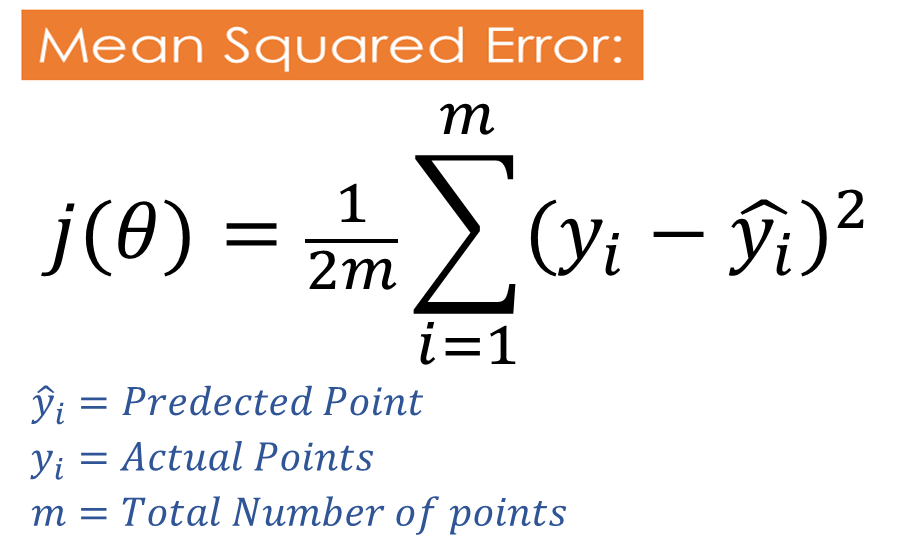
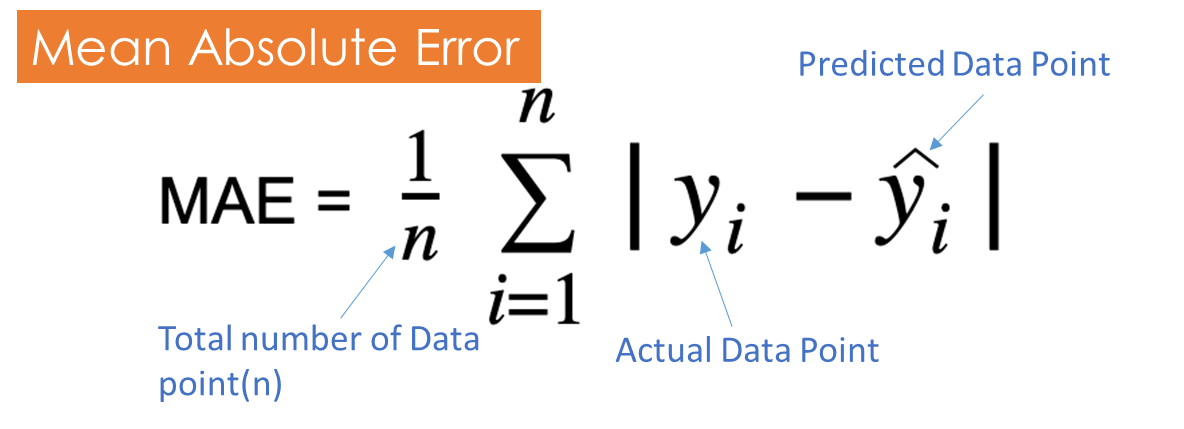
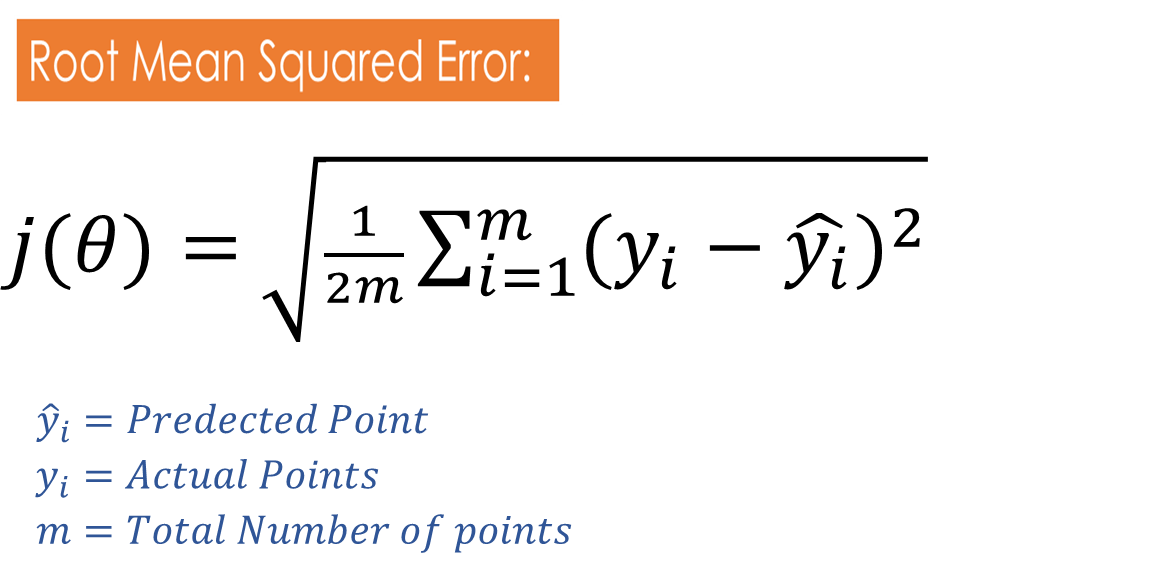
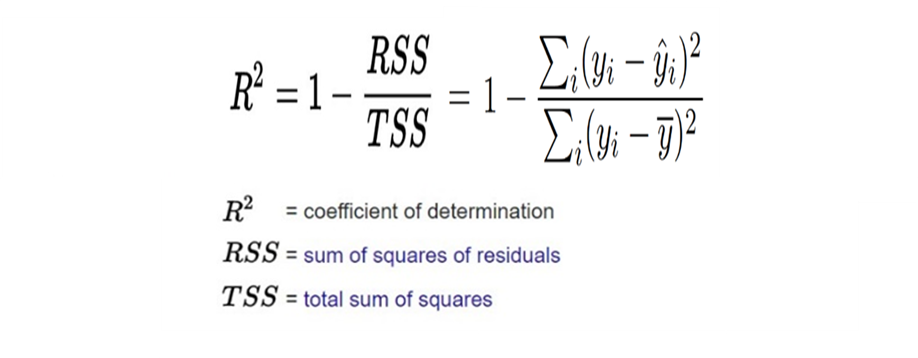
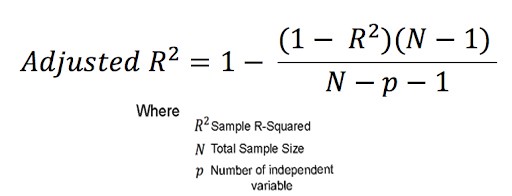
**Linear Regression**



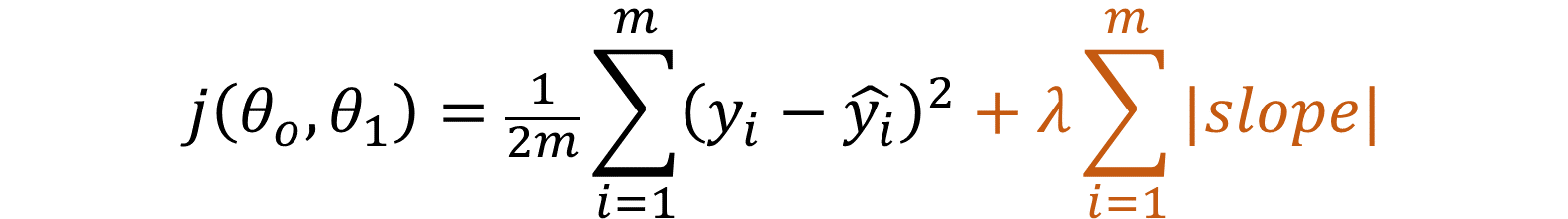


****

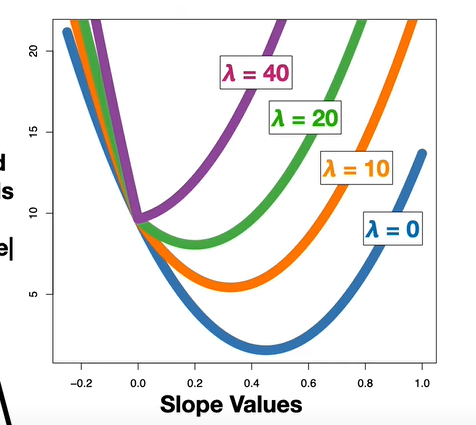
****



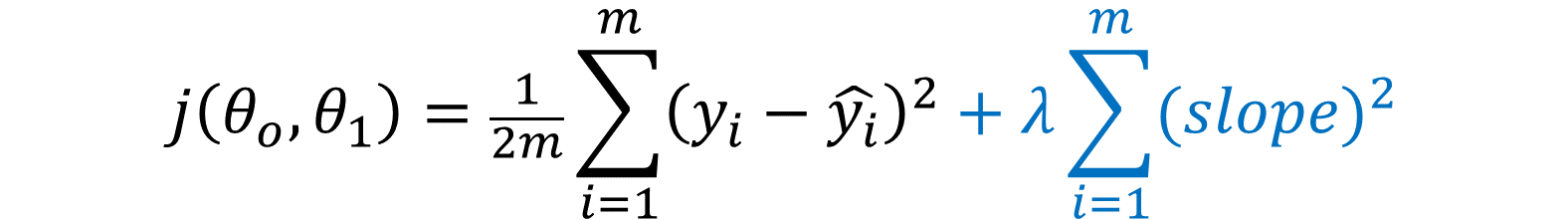
**Equation after adding the Lasso(L1)**



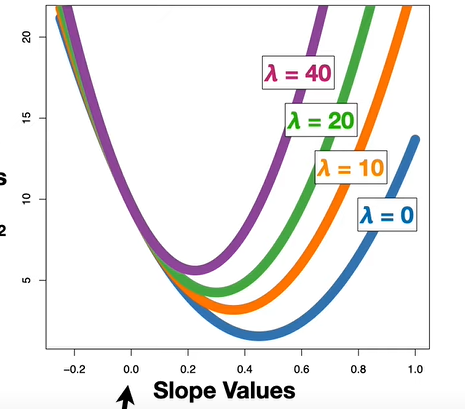
The main aim of the Lasso is to reduce the **number of features** it is used to do the **feature selection**.



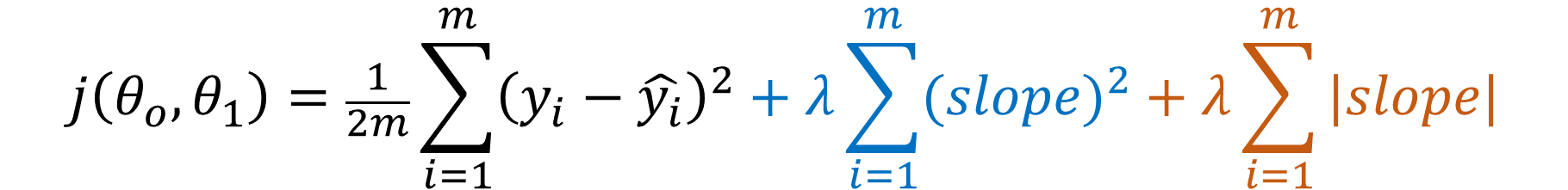
**Equation after adding the Ridge(L2)**



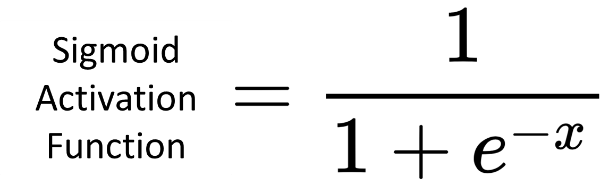
* **Avoid overfitting**.
* Useful for dealing with **multicollinearity**
* when you want to reduce the impact of all features without eliminating them entirely.



**ElasticNet Regression:**



**Logistic Regression**



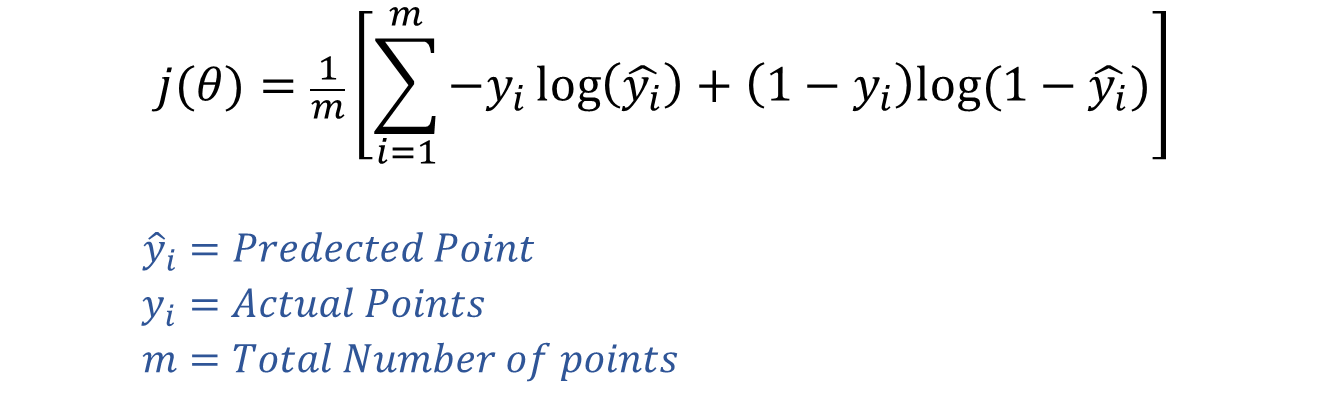
|  |
| --- |
| Equation of the straight line |
| **hθ(x)** = **θ0**+ **θ1** **X1** |

Let’s consider -🡪 Z = **θ0**+ **θ1** **X1**



Above sigmoid activation creates non-convex function, which creates multiple minima.

**Log Loss Cost Function:**



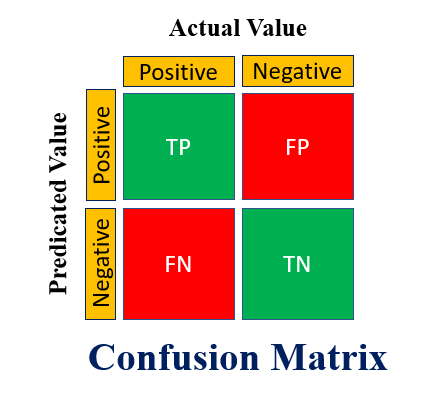
Above Equation creates a convex function, so there is only one minima.

**Performance Metrix for classification.**

1. Confusion matrix
2. Accuracy
3. Precision
4. Recall
5. F-Beta Square

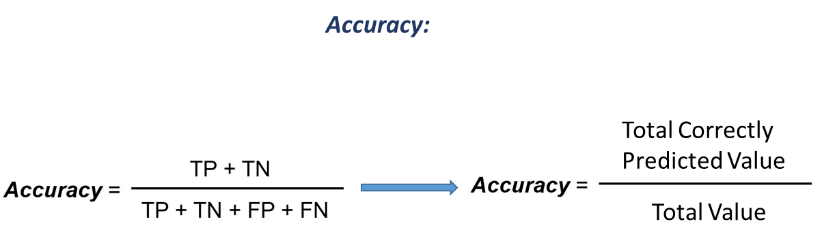
**Confusion matrix :**

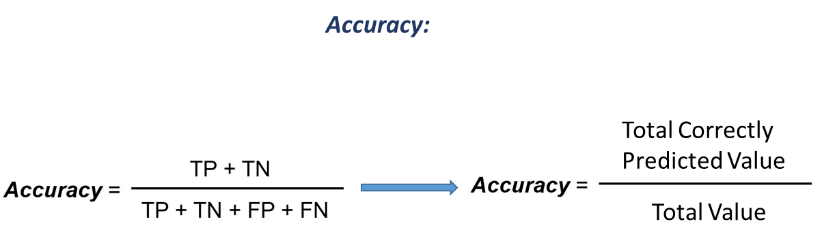
The confusion matrix can be explained by studying the graph below.



**Accuracy:**

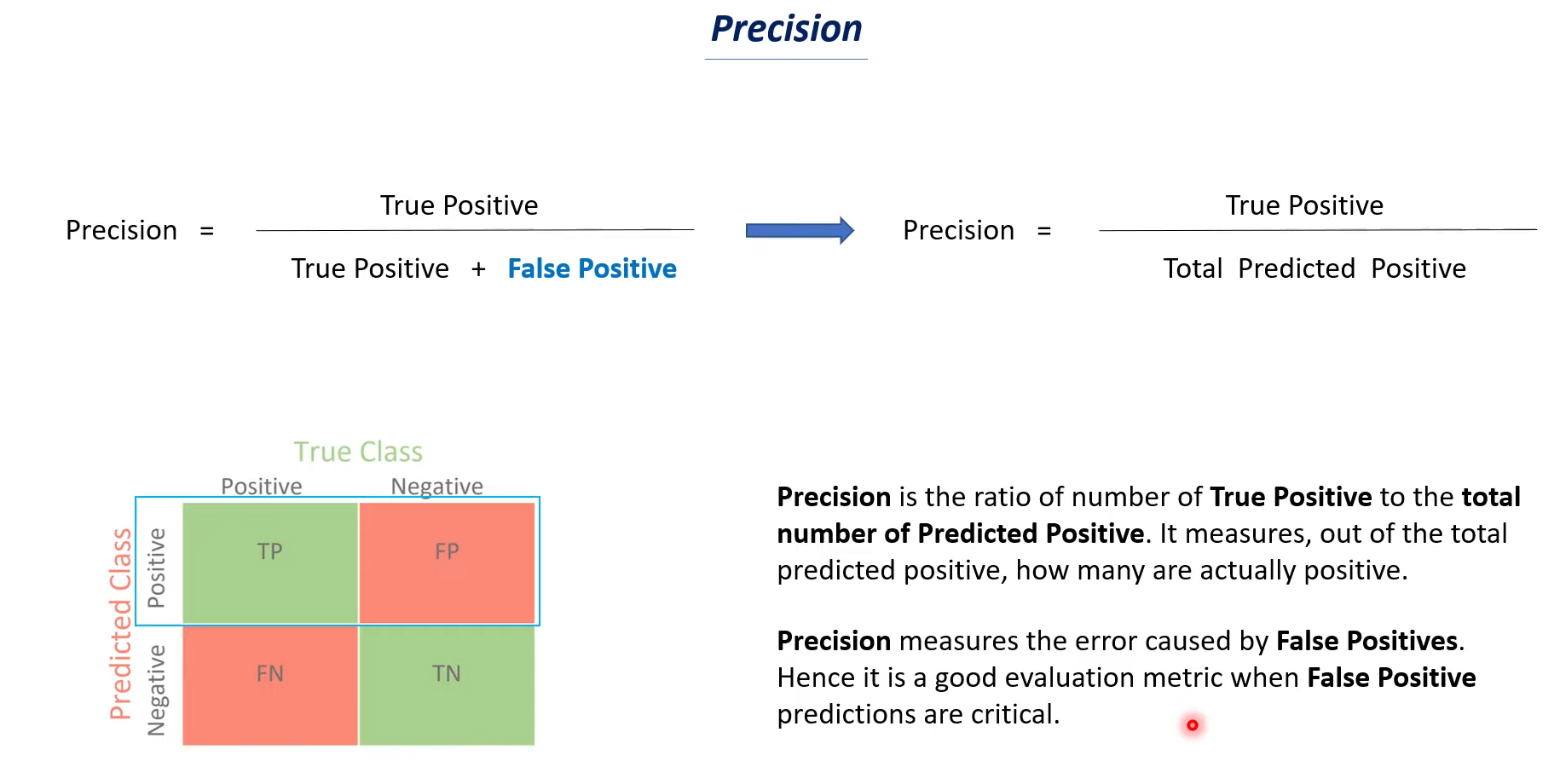
The accuracy function will predict the overall performance of the model.

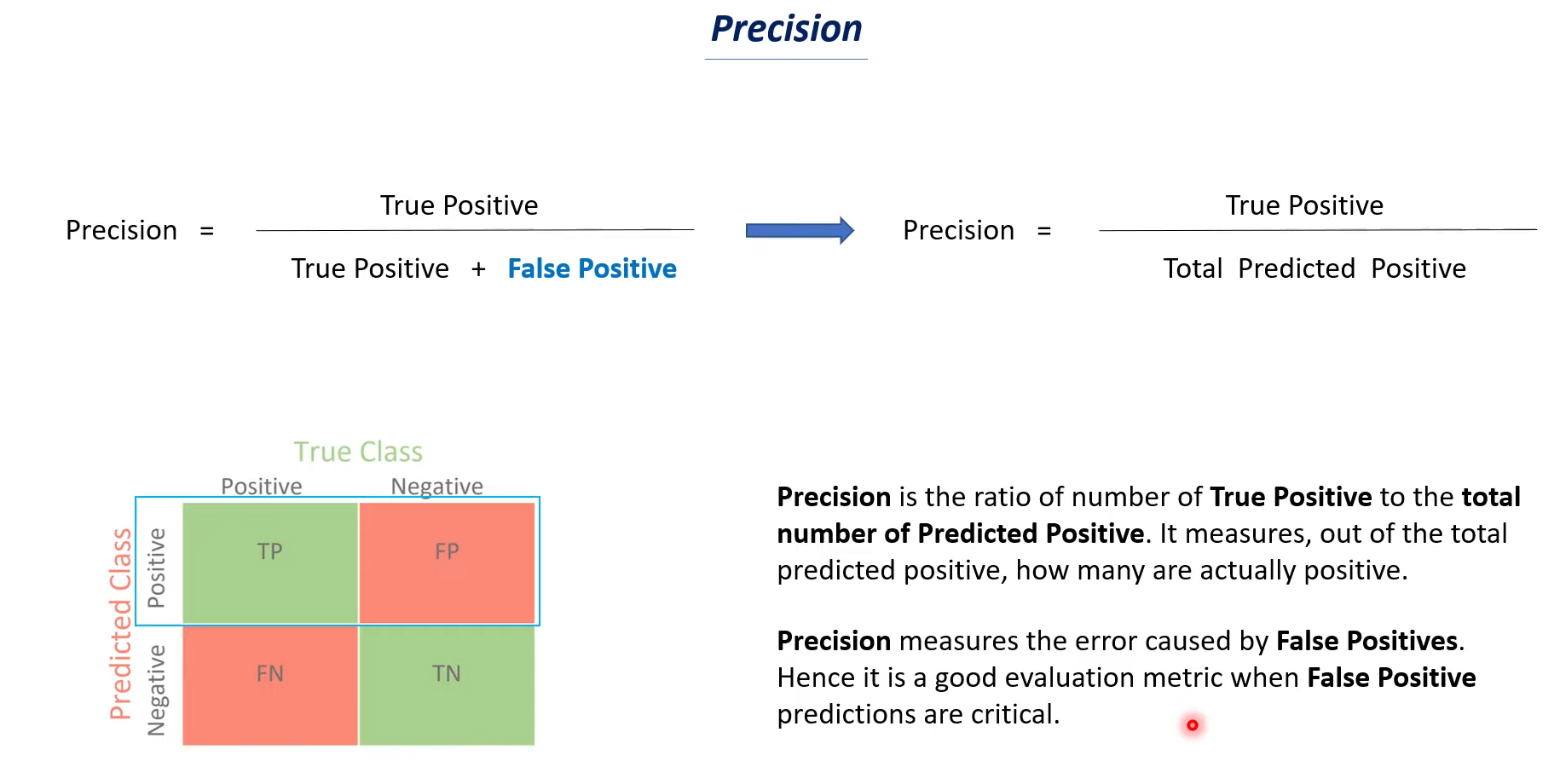


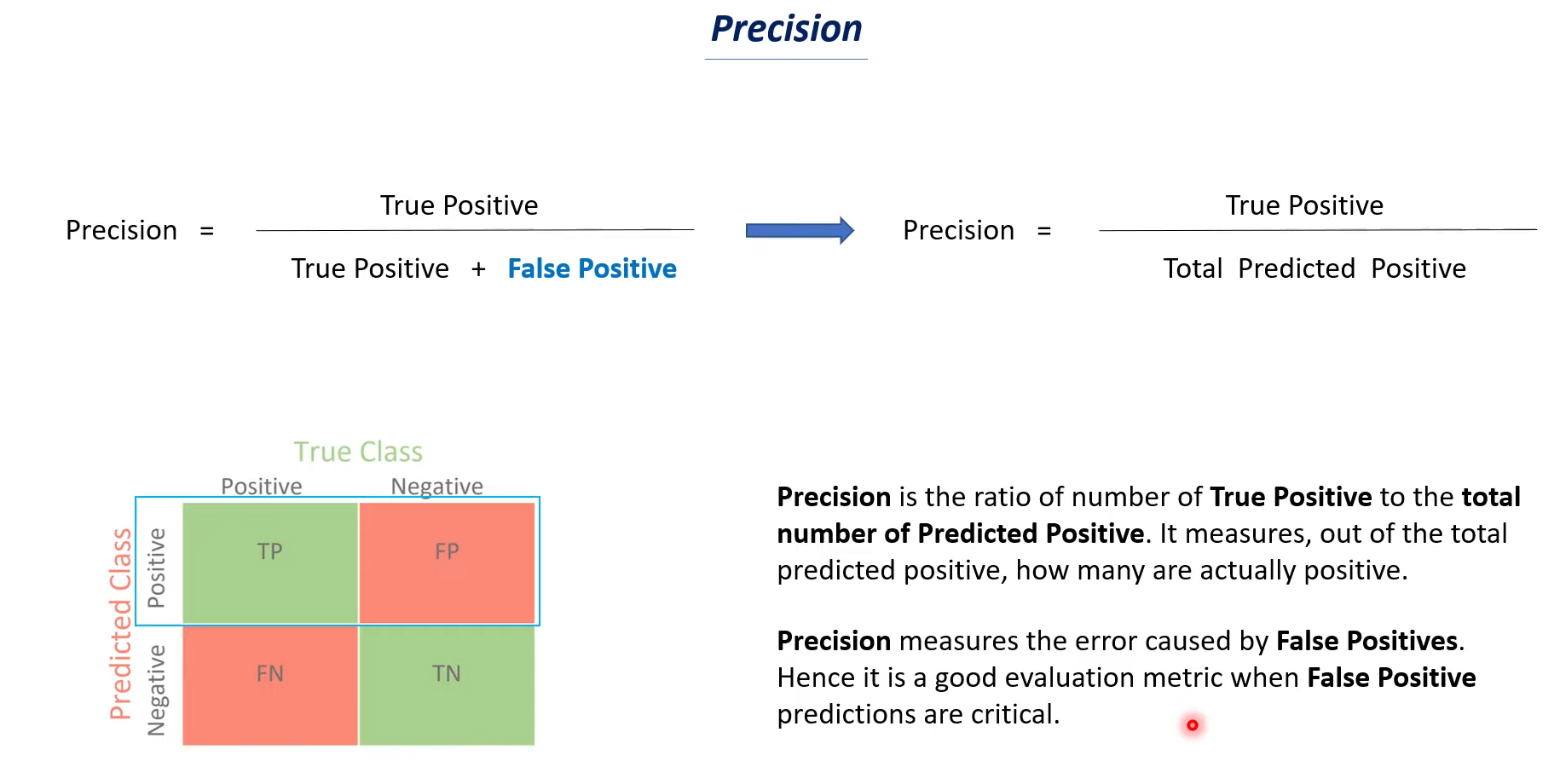


**Precision**

**Minimize** “**false positives**”







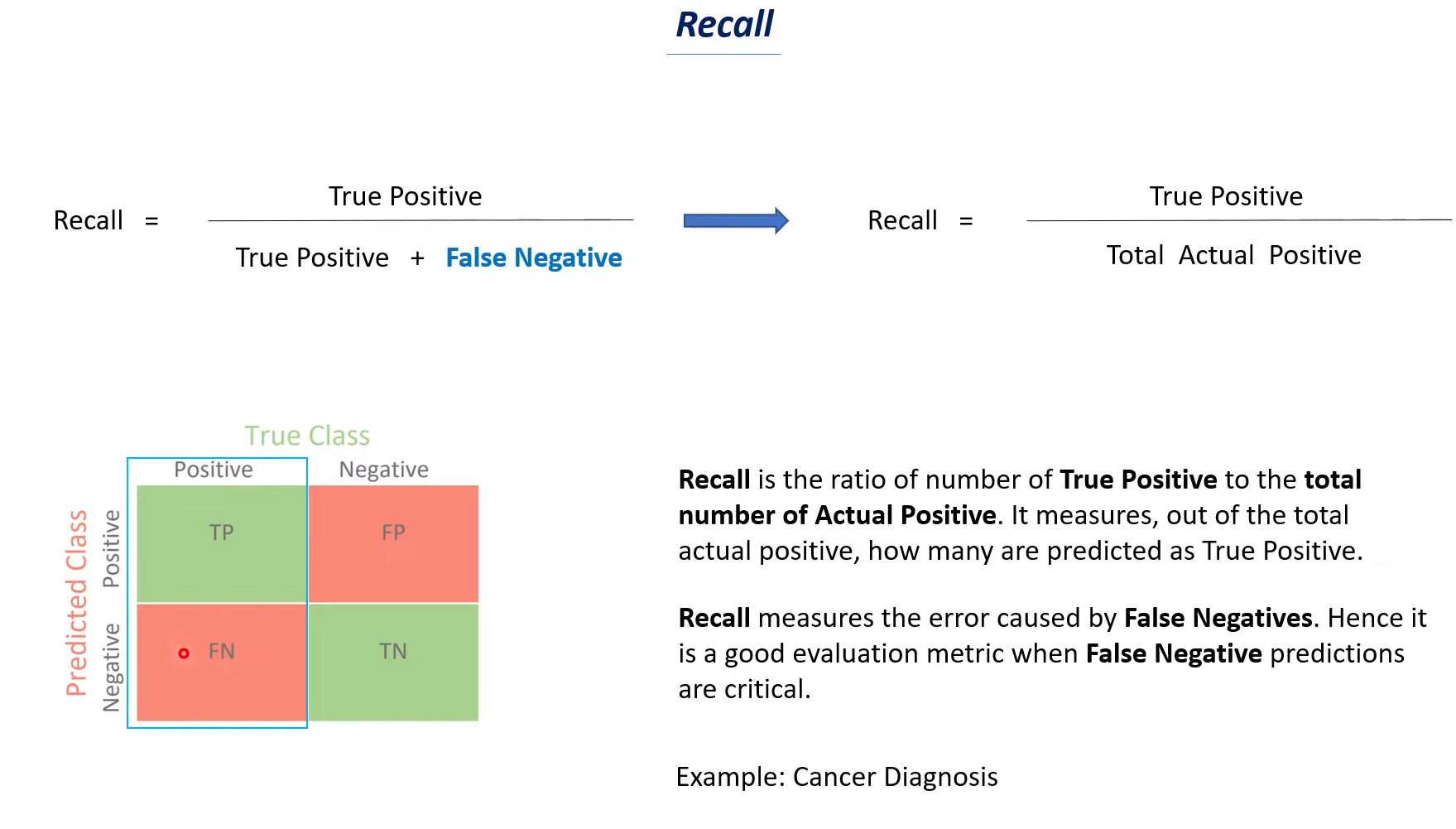
Example :

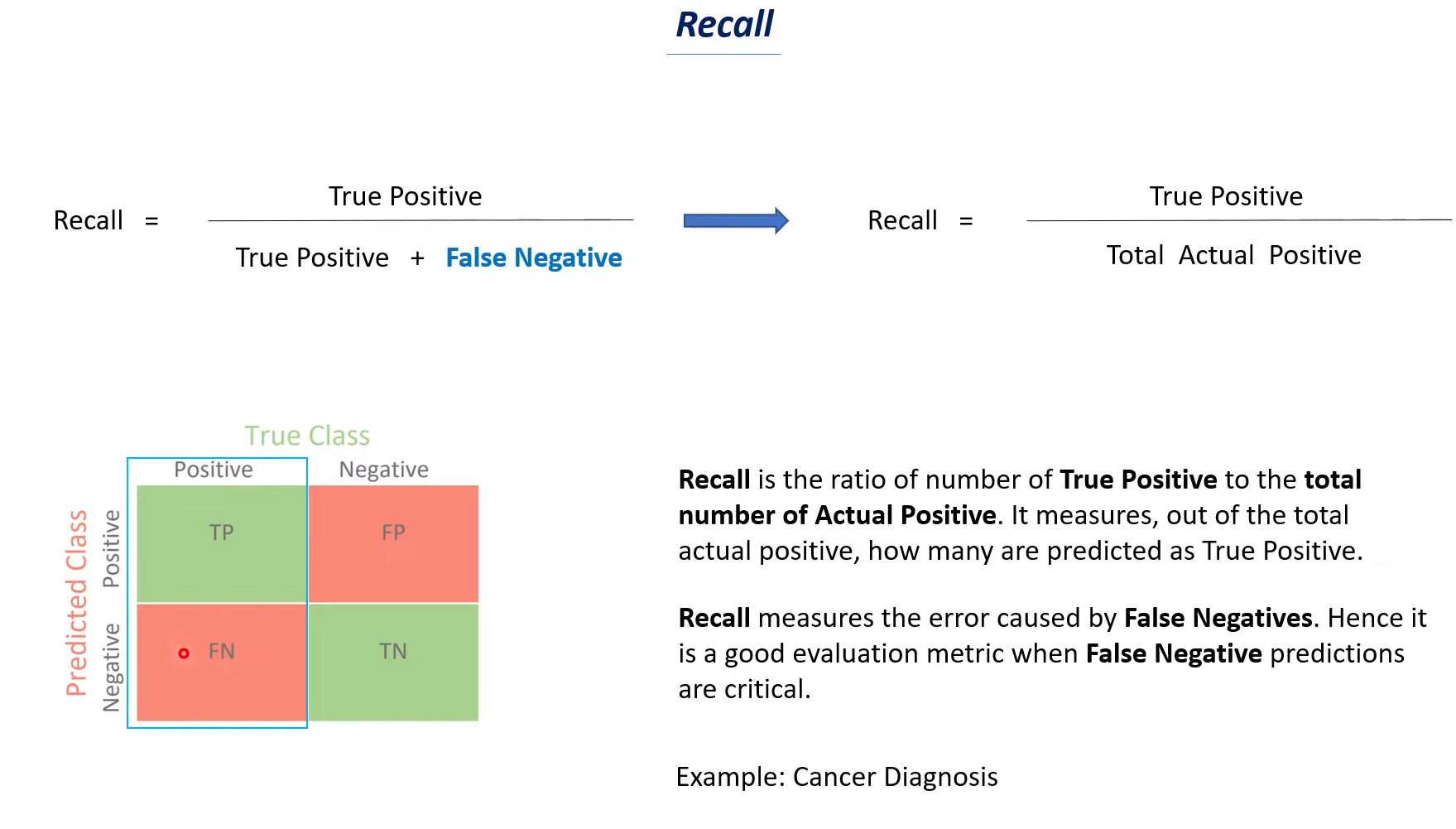
**Face recognition:**

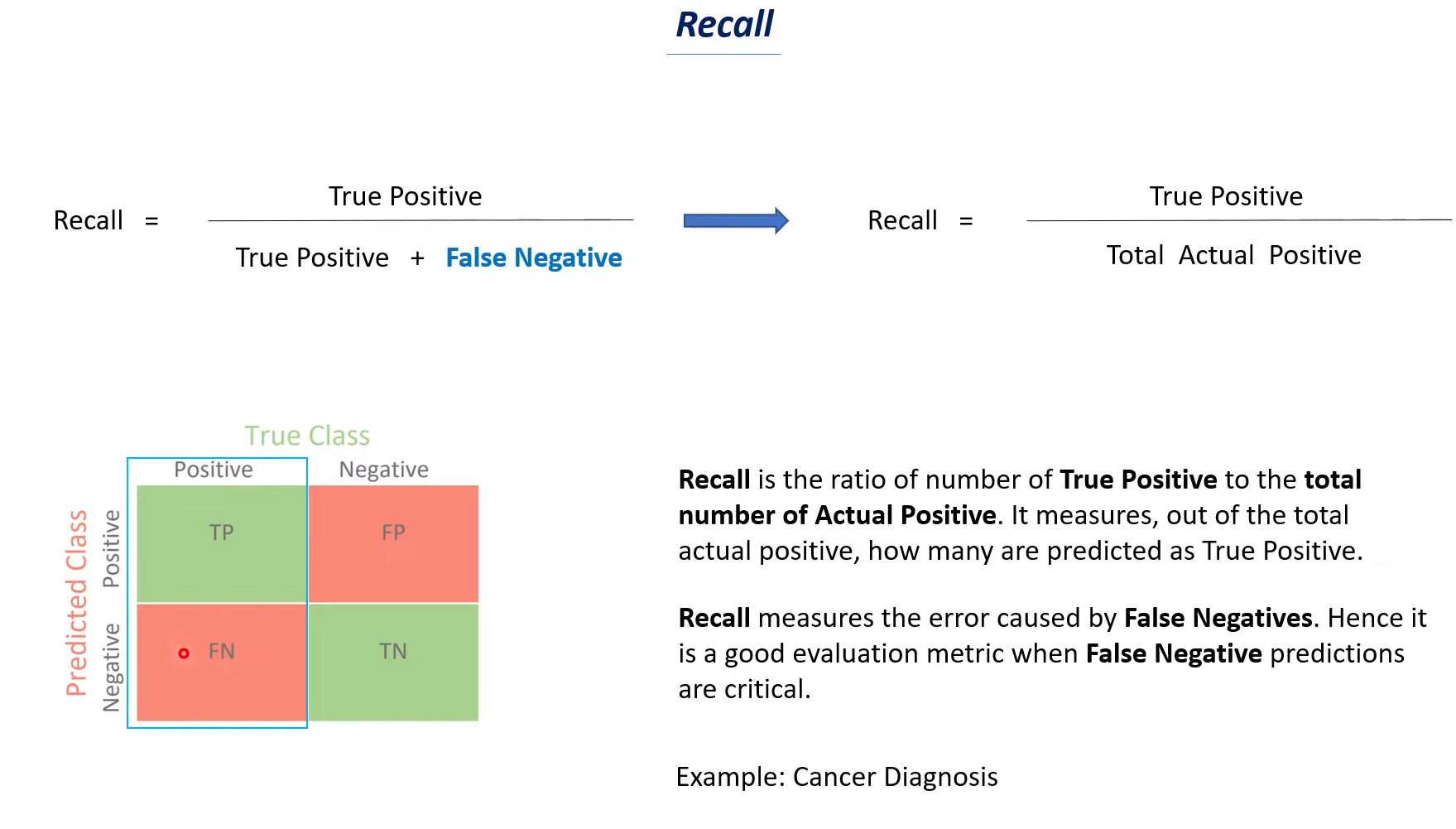
**Email classification to Spam:**

**Recall:**

**Minimize** **“false negatives”**. For example, in medicaldiagnostics,

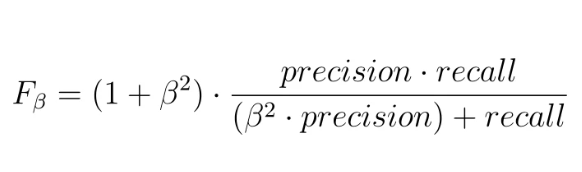






**F-measure:**

If both True positive and False Negative is important.



**SVM**

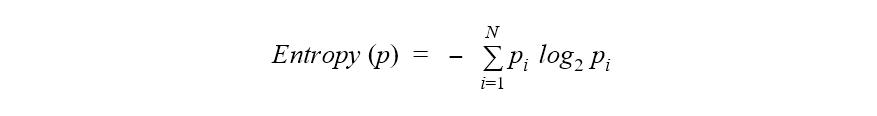
Hard Margin

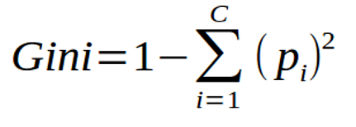
To find out the location of the point needs to be divided by then we can use the below equation to locate the point which falls above and below the hyperplane.

**Soft Margin**

Sum of the distance of all incorrectly classified points. Here the distance will be taken from the marginal plane.

**Descision Tree**





H(S) = Entropy or Gini of the root Node.  
Sv = total values in a particular node.  
S = Total Number of Values of Output Feature  
H(Sv) = Entropy or Gini of the Child Node

Adaboosting:

**Performance of Stump**

Recalculate the weight using below formula

XGBoost (Extreme Gradient Boosting) is a powerful machine-learning algorithm

One of the key reasons for XGBoost's success is its ability to handle complex relationships in data while being computationally efficient.

* It includes **regularization techniques like L1 and L2 regularization** to prevent overfitting, making it more robust and improving its generalization capabilities.
* XGBoost can **handle missing values** in data without requiring explicit imputation, which simplifies the preprocessing steps.
* Additionally, it employs **tree pruning** to control the growth of decision trees, preventing overfitting and improving model performance.
* Its ability to **perform parallel processing** across multiple CPU cores makes it even faster and scalable to large datasets.

Due to its speed, accuracy, and numerous optimizations, XGBoost is commonly used for various machine learning tasks, including classification, regression, ranking, and recommendation systems.

**Gradient Boosting:**

Gradient boosting is a machine learning technique used for building predictive models. Gradient boosting is an ensemble machine-learning technique that combines multiple weak prediction models, to create a powerful predictive model. It sequentially trains models to correct the mistakes of previous models by focusing on the residual errors.

It is effective at capturing complex patterns and is widely used for regression, classification, and ranking tasks.

1. Develop the first model (m1) to predict the mean (average) of the output column.
2. Calculate the “pseudo residual” of the first model (m1) using the formula below

“Pseudo Residual” = “Actual Y” – “Mean”

1. Prepare the second model (m2) to predict the residual. In this model, use the input features (x1,x1…..xn) as inputs, as mentioned instead of predicting Y, it will predict the residual (error).
2. The purpose of the second model (m2) is to assess the intensity of the residual, rather than predicting the Y value directly. To calculate the “Y value”, use the following formula: Y predict = m1 + m2 (where M1 is the output (Mean) of the first model and m2 is the predicted residual from the second model)
3. By using the above formula, you will observe that it predicts the correct “Y value”, which closely matches the **“actual Y value”**. This indicates that the model may be overfitted. To avoid overfitting, introduce a "learning rate" and modify the formula as follows: Y predict = M1 + (learning rate \* m1)
4. Repeat the above steps until the desired result is achieved.