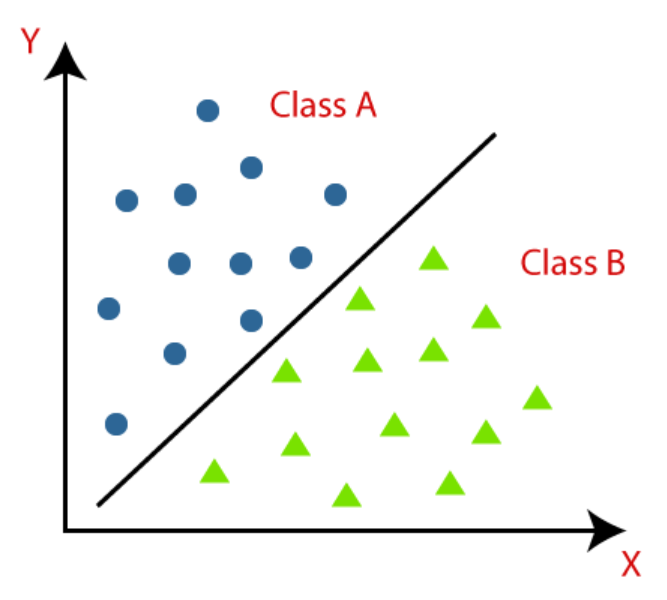
**CLASSIFICATION**

**Introduction**

* **Classes/groups:** **Yes** or **No**, **True** or **False**, **1** or **0** etc.
* **Classes** are also known as **targets**, **labels** & **categories**.
* **Classifier:** A classification algorithm.
* **Learner:** An alternative term for **ML algorithm**.



**Types of Classifiers**

* **Binary classifier:** Classifiers with **two** possible outcomes.
* **Multi-class classifier:** Classifiers with **more than two** possible outcomes.

**Types of Learners**

* **Lazy learners:** Learners which develop classification model **after** receiving both training and test dataset & it takes **less** time in **training** but **more** for **predicting**.
* **Eager learners:** Learners which develop classification model **immediately after** receiving training dataset & it takes **more** time in training but **less** in predicting.

**Types of Classification Algorithms**

Linear models:-

* Logistic regression
* Support vector machines

Non-linear models:-

* K-nearest neighbours
* Kernel SVM
* Naïve Bayes
* Decision tree classification
* Random forest classification

**Methods for Evaluating Classification Models**

* Log loss or cross-entropy Loss
* Confusion matrix
* AUC-ROC curve

**Log Loss/ Cross Entropy Loss**

* Used for **evaluating classifiers** which give **probability value** (between **1** & **0**).
* Good binary classification model gives probability value **nearby 0**.

Cross entropy for binary classifications:-

**H(y,p) = y log(p) + (1-y) log(1-p)**

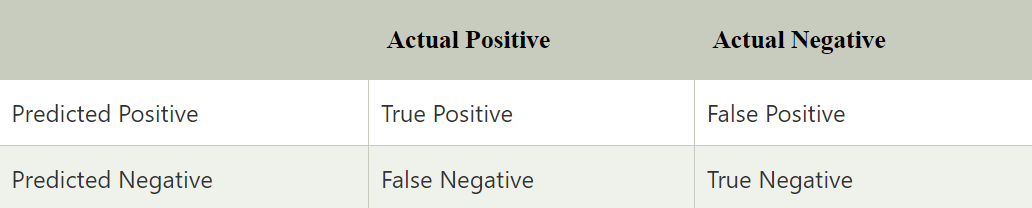
**y = Actual output**

**p = Predicted output**

**Confusion Matrix**

* Also known as **error matrix**.
* Provides a **feedback table** based on model’s performance.

**For example:**

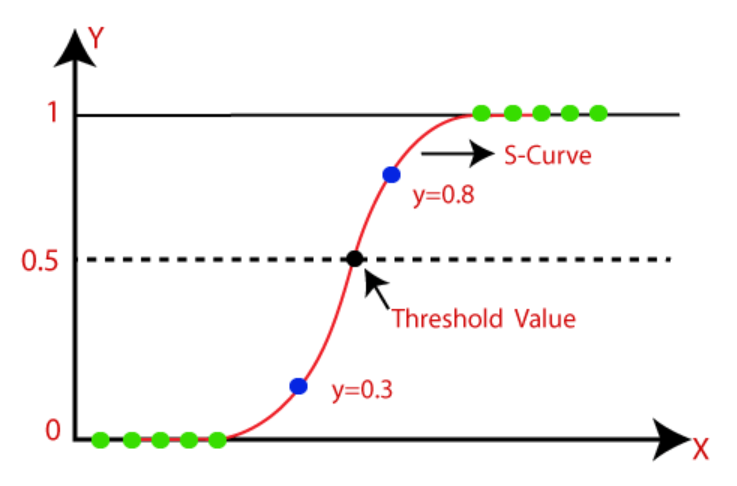


**AUC-ROC Curve**

* **AUC:** Area under the curve.
* **ROC:** Receiver operating characteristic curve.
* It is a **graph** showing performance of the model.
* The curve is plotted with **FPR** (**False positive rate**) on **x-axis** & **TPR** (**True positive rate**) on **y-axis**.

**Logistic Regression**

* A **classification** method.
* Gives a **probabilistic value** instead of **0** or **1**.
* We fit **S-shaped function** in this instead of **regression line**.
* This S-shaped function is called ***sigmoid function*** (will read more about it).

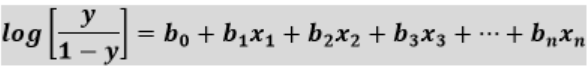


Sigmoid function:-

* A function used to **map** predicted values to **probabilities**.
* ***Sigmoid function*** is also known as ***logistic function***.
* We have a **threshold value** in it, values above it tends to **1** & below tends to **0**.

Assumptions:-

* Dependent variable is **categorical**.
* Independent variable is **not** multi-collinear.

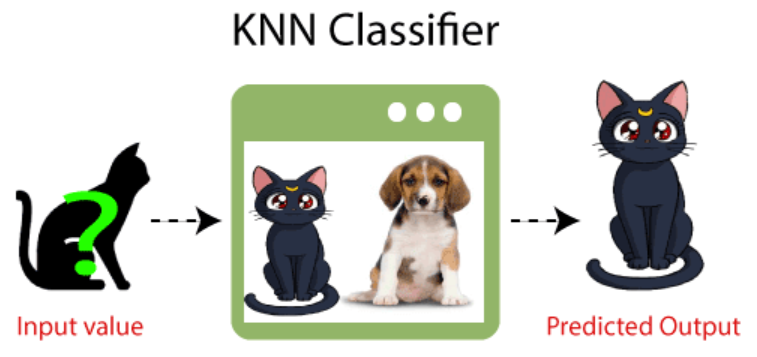


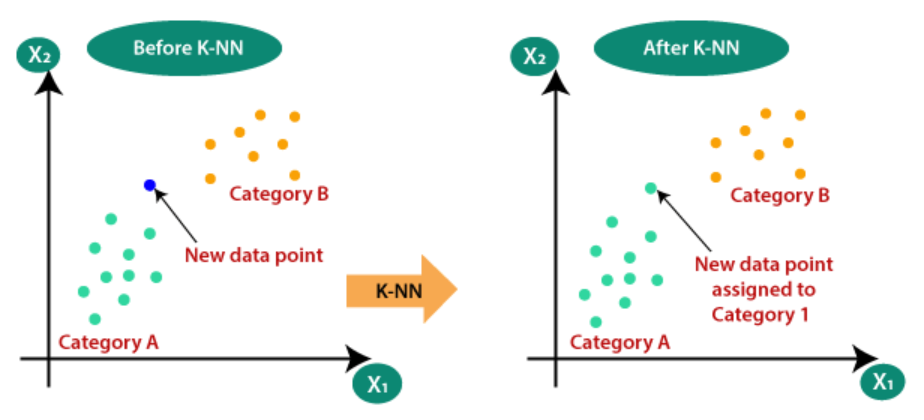
**Types of Logistic Regressions**

* **Binomial:** Two possible values.
* **Multinomial:** Multiple possible values.
* **Ordinal:** High, medium & low.

**K-Nearest Neighbour (KNN)**

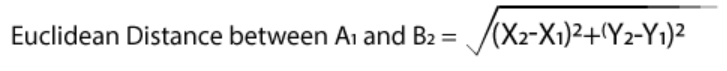
* This algorithm first **stores** the datapoint & **then** **decides** its category.
* Used in both ***regression*** & ***classification***.
* It’s a **non-parametric** & **lazy learner** algorithm.
* **Non-parametric:** **Doesn’t** make any assumptions on data.

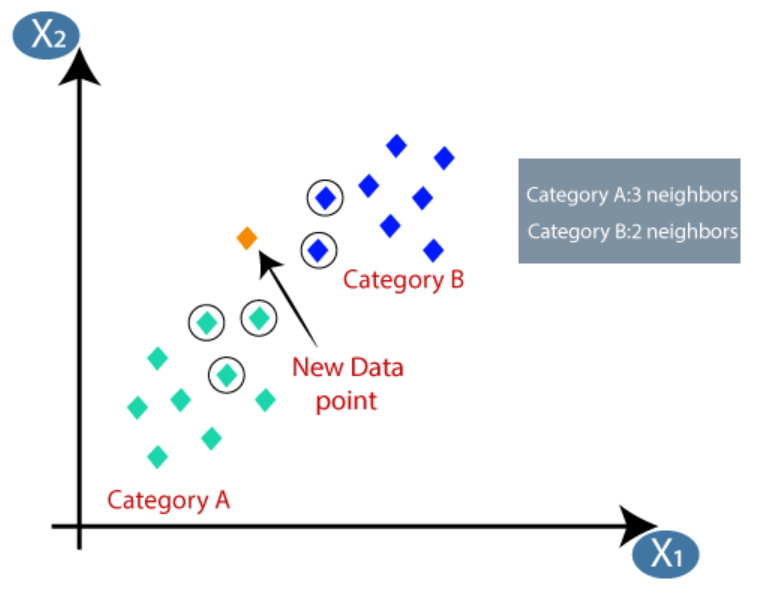




**Working of KNN**

* **Step 1:** **Select** **K**, the number of neighbours.
* **Step 2:** **Calculate** **Euclidean distance** of each neighbour.
* **Step 3:** **Mark** **K-nearest** neighbours.
* **Step 4:** **Check** how many points belong to which category.
* **Step 5:** **Assign** our new datapoint the category with **highest number** of neighbours.





Points to consider:-

* One can choose **any** value for **K**.
* **5** is the **most preferred** value however.
* Low values of **K** like **1** or **2** **doesn’t** provide guaranteed accuracy.
* **Noisy:** High fluctuation in data.
* That **doesn’t** mean that a very large value must be chosen, which does provide **accuracy** but one may find it **difficult** to handle over time.

**Merits & Demerits of KNN**

Advantages:-

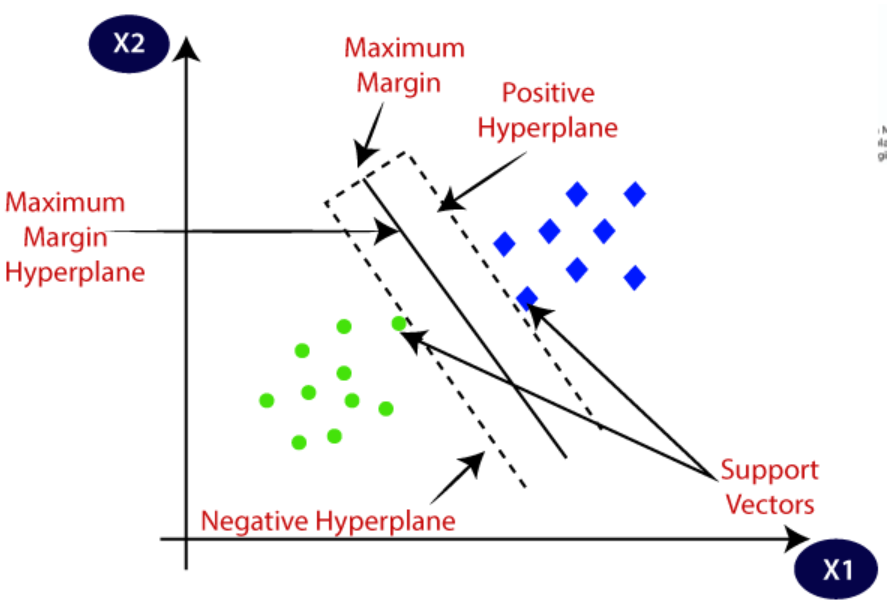
* Can work with **noisy** data.
* Effective with large **training data**.

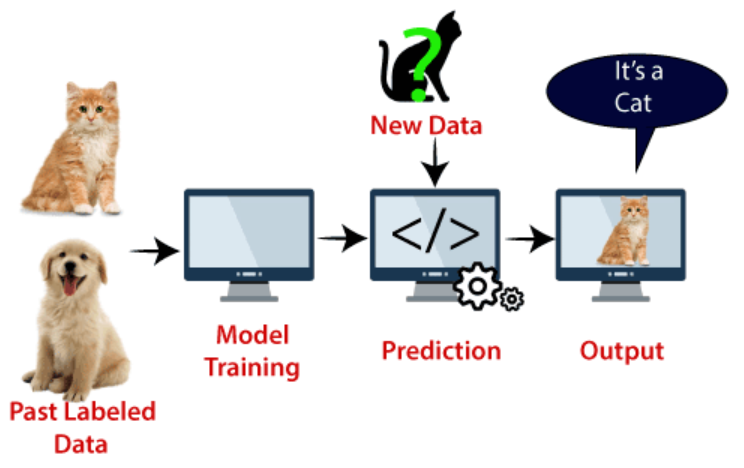
Disadvantages:-

* Have to decide **K’s value** for including **each data**.
* High computation cost (for calculating **distances** multiple times).

**Support Vector Machine (SVM)**

* SVM is also used for **both** ***regression*** & ***classification***.
* Its main goal is to create a **separation line** separating classes.
* This separation line is known as ***hyperplane***.
* And a datapoint added in future joins one of the two classes.
* Points from both classes which are **closest** to ***hyperplane*** are called **support vector**.





**Types of SVM**

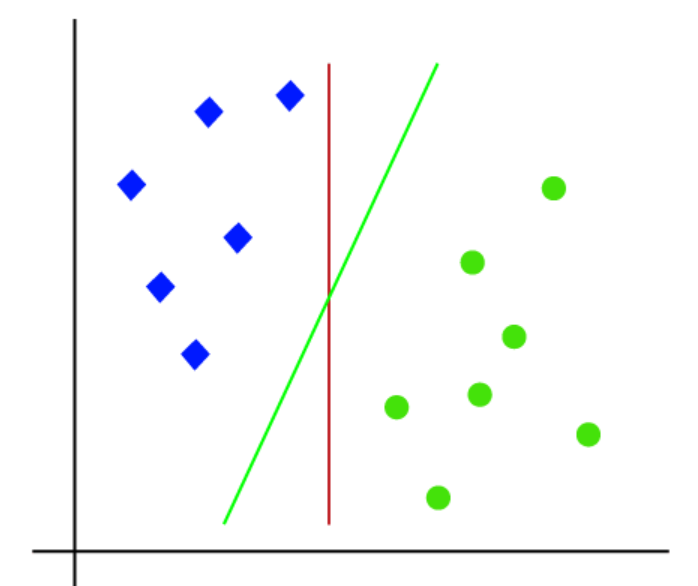
* **Linear SVM:** Has linearly separable classes.
* **Multiple SVM:** **Not** separable with a straight line.

**Hyperplane in SVM**

* ***Hyperplane*** may vary in dimensions.

**SVM Working**

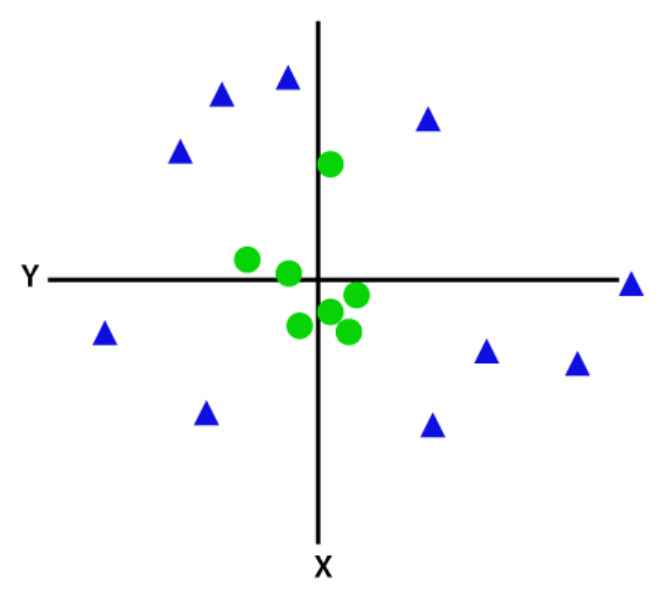
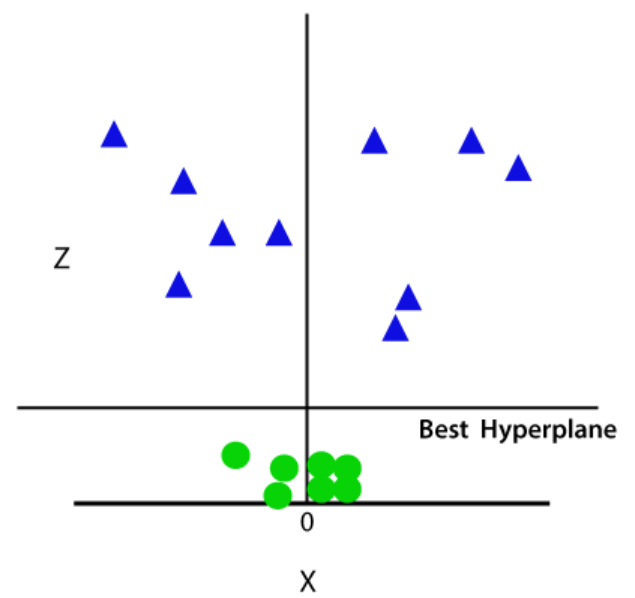
Linear SVM:-



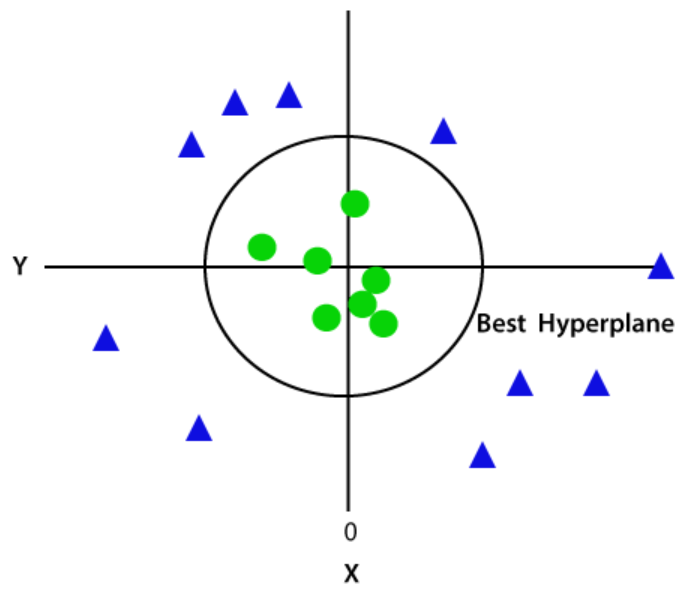
* There can be multiple ***hyperplanes***.
* But our goal is to choose the one with **highest margin value**.

Non-linear SVM:-

* For **non-linear** situations, we will represent datapoints in 3D. For example:

* Also, a different **2D representation** is possible:



**Naïve Bayes Algorithm**

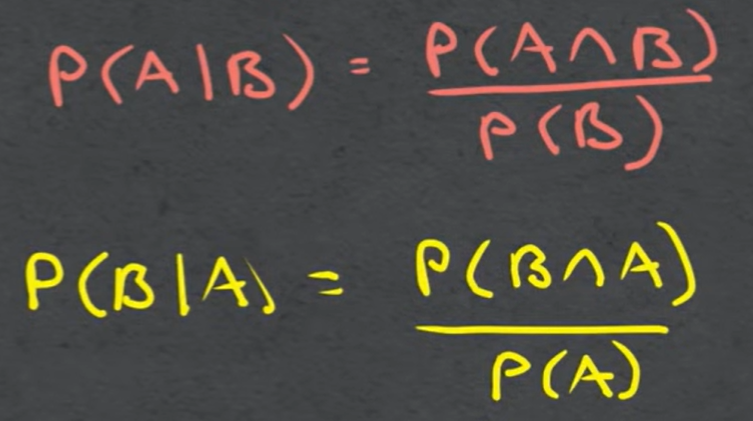
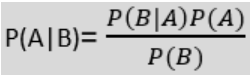
* Used in **text classification** & **sentimental analysis** etc.
* Classifies on the basis of **probability**.

**Name “Naïve Bayes”**

* ***Naïve*** means **lack of experience** or **independent of experience**.
* In this algorithm, an object is identified as per its **individual feature**.
* For example, if an object is **red**, **round** & **sweet** then its **apple**.
* ***Bayes*** because it depends on **Bayes theorem**.

**Bayes Theorem**

* Also known as **conditional probability**.

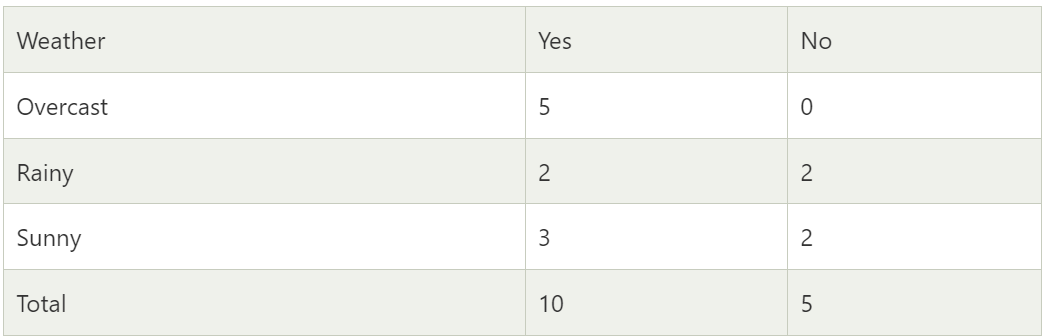
 

* **A** is **hypothesis** & **B** is **event**.
* **Posterior probability:** **P(A|B)** i.e. probability of **hypothesis A** given **event B** has occurred.
* **Likelihood probability: P(B|A)** i.e. probability of occurrence of **event B** given **hypothesis A** is true.
* **Prior probability:** **P(A)** i.e. probability of **a** **hypothesis**.
* **Marginal probability:** **P(B)** i.e. probability for **an event** to occur.

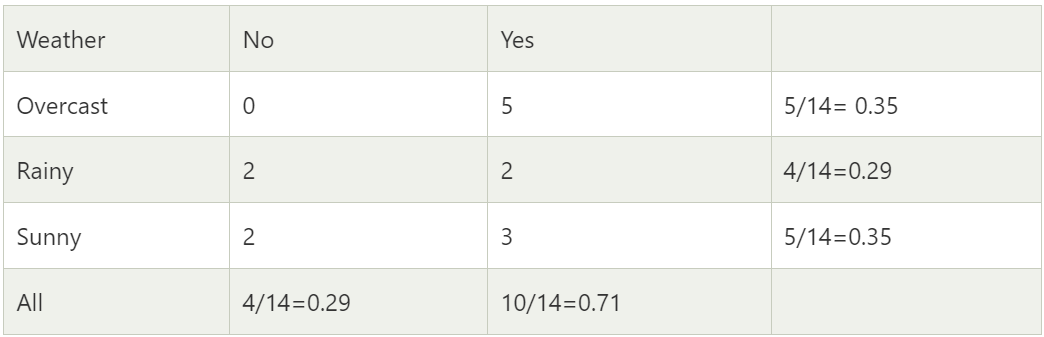
**Working of Naïve Bayes Classifier**

We will use example of if a kid should play outside or **not**:-

* **Step 1:** Get the table telling if kid should **play or not** in various **weathers**.
* **Step 2:** Then make a **frequency table** based on original table.



* **Step 3:** Now calculate the **likelihood** in a new column.



**P(Yes|Sunny) = P(Sunny|Yes)\*P(Yes)/P(Sunny)**

**P(Sunny|Yes) = 3/10 = 0.3**

**P(Sunny) = 0.35**

**P(Yes) = 0.71**

**So, P(Yes|Sunny) = 0.3\*0.71/0.35 = 0.60**

**P(No|Sunny) = P(Sunny|No)\*P(No)/P(Sunny)**

**P(Sunny|NO) = 2/4 = 0.5**

**P(No) = 0.29**

**P(Sunny) = 0.35**

**So, P(No|Sunny) = 0.5\*0.29/0.35 = 0.41**

**P(Yes|Sunny)>P(No|Sunny)**

**Hence the kid can play on sunny days.**

**Merits & Demerits of Naïve Bayes Classifier**

Advantages:-

* **Fast** algorithm.
* Can also be used in **multi-class** classifications.
* Performs **multi-class** classification **better** than others.

Disadvantages:-

* Model **doesn’t** learn on the basis on relationship between features.

**Types of Naïve Bayes Models**

* **Gaussian:** Predictor takes **continuous values** instead of discrete.
* **Multinomial:** Categorizes on the basis of **frequency of words** (used in **NLPs**). Like whether a document belong to school, office or finance.
* **Bernoulli:** Similar to multinomial but it tells in **binary**. Like if a word is present or not in a document.