**Packages to be known for Machine learning**

1. Numpy
2. Pandas
3. Scipy
4. Scikit-learn
5. Matplotlib
6. Seaborn

Scikit-learn :

* Lots of machine learning libraries.
* In built classification, Regression and clustering algorithms
* Build with flexibility to use Numpy and Scipy which helps easy to play with data.
* Most preprocessing tasks are inbuilt in scikit-learn such as Data preprocessing, Future Selection, extraction and train – test splitting, algorithm, prediction, evaluation and exploring the model.

**Supervised vs Unsupervised learning:**

**Supervised learning:**

Train the model with labelled dataset.

1. Classification – process of predicting discrete class label or category.

Eg. Predicting the class such as finding the cancer in cells of human, customer churn in telecom.

1. Regression – process of predicting a continuous value.

Eg. Predicting the house price, stock price.

**Unsupervised learning:**

We do not supervise the model, but the model trains on its own and discover information that are invisible to human eye.

1. Dimension reduction – reducing the redundant future to make classification easier.
2. Density estimation – explore data to find substructure
3. Market basket analysis or Association – based on theory of if a customer buys certain product, then he is likely to buy another product as well.
4. Clustering – grouping of data points that are similar somehow. Eg. Segment customer based on credit score in banking. Used for Discovering structure, summarization and Anomaly detection.

**Few more Machine learning techniques:**

Anomaly detection – discovering abnormal cases eg. Credit card fraud detection

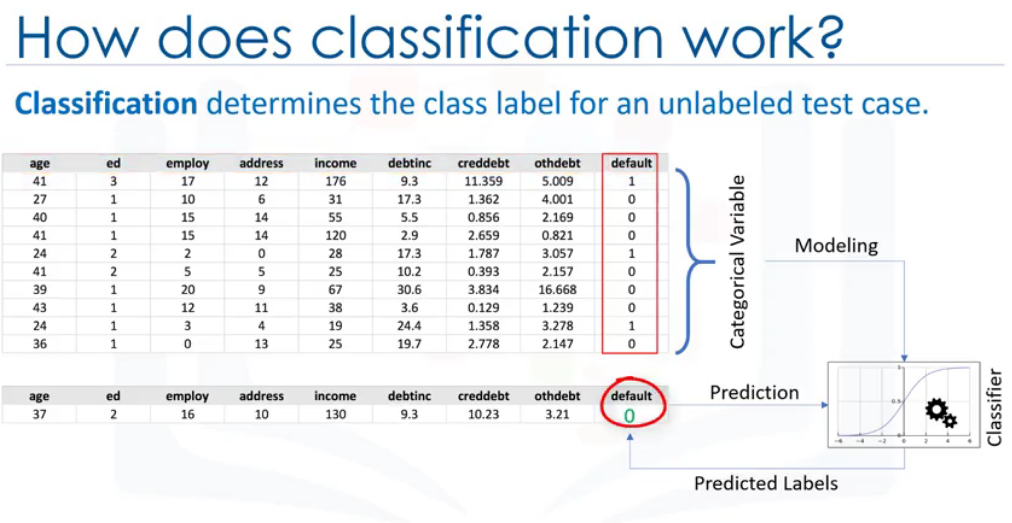
Sequence mining – predict the next occurrence eg. Click stream

Recommender systems – Recommending items eg Netflix, youtube.

**Classification**:

* Supervised learning approach.
* Categorizing some unknown items to a discrete set of groups or classes.
* Target attribute is a categorical variable with discrete value.

Eg. Loan default prediction, customer category, customer churn in telecom, customer response to particular advertisements.



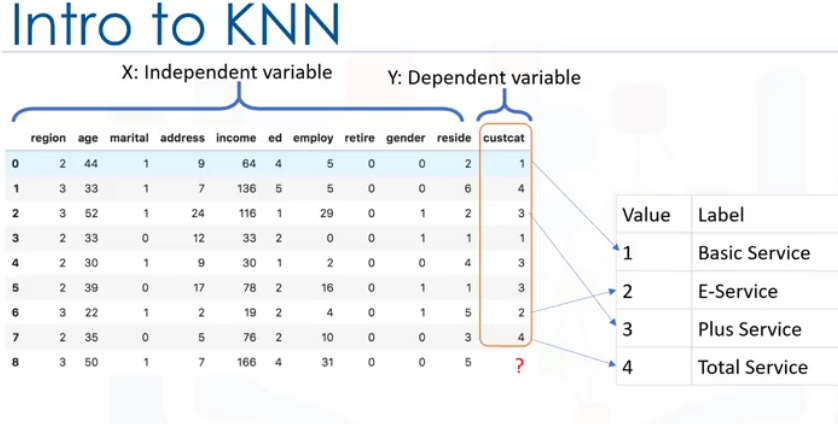
Above screenshot is a binary classifier example.

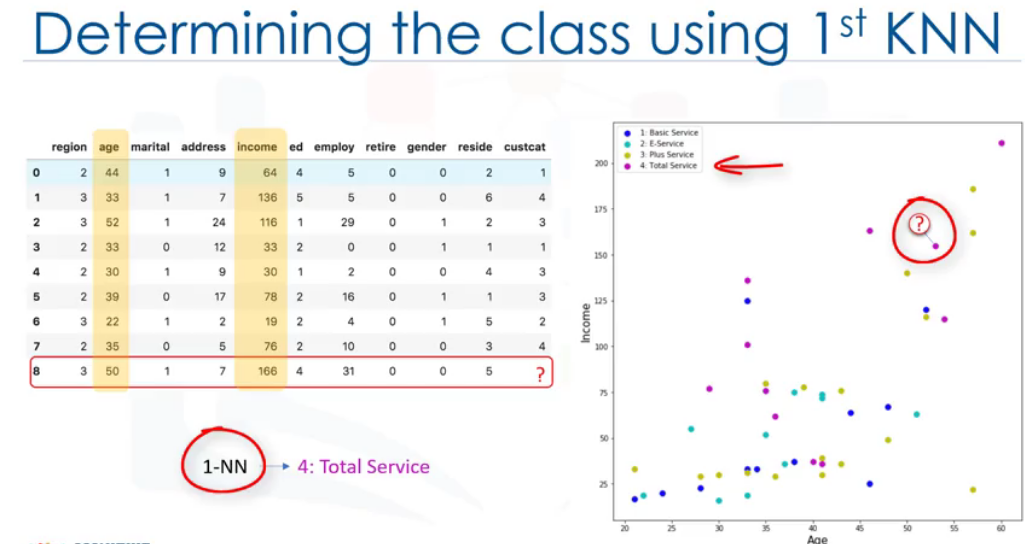
**Types of Classification algorithm:**

* KNN algorithm
* Decision tree
* Logistic regression
* SVM (Support vector machine)
* Naïve Bayes
* Neural Networks

**K-Nearest Neighbor:**

Finds the nearest neighbor and get its label from the neighbor group.



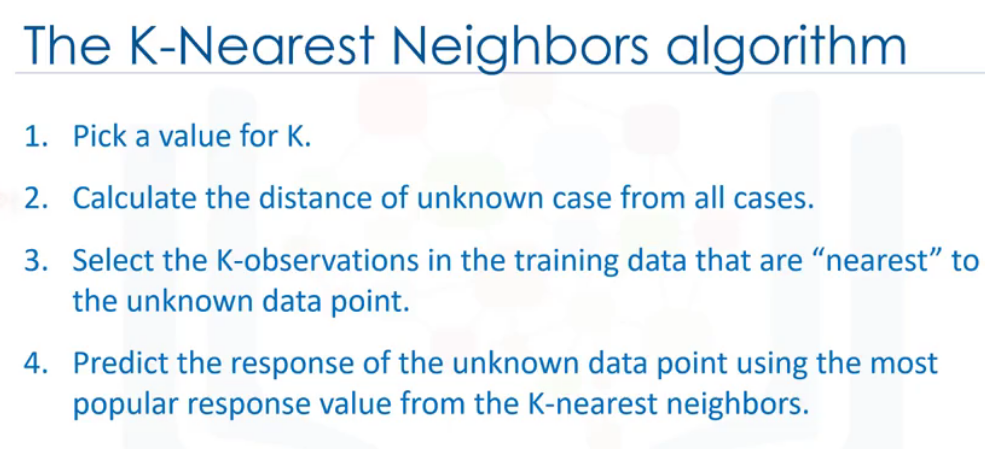


**Determining the neighbor using first level may be sometimes wrong. Above screenshot is an example of such case. Doing so and assigning Total service label may result in wrong classification since the point may be an outlier.**

**To overcome, use higher value of K.**

* Method of classifying cases based on the similarity to other cases.
* Cases that are surrounded are called neighbors.
* Similar cases with same class labels are near each other. (Distance between two cases is measure of dissimilarity, calculated using Euclidian distance)

**Algorithm:**



**Euclidian Distance**: root of (x1 -x2)^2

**Choosing right K value:**

Low value of K will result in highly complex model which may be lead to overfitting.

Prediction for out of sample cases is not generalized, if there is very small k value.

Overfitting is always bad.

Higher value of k result in more generalization. (k=25).

General solution: Reserve part of your data to calculation of the accuracy.

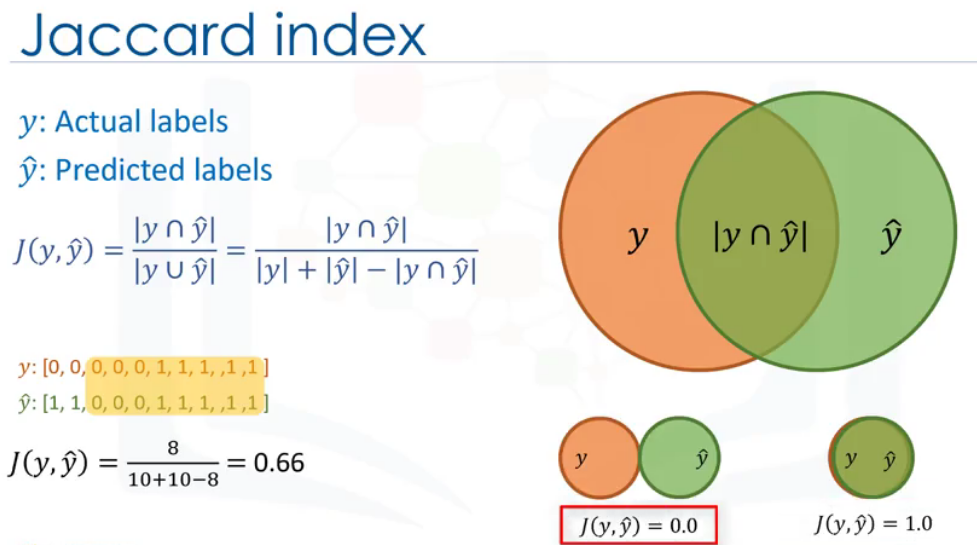
Repeat the model by increasing K value to achieve the better accuracy and K value.

KNN can also be used for **regression** technique (by forming a neighbor and calculate the median of the neighbor).

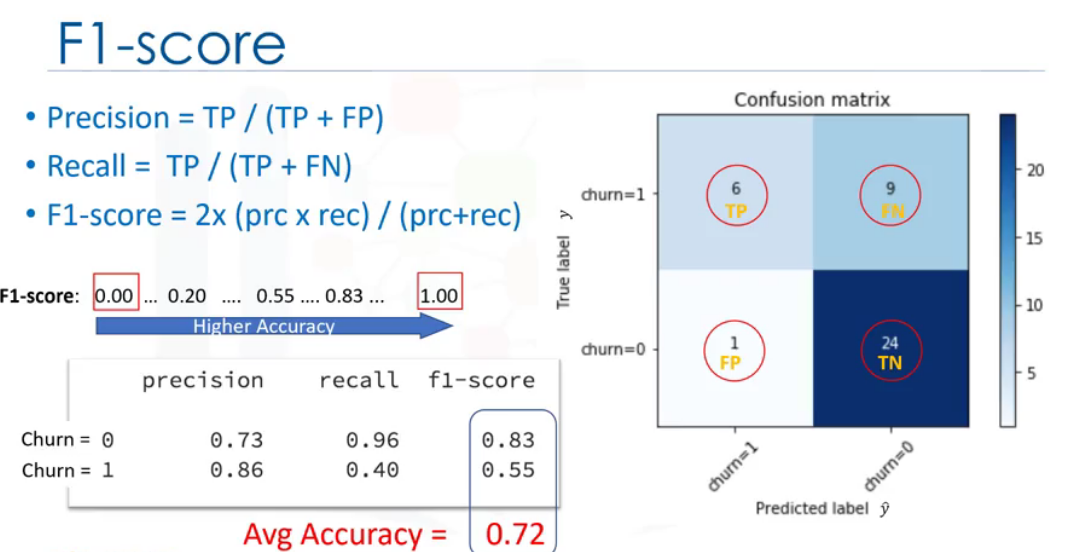
**Evaluation metrices for classification models:**

* **Jaccard Index**
* **F1 score**
* **Log loss**

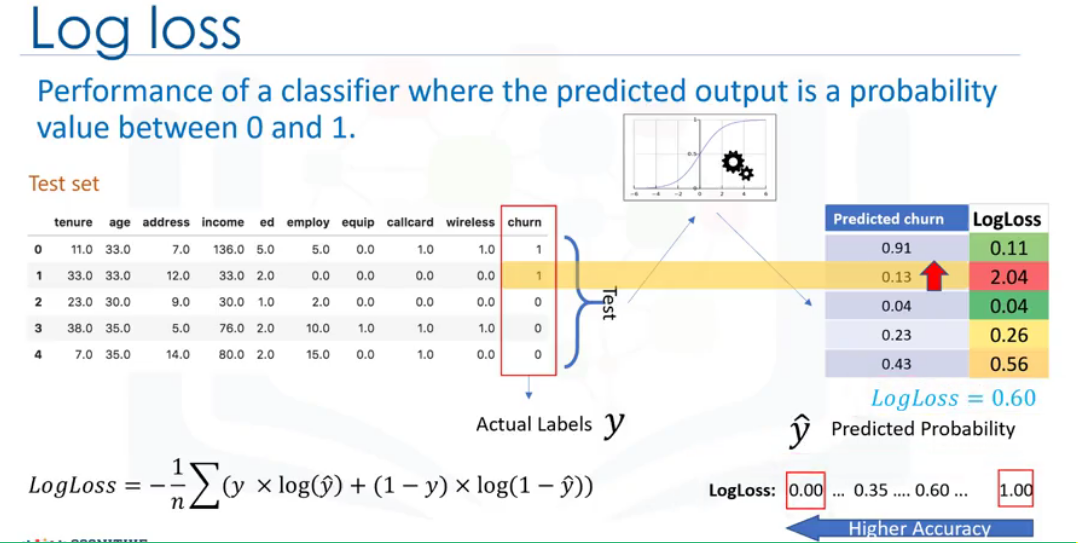
**Jaccard Index:**



**F1- score (Using confusion matrix)**

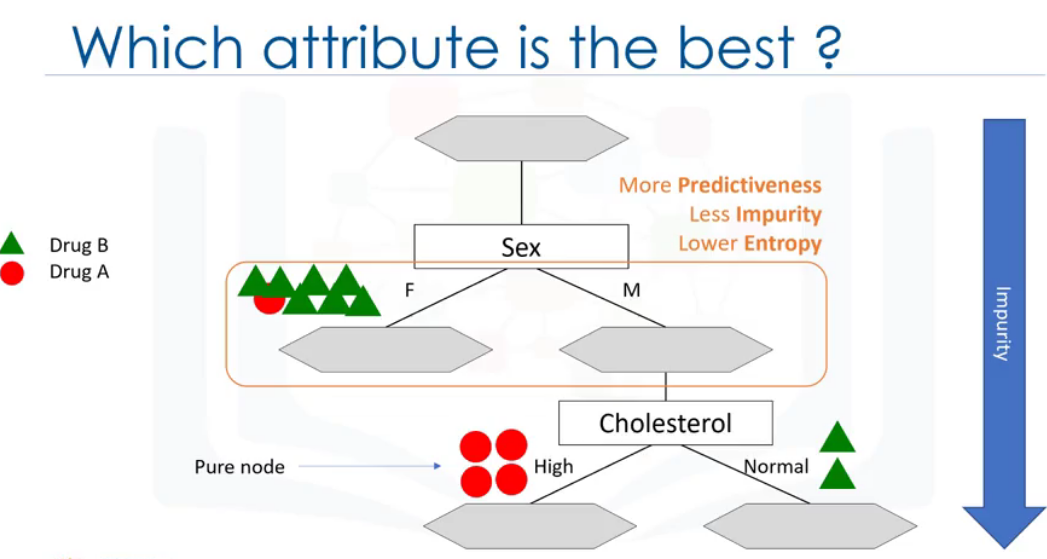


**Log loss**



**Decision Tree**

* Using recursive partitioning to classify the data.
* Determine which attribute is the best to split the data and classify them.

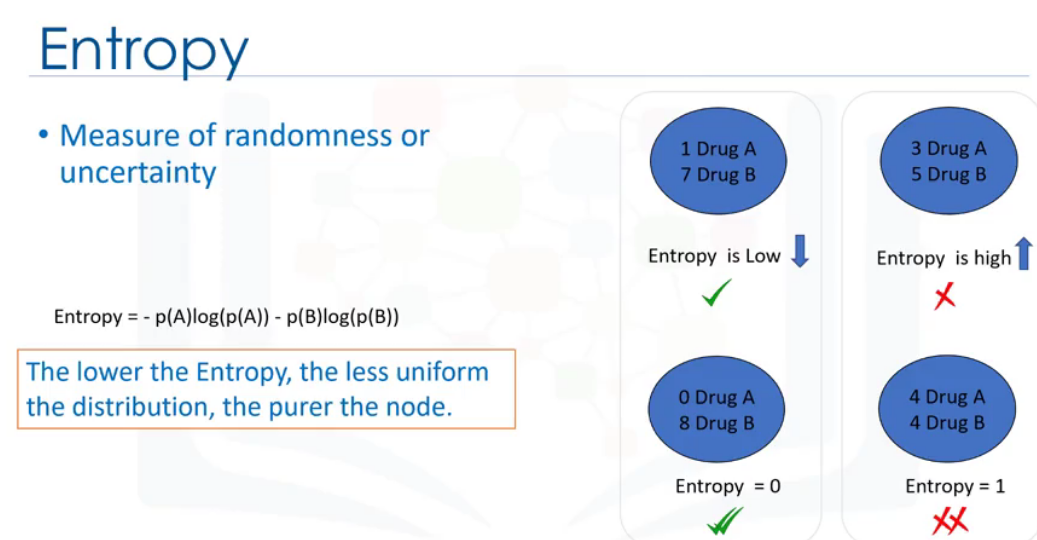


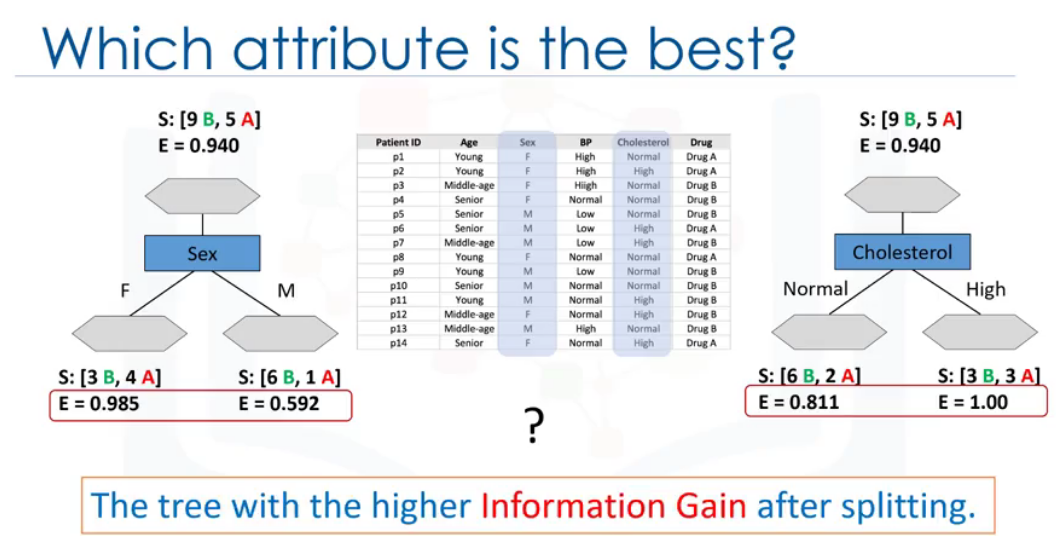
This approach uses recursive partitioning to minimize the impurities of the data. A node is considered as pure if 100% of cases fall in specific category of target field.

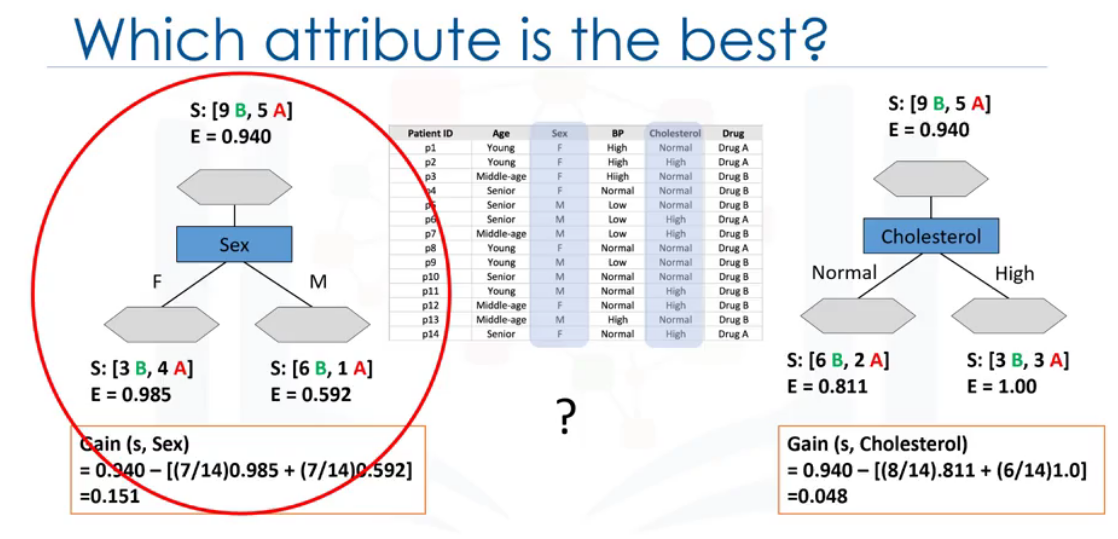
Impurity of node is calculated by entropy of the data. Entropy is measure of randomness or uncertainty.

Lower entropy, less the uniform distribution, the purer the node.

There are inbuilt packages to calculate the entropy.



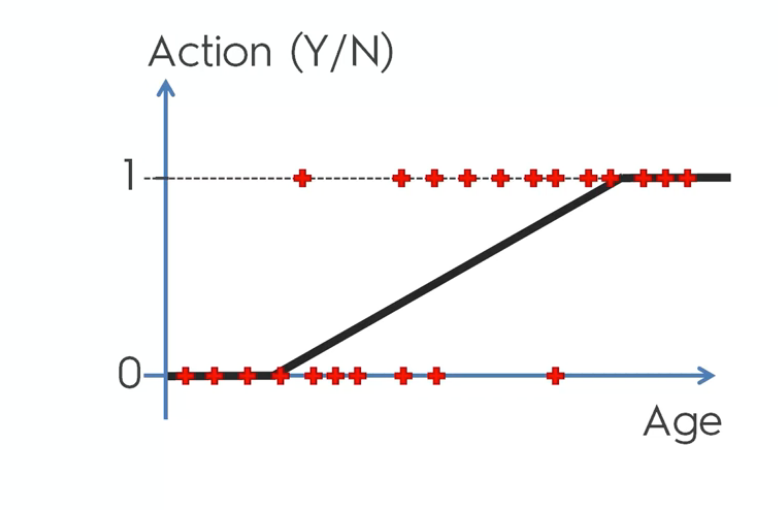




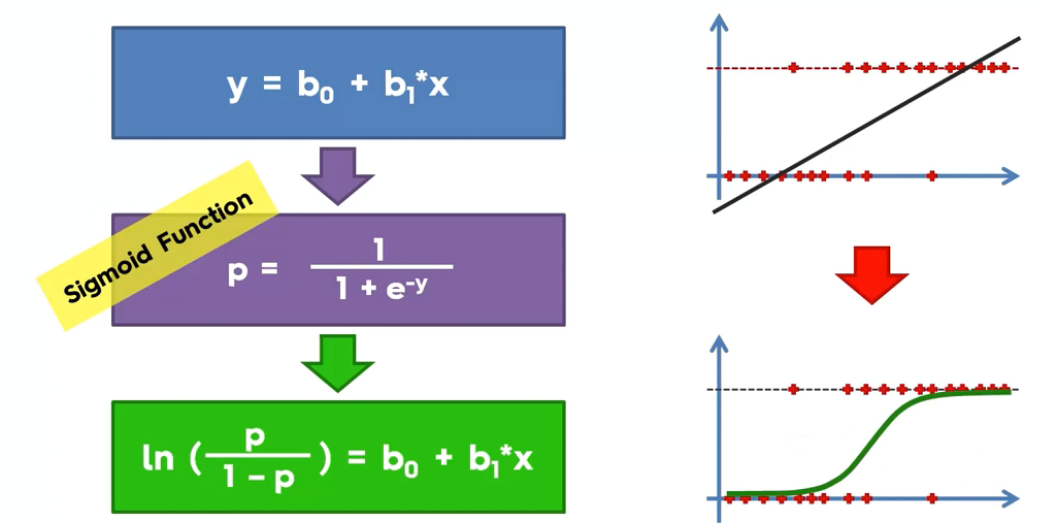
**Logistic Regression:**

**Use Case :**  super markets send emails to customers regarding recent offers and find the customer who respond to the offer and who didn’t using logistic regression with binary value of 1 being customer responded and 0 being customer who didn’t.

Predict the probability of whether the user responds to the offer.



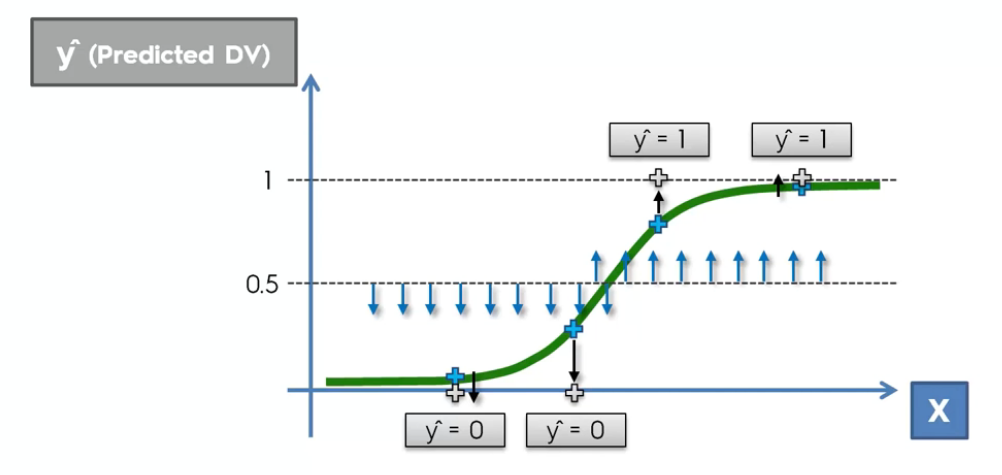
**Scientific Approach:**



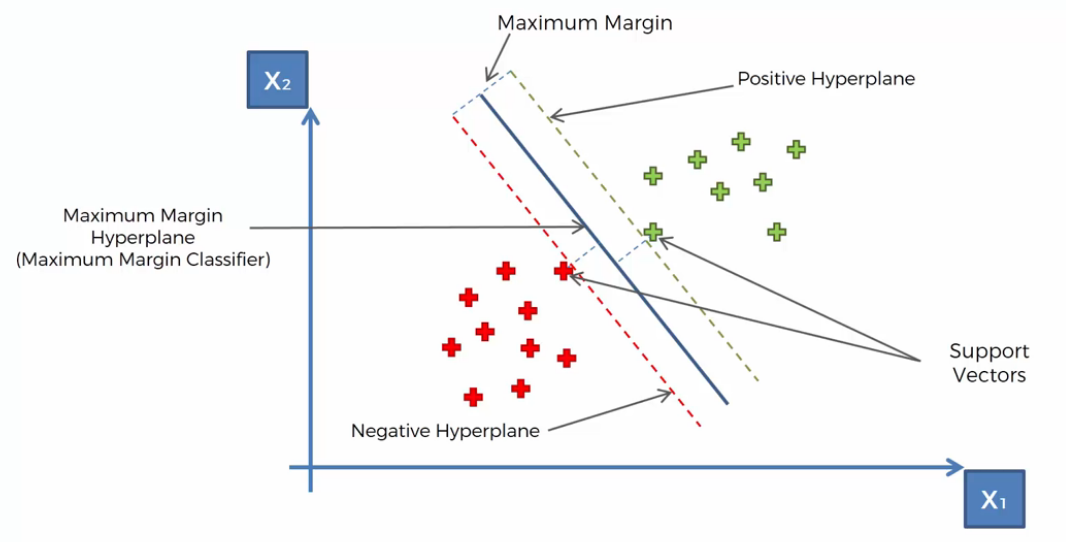


Use probability to make a score.

Probability less than 0.5 is considered as 0 and probability greater than 0.5 is considered as 1.



**SVM Classification:**



Find the best fit line or decision boundaries to classify the data points.

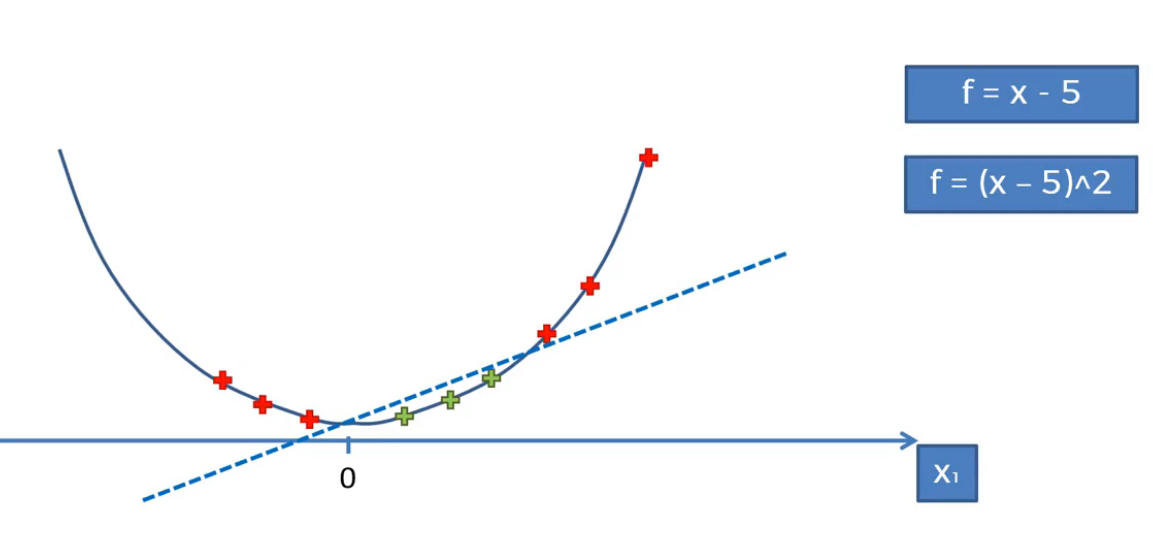
Line is searched through maximum margin which separates two classes.

Sum of the support vector points are to be maximized. These support vector points are supporting the whole algorithm.

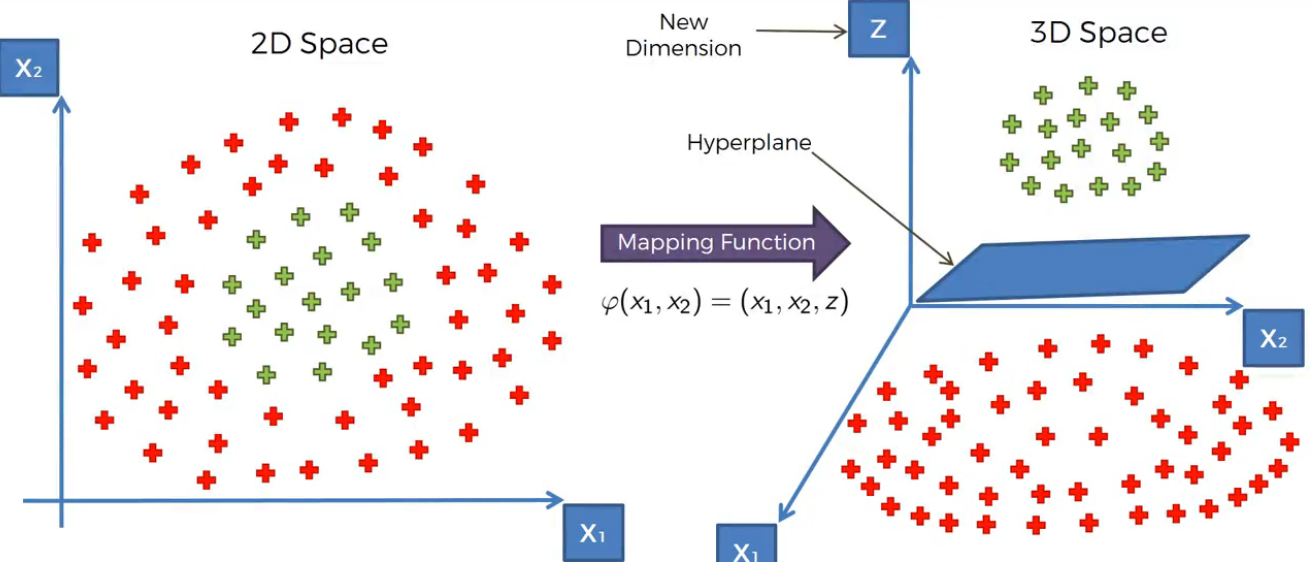
**Kernal SVM:**

Useful when data are non-linearly separable. SVM are used in linear separable cases.

* Map the non-linear data set in high dimension. Eg given below



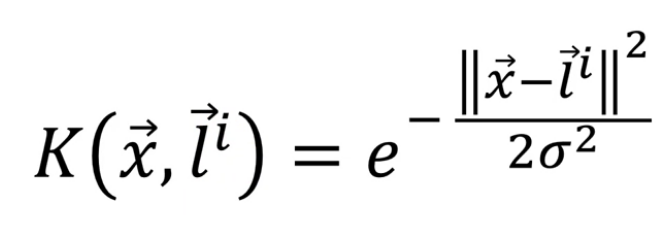
* Non-linear data in 2 dimensional space.



But the drawback Is mapping higher dimension is highly compute-intensive. Kernal trick help us to resolve the issue.

Kernal Tricks:

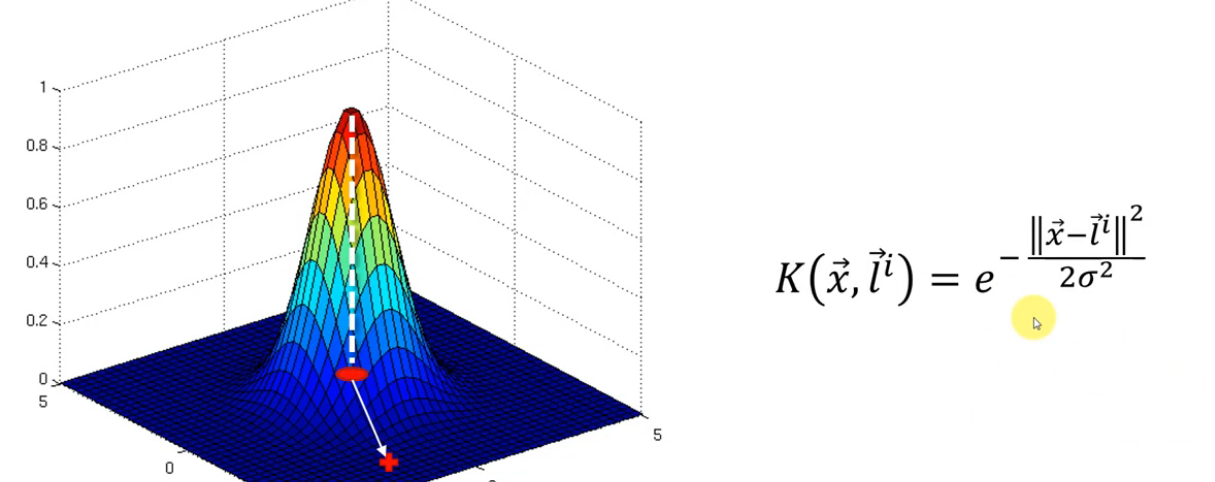
* Gaussian RBF (Radial base function) kernel



K = kernel ; x= data point ; l = landmark.

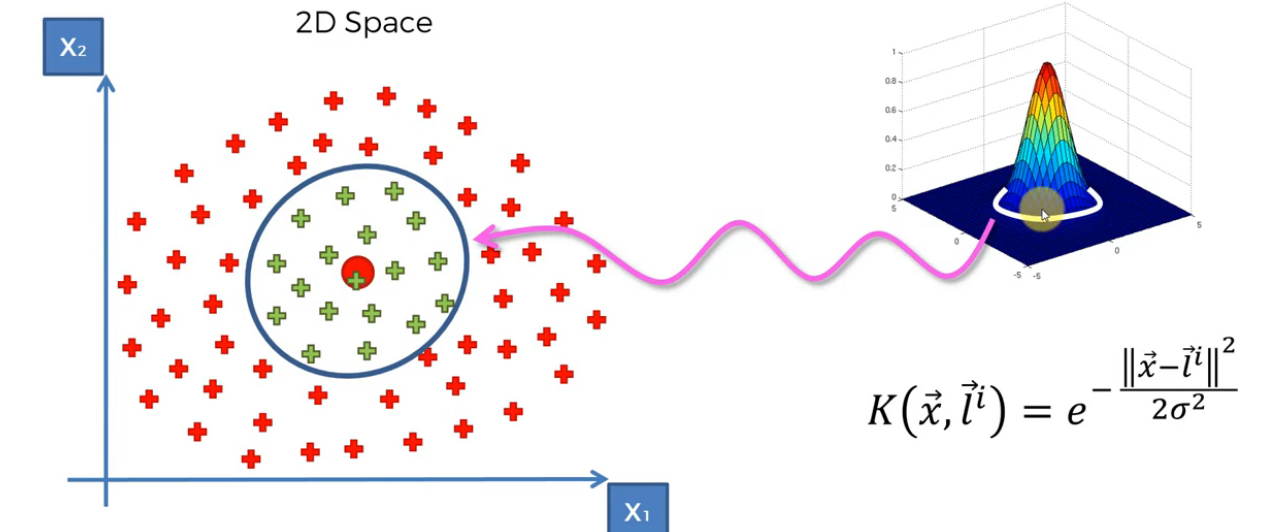
Find the distance between the landmark and the datapoint using the above formula. Here, for the below figure, there will be e raise to power on negative of some large number which will be close to 0.

**Above inference says, if the datapoint is far away from the landmark, then you get close to 0 in vertical axis.**

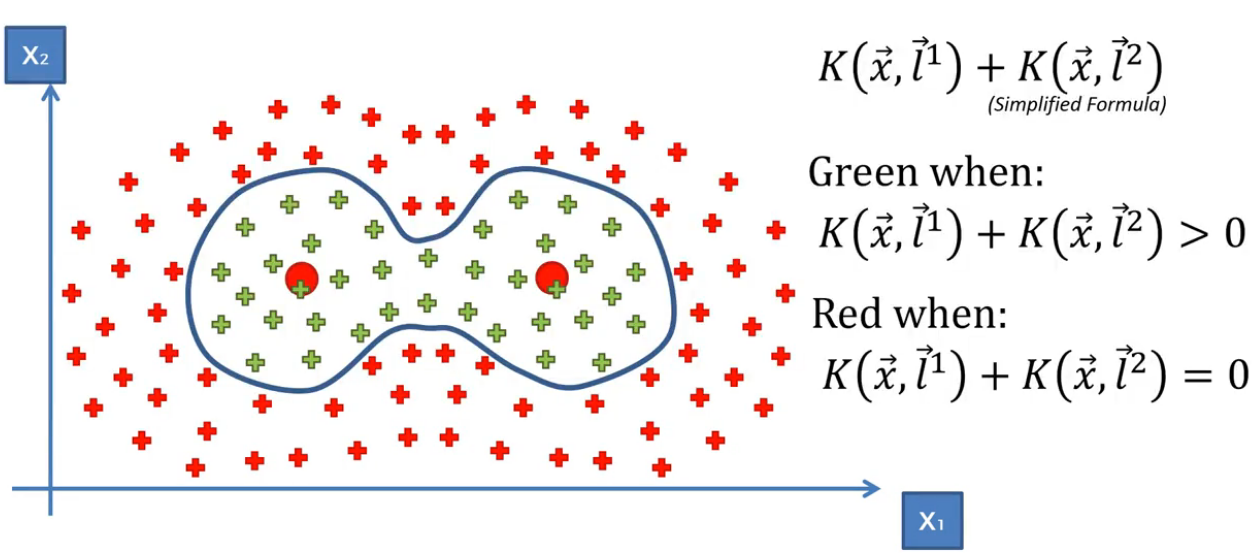


**Alternatively, if the datapoint is close to landmark then you will get e to power of 0 i.e will be 1.**

This kernel function is used to separate our dataset.

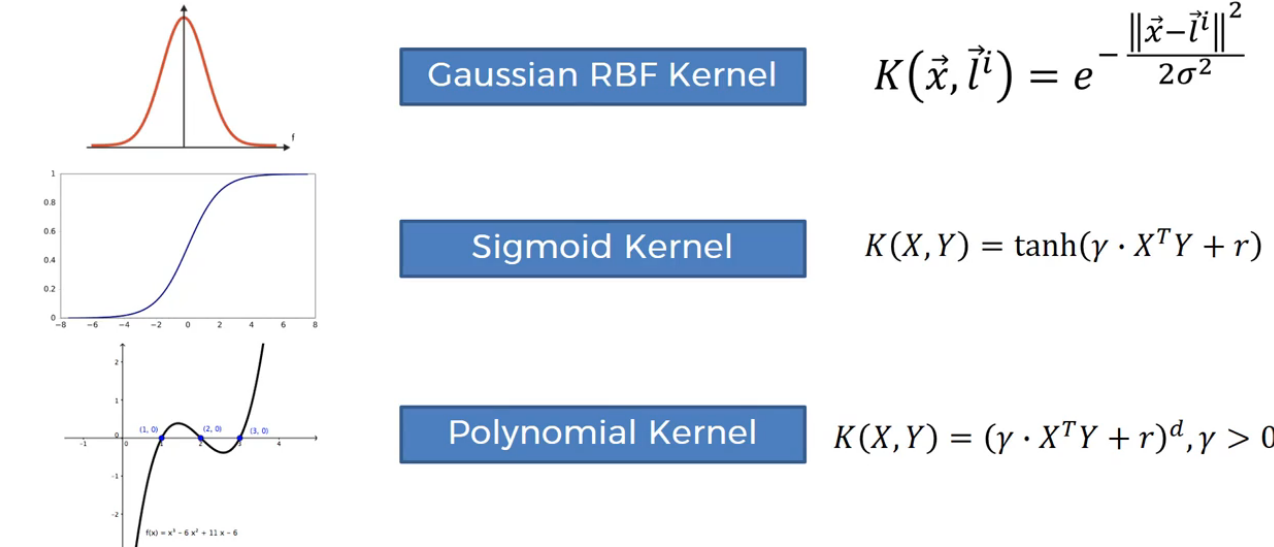


Sigma defines the circumference of the circle in the hyperplane. Increase in sigma will increase the circumference and decrease will shrink the circumference.



**Types of kernel functions:**

* **Gaussian Kernel**
* **Sigmoid kernel**
* **Polynomial kernel**

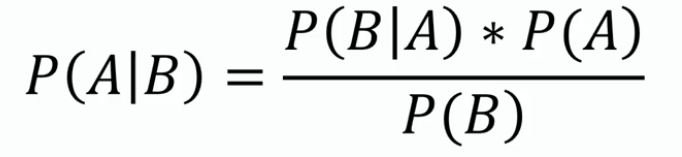


To understand more: <https://mlkernels.readthedocs.io/en/latest/kernels.html>

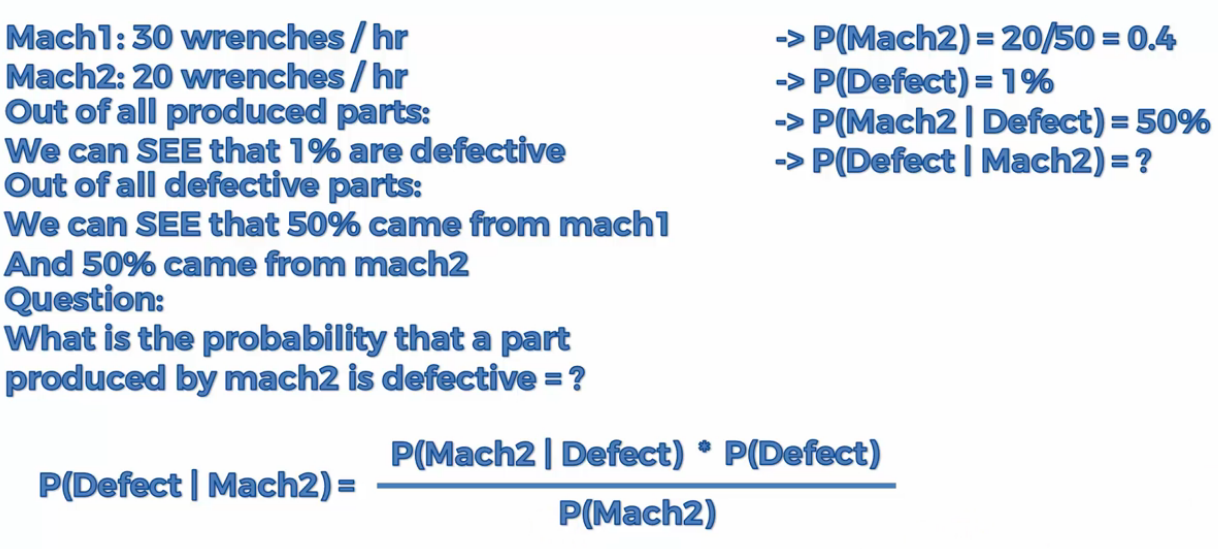
**Naïve Bayes Classification:**

Bayes theorem does independent assumption.

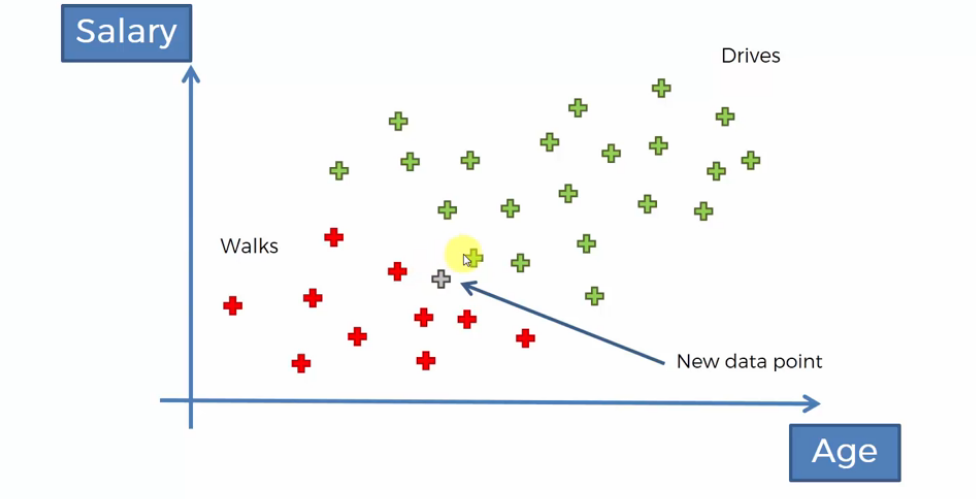
**Bayes Theorem:**



Example of Bayes theorem,



**Naïve Based classifier:**



**Implement Bayes theorem twice to the new data point, first with walk and then with drives.**

P(Walks| x) = P(x|Walks) \* P(Walks)/ P(x)

P(Drives| x) = P(x|Drives) \* P(Drives)/ P(x)

**Steps 1:**

1. Calculate Prior probability i.e P(Walks)
2. Calculate marginal probability P(x)
3. Calculate likelihood i.e P(x | Walks)
4. Calculate Posterior probability I.e P(Walks | X)

**Step 2:**

1. Calculate Prior probability i.e P(Drives)
2. Calculate marginal probability P(x)
3. Calculate likelihood i.e P(x | Drivess)
4. Calculate Posterior probability I.e P(Drives | X)

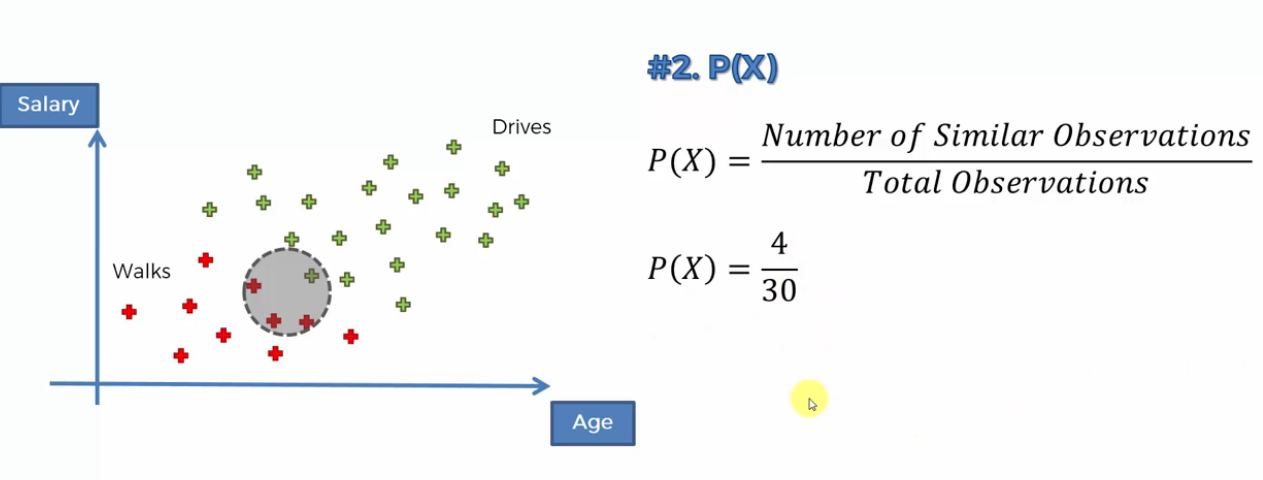
**Step 3:**

Compare both the probability and assign the one with higher probability,

Marginal probability:

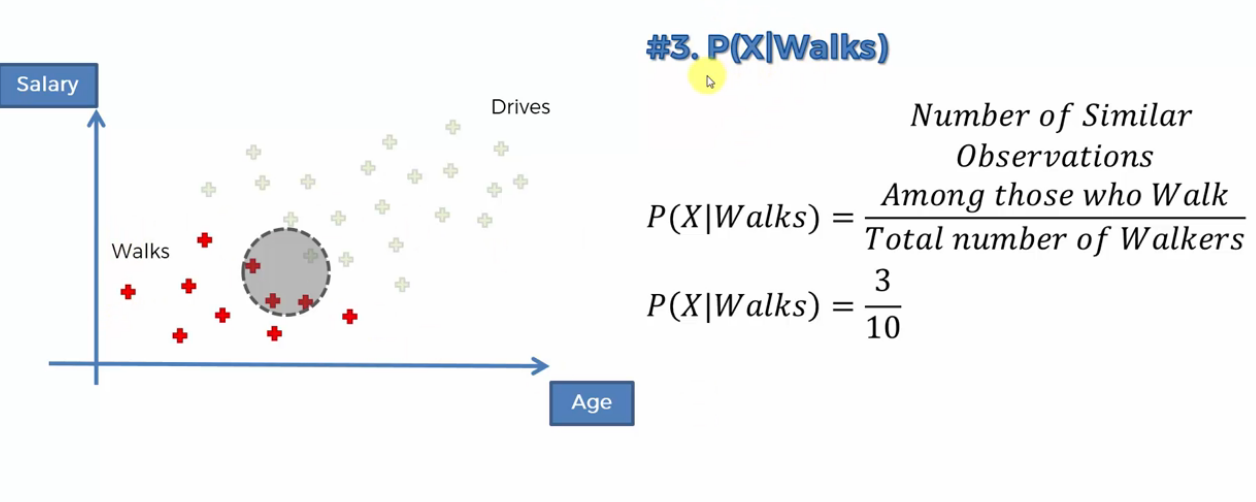
P(X) is the likelihood of what is the probability that the newly added datapoint fall in the circle. i.e which exhibits the behavior.

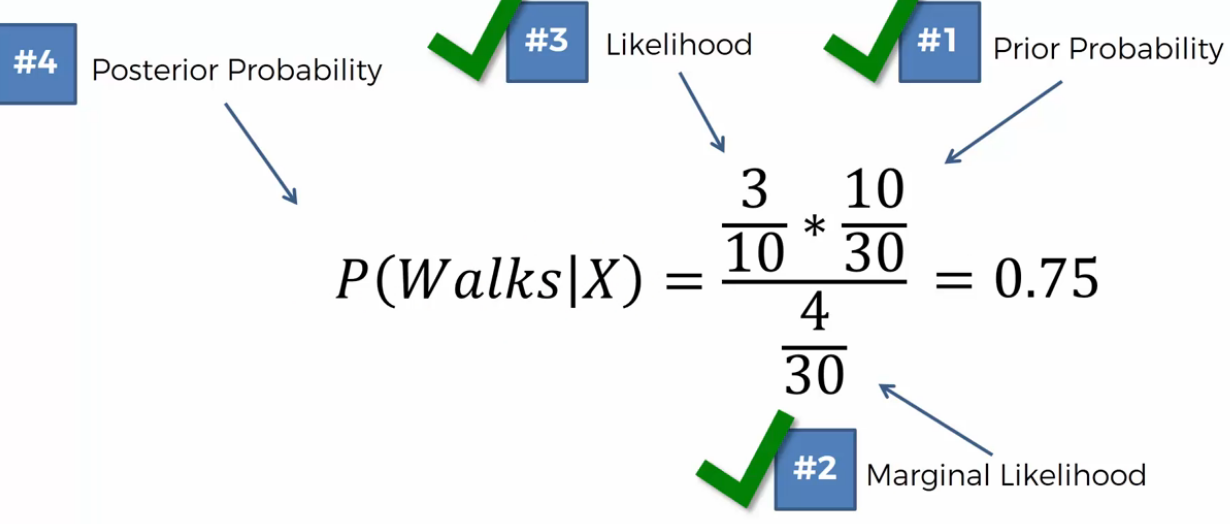
To compute marginal probability, plot a circle with radius which will be the input in algorithm.



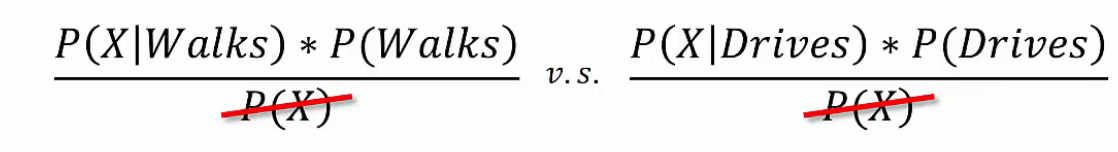
Likelihood:

P(x | Walks) : What is the probability that the new datapoint fall in the circle given that he walks.





**Step 3 is modified as when you are comparing the two category.**



**Better to go with Naïve Bayes compared to logistic and SVM since they are linear.**

# References

Aghabozorgi, S. (n.d.). *coursera.* Retrieved from www.coursera.org: https://www.coursera.org/learn/machine-learning-with-python