# Machine Learning

## Terminologies

### Variables / Features / Attributes / Vectors

In machine learning and pattern recognition, a feature is an individual measurable property of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression. Features are usually numeric, but structural features such as strings and graphs are used in syntactic pattern recognition.

In character recognition, features may include histograms counting the number of black pixels along horizontal and vertical directions, number of internal holes, stroke detection and many others.

In speech recognition, features for recognizing phonemes can include noise ratios, length of sounds, relative power, filter matches and many others.

In spam detection algorithms, features may include the presence or absence of certain email headers, the email structure, the language, the frequency of specific terms, the grammatical correctness of the text.

In computer vision, there are a large number of possible features, such as edges and objects.

### Dependent and Independent Variables

An independent variable, sometimes called an experimental or predictor variable, is a variable that is being manipulated in an experiment in order to observe the effect on a dependent variable, sometimes called an outcome variable or target variable.

### Labeled and Unlabeled Data / Points

Labeled Data points are the ones which are pinned with target variable. Training data set has labeled data points. Also called as class.

Unlabeled data points are part of test data. Where target variable has to be predicted based on the training data set (i.e. with labeled data points)

### Categorical / Discrete Variables

Qualitative variables – variables that have two or more categories.

* Nominal – two or more categories like real-estate property type can be house, condos, co-ops.
* Dichotomous – only two categories like male/female
* Ordinal – same or nominal, but categories can be ordered or ranked like, A+, A, B+, B grades

### Continuous Variables

Quantitative variables - variables that can be measured along a continuum (i.e. in a range) – like temperature, height, weight, age, win/loss etc..,

* Interval
* Ratio

Reference: <https://statistics.laerd.com/statistical-guides/types-of-variable.php>

### Overfitting and Overgeneralization (underfitting)

The cause of poor performance in machine learning is either overfitting or underfitting the data.

Overfitting refers to a model that models the training data too well. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.

Underfitting refers to a model that can neither model the training data nor generalize to new data.

## Types of Algorithms

**Supervised learning**: This algorithm consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. These types of models use labeled data to learn. Classification and Regression are commonly classified as supervised learning.

Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

Regression is used to predict continuous variables. Dependent variable (continuous) value can range from 0 to infinity, it doesn’t have a negative value, and also no higher range, so we use regression model.

Classification is used to predict (categorical / discrete variables) which class a data point is part of.

**Semi-Supervised learning:** It is a class of supervised learning tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data.

**Unsupervised** **learning**: When a model does not require labeled data, we refer to unsupervised learning. In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Clustering is commonly classified as unsupervised learning

Examples of Unsupervised Learning: Apriori algorithm, K-means.

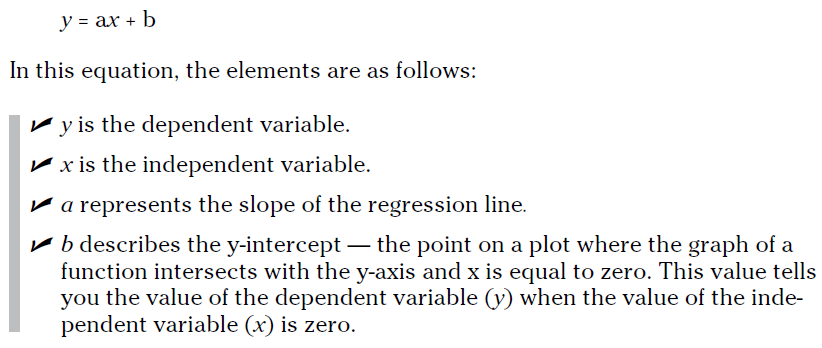
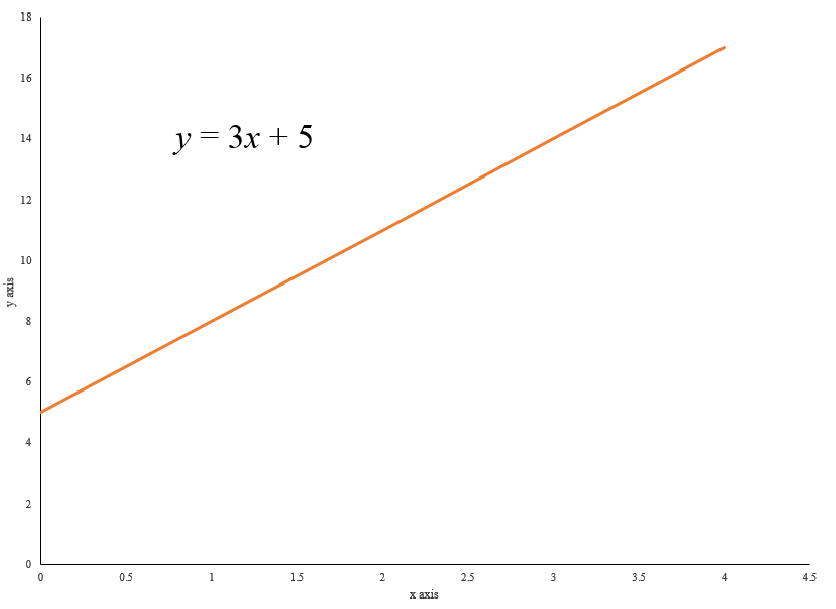
**Reinforcement Learning:**

Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions.

Example of Reinforcement Learning: Markov Decision Process

## Linear Regression

Use to estimate real values based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a \*X + b.



**Ordinary Least Squares (OLS)** is a statistical method that fits a linear regression line to an observational dataset. With OLS, you do this by squaring the vertical distance values that describe the distances between the observational data points and the best‐fit line, adding up those squared distances, and then adjusting the placement of the best‐fit line so that this summed squared distance value is minimized. Use OLS if your goal is to construct a function that’s a close approximation of observational data.

Y = ax + b + 

To get meaningful results from your analysis, you need to make certain assumptions about your data. The  (random variable) regression errors for each of the observations should be normally distributed, the regression errors should average out to zero, and the regression errors should have approximately the same amount of variability. If your data does not meet these specification, then a linear model might not be appropriate.

Linear Regression is of mainly two types: Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is characterized by one independent variable. And, Multiple Linear Regression (as the name suggests) is characterized by multiple (more than 1) independent variables. While finding best fit line, you can fit a polynomial or curvilinear regression. And these are known as polynomial or curvilinear regression.

## Logistic Regression (Classification)

Don’t get confused by its name! It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s).

Logistic regression is similar to linear regression, but the curve is constructed using the natural log-arithm of the “odds” of the target variable, rather than the probability.



odds= p/ (1-p) = probability of event occurrence / probability of not event occurrence

ln(odds) = ln(p/(1-p))

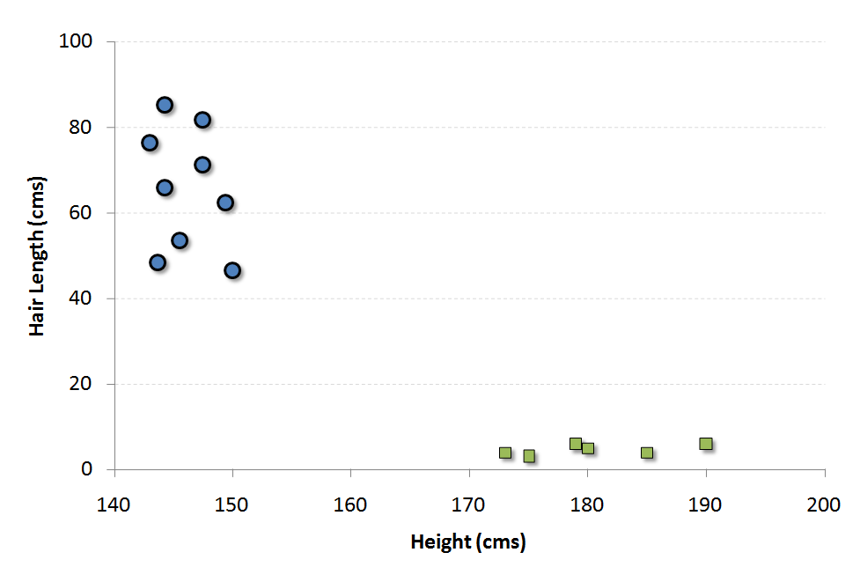
Y is never a negative in logistic, but it can be in linear regression. Y value is always between 0 and 1.

## Support Vector Machine

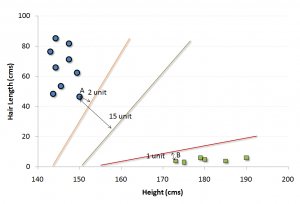
SVM is for linearly separable binary sets.

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we’d first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as **Support Vectors**)

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/SVM1.png)

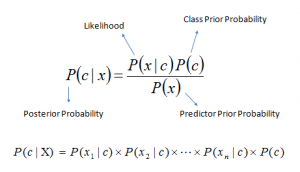
Now, we will find some *line/hyper-plane* that splits the data between the two differently classified groups of data. This will be the line such that the distances from the closest point in each of the two groups will be farthest away.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/SVM2.png)

In the example shown above, the line which splits the data into two differently classified groups is the *black* line, since the two closest points are the farthest apart from the line. This line is our classifier. Then, depending on where the testing data lands on either side of the line, that’s what class we can classify the new data as.

## Naive Bayes

It is a classification technique based on [Bayes’ theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). [](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/Bayes_rule.png)

* *P*(*c|x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
* *P*(*c*) is the prior probability of *class*.
* *P*(*x|c*) is the likelihood which is the probability of *predictor* given *class*.
* *P*(*x*) is the prior probability of *predictor*.

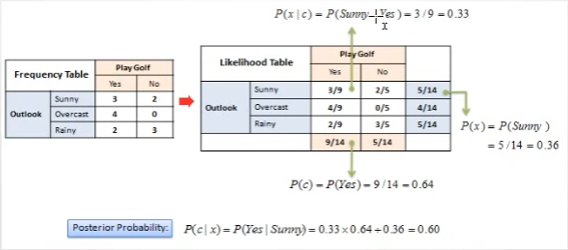
Example: Non-Numerical Predictors



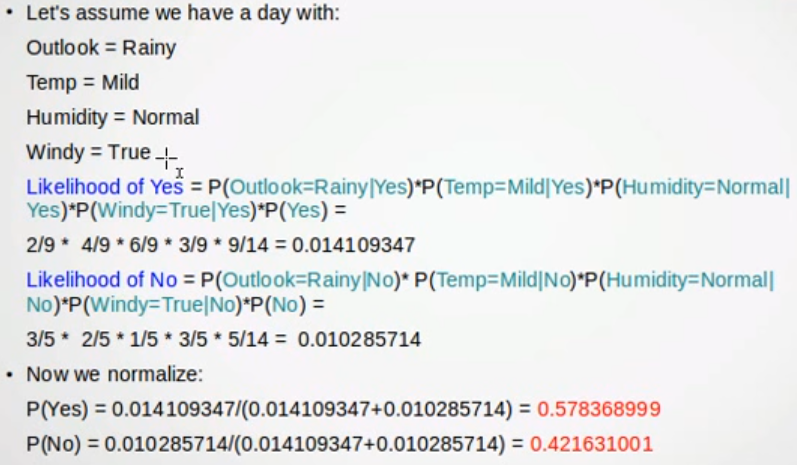
Posterior probability can be calculated by first, constructing frequency table for each attribute against the target.



Then transforming the frequency tables to likelihood tables and finally using the above equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction.

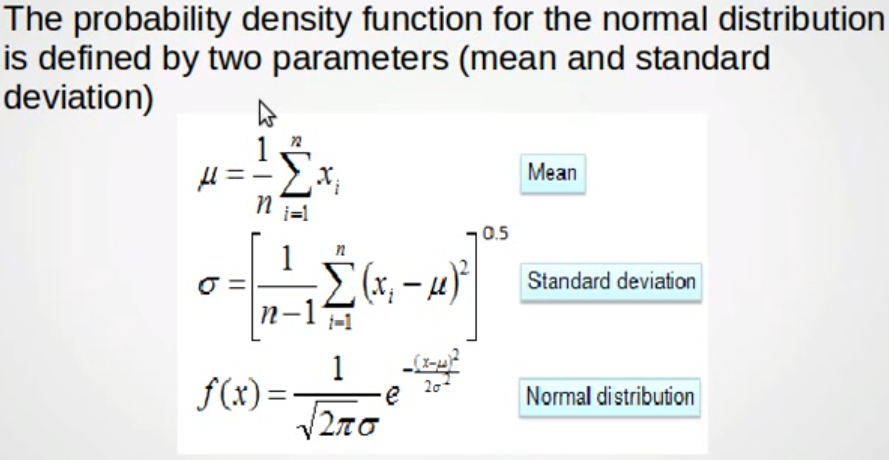


To find the probability of a particular condition

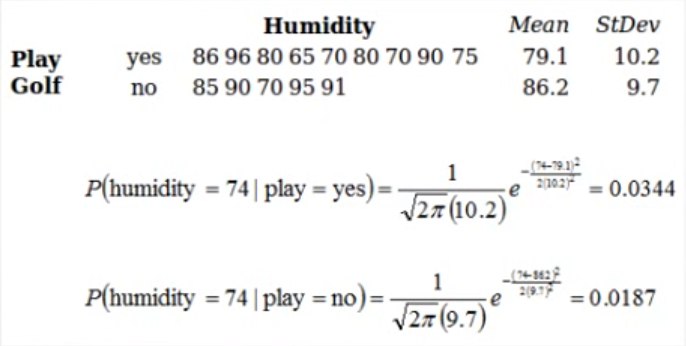


Numerical Predictors

Numerical variables needs to be transformed to their categorical counterparts (called binning) before constructing their frequency tables. Else the other option that we have is to use distribution of the numerical variable (like normal distribution)



Probability of likelihood of playing on humidity 74, is calculated as below,



\*\* Zero Frequency Problem – if an attribute value in the frequency table is 0, then add 1 to each of the values, to make it non-zero.

## Decision Trees (CART)

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.

### Classification Tress

#### How to Split using ID3 Algorithm

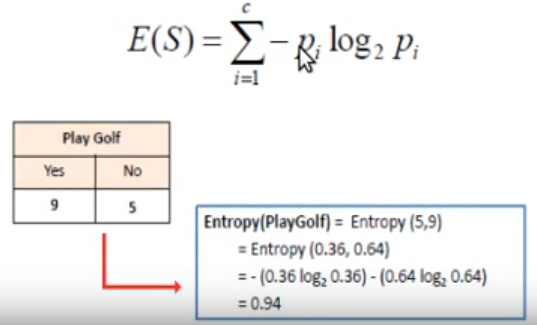
ID3 – using entropy and information gain. If the sample is completely homogeneous the entropy is 0, and if the sample is equally divided then the entropy is 1.

Using the example data set

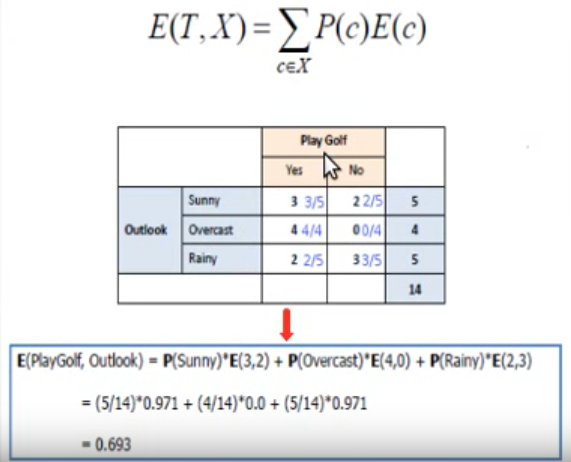


Entropy and Information Gain – Calculation

Entropy of the Target



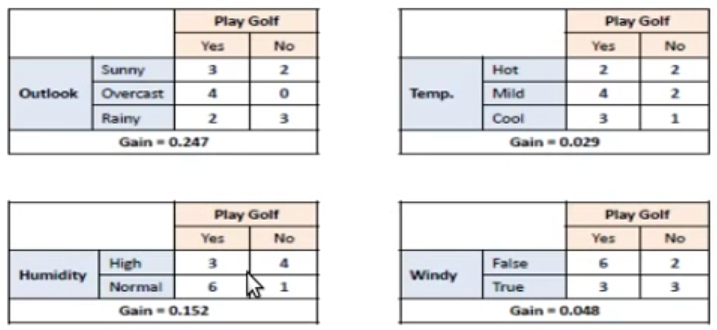
Calculate the entropy of an each predictor (e.g. Outlook)



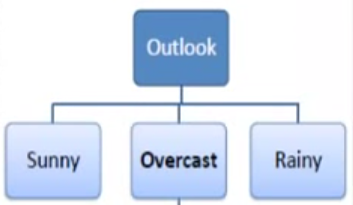
Similarly you need to calculate for rest of the predictors, then based on the information gain (below formula) will decide which predictor to be used to split.



i.e. difference of “Entropy of Target” and “Entropy of a predictor” is the information gain of a predictor. Highest information gain predictor will be used to split.



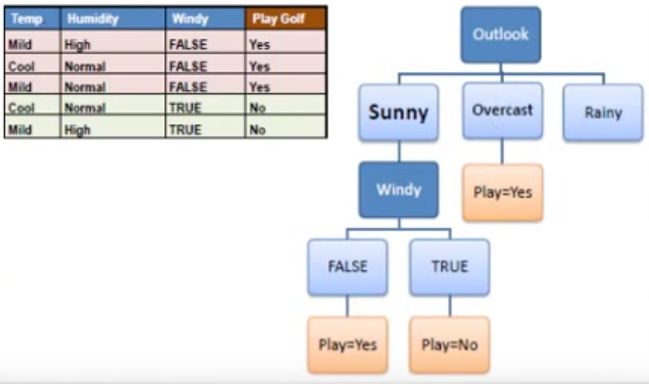
Clearly predictor outlook has the highest gain, so it is used for branching



A branch with entropy of 0 is a leaf node. Refer the complete data set table above, where set “Overcast” has all values as YES for target (Play Golf) variable. This is called completely homogenous set.



A branch with entropy more than 0 needs further splitting. The algorithm will run recursively on all the non-leaf branches, until all data is classified.



A decision tree can easily be transformed to a set of rules by mapping from root node to leaf nodes one by one.



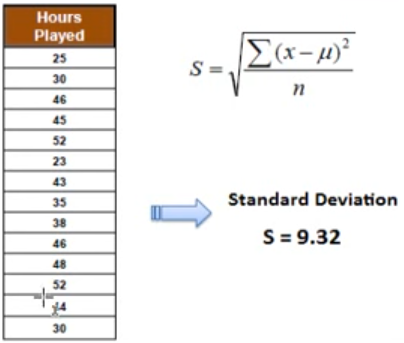
### Regression Trees



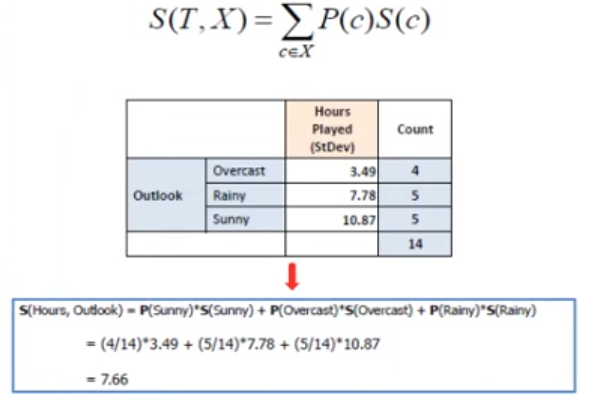
#### How to split using ID3 Algorithm

The ID3 algorithm can be used to construct a decision tree for regression by replacing information gain with standard deviation reduction. Means standard deviation is used to calculate the homogeneity of the node, if the standard deviation is zero then it is leaf node.

Standard deviation for target variable



Standard deviation for each predictor

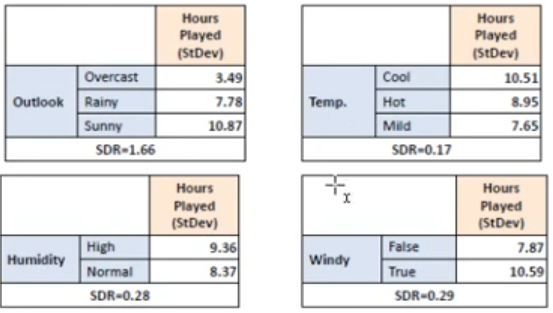


S(c) is standard deviation of all the target-variable values for that class. For example,

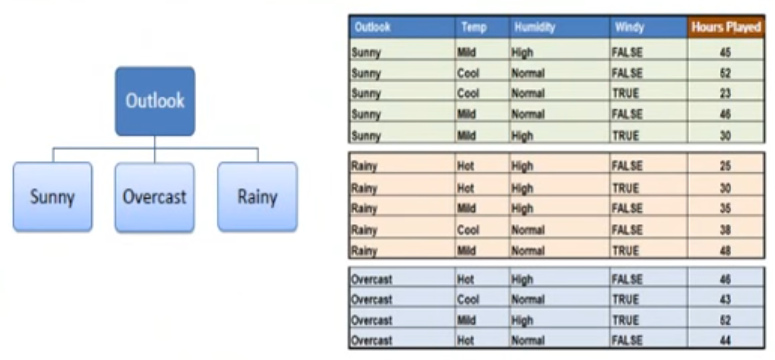
|  |  |
| --- | --- |
| Outlook | Hours Played |
| Overcast | 48 |
| Overcast | 43 |
| Overcast | 62 |
| Overcast | 44 |

Standard deviation is calculated for other predictors with respect to target i.e. S(T, X).

The difference between standard deviation of target with predictor is “Standard Deviation Reduction”. The predictor with highest SDR, will be used to split i.e. as branch node.



Now the dataset is divided based on the values of the selected attribute



A branch set with standard deviation more than 0 needs further splitting. The algorithm runs recursively on all non-leaf nodes until below mentioned criteria is met.

The termination (leaf node) criteria is, when the standard deviation for the branch becomes smaller than 5% of the standard deviation for the full dataset OR when too few instances remain in the branch (eg. 3).

When number of instances is more than 1 at leaf node, then we calculate average as the final value for the target.

### Regression Trees vs Classification Trees

* Regression trees are used when dependent variable is continuous. Classification trees are used when dependent variable is categorical.
* In case of regression tree, the value obtained by terminal nodes in the training data is the mean response of observation falling in that region. Thus, if an unseen data observation falls in that region, we’ll make its prediction with mean value.
* In case of classification tree, the value (class) obtained by terminal node in the training data is the mode of observations falling in that region. Thus, if an unseen data observation falls in that region, we’ll make its prediction with mode value.
* Both the trees divide the predictor space (independent variables) into distinct and non-overlapping regions. For the sake of simplicity, you can think of these regions as high dimensional boxes or boxes.
* Both the trees follow a top-down greedy approach known as recursive binary splitting. We call it as ‘top-down’ because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree. It is known as ‘greedy’ because, the algorithm cares (looks for best variable available) about only the current split, and not about future splits which will lead to a better tree.
* This splitting process is continued until a user defined stopping criteria is reached. For example: we can tell the algorithm to stop once the number of observations per node becomes less than 50.
* In both the cases, the splitting process results in fully grown trees until the stopping criteria is reached. But, the fully grown tree is likely to overfit data, leading to poor accuracy on unseen data. This bring ‘pruning’. Pruning is one of the technique used tackle overfitting. We’ll learn more about it in following section.

## Random Forest (Ensemble)

Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we’ve collection of decision trees (so known as “Forest”). To classify a new object based on attributes, each tree gives a classification and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

## KNN (K- Nearest Neighbors)

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. KNN is just a generalization of the nearest neighbor algorithm. Instead of considering the nearest neighbor, you consider k-nearest neighbors from a dataset containing n data points — k defines how many nearest neighbors will have an influence on the classification process.

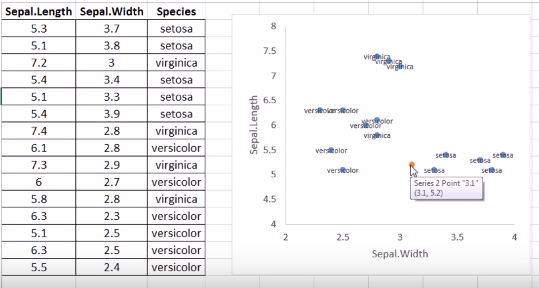
These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables.

Example:

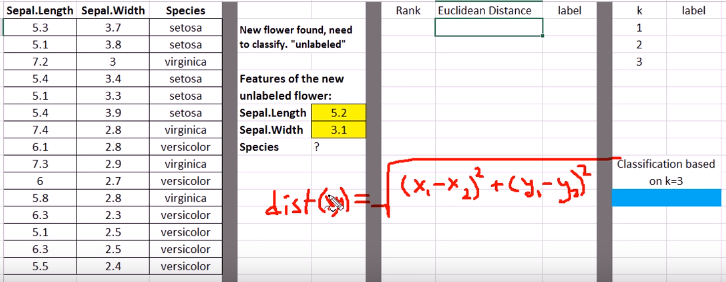
We have a set of labeled data points, now we need to classify the unlabeled data point to one of them.

Let’s plot them, as we have only 2 dependent variables, they become X and Y axis.

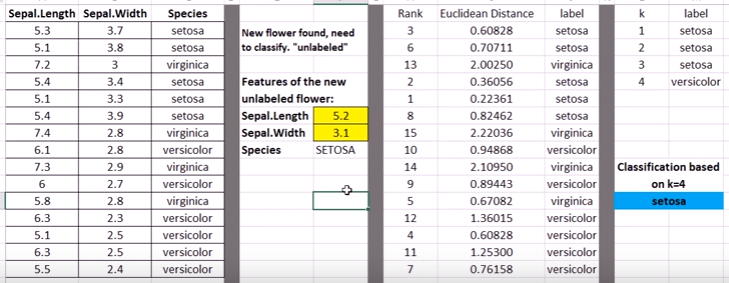
The one in the orange is the unlabeled data point.



Next we have to find the distance between that unlabeled data point with each labeled data point. Will use Euclidean Distance for that.



Once the distance calculated, then we have rank them in ascending order, so that nearest had the highest rank.



Based on the K, we gonna find the majority of labeled data point, to which the unlabeled point has to be classified.

K is calculated based on the squareroot of number of samples, in this example for the 15 samples, square root is between 3 and 4, and so K can be either 3 or 4.

So the answer is SETSOA.

## K-Means (Clustering)

To subdivide data points of a dataset into clusters based on nearest mean values. To determine the optimal division of your data points into clusters, such that the distance between points in each cluster is minimized, you can use k-means clustering. In the term k-means, k denotes the number of clusters in the data. Since the k-means algorithm doesn’t determine this, you’re required to specify this quantity.

The k-means algorithm works by placing sample cluster centers on an n-dimensional plot and then evaluating whether moving them in any one direction would result in a new center with higher density — with more data points closer to it, in other words. The centers are moved from regions of lower density to regions of higher density until all centers are within a region of local maximum density — a true center of the cluster, where each cluster gets a maximum number of points closest to its cluster center.

One weakness of the k-means algorithm is that it may produce incorrect results by placing cluster centers in areas of local minimum density. This happens when centers get lost in low-density regions — in other words, regions of the plot that have relatively few points plotted in them — and the algorithm driven directional movement — the movement that’s meant to increase point density — starts to bounce and oscillate between faraway clusters. In these cases, the center gets caught in a low-density space that’s located between two high-point density zones. This results in erroneous clusters based around centers that converge in areas of low, local minimum density.

Kernel density estimation (KDE) is just such a smoothing method; it works by placing a kernel — a weighting function that is useful for quantifying density — on each data point in the data set and then summing the kernels to generate a kernel density estimate for the overall region. Areas of greater point density will sum out with greater kernel density, while areas of lower point density will sum out with less kernel density.

Hierarchical and Nearest Neighborhood Methods – resource intensive

Decision Trees and Random Forest

Average nearest neighbor algorithms are used in pattern recognition – classification

## What Algorithm to Use



