendroprogram plot. This reveals 2 primary branches that correspond perfectly with Sex variable.
Practical Exercise – K Means CLUSTERING
Dataset remains the same as in previous exercise.
Download the dataset of `drosophilaaging_wide.sas7bdat` (example dataset located in Sample Data\Microarray\Scananalyse Drosophila directory included with JMP Genomics.)

**STEPS:**
- Open JMP Genomics → Genomics → Pattern Discovery → K means Clustering → Choose → Open Data File → Navigate into the Sample Data\Microarray\Scananalyse Drosophila directory → select “drosophilaaging_wide.sas7bdat” → Open.
- Highlight Array in the Available Variable field.
- Click-> to add Array to the Label Variable field.
- Since a large number of variables are to be clustered, specify these variables using List-Style Specification option.
- Leave the Variables Whose Rows are to be Clustered field blank.
- To specify those variables, all of which begin with the string log2in, whose rows are to be clustered, use the List-Style Specification field.
- Type `log2in:` in the List-Style Specification of Variables Whose Rows are to be Clustered field.

To specify an output folder, complete the following steps:
- Click Choose
- Navigate to the folder where you wish to place the output or create new folder.
- Click OK.
- Completed General tab is shown as follows:

- Click Analyse
- Examine Analysis tab
- Make sure that Automated K Means is selected as the clustering method
- Type 5 in the Number of Clusters field
- Make sure that the remainder of the settings are set to default values.
- Complete Analysis tab should appear as follows:
Unsupervised Machine Learning Techniques in Genomics

- Click **Options**
- Examine **Option** tab
- Click **Run**