DATA AND
ARTIFICIAL INTELLIGENCE

Classification

©Simplilearn. All rights reserved.

DATA AND
ARTIFICIAL INTELLIGENCE

Logistic Regression

©Simplilearn. All rights reserved.

Logistic Regression

The Logistic function relates the predictor variables to the probability of the event occurring.

$$P(y_i=1) = F(\beta_0 + \beta_1 X_i)$$

$$P(y_i=0) = 1 - P(y_i=1) = 1 - F(\beta_0 + \beta_1 X_i)$$

- Where, F(.) is the ‘Logistic Function’.
- F(.) should return values between 0 and 1.
- e.g., If response variable (y) is ‘Customer Churn’ which is binary. ‘$P(y_i=1)$’ is the probability of a customer to churn.

Sigmoid Function

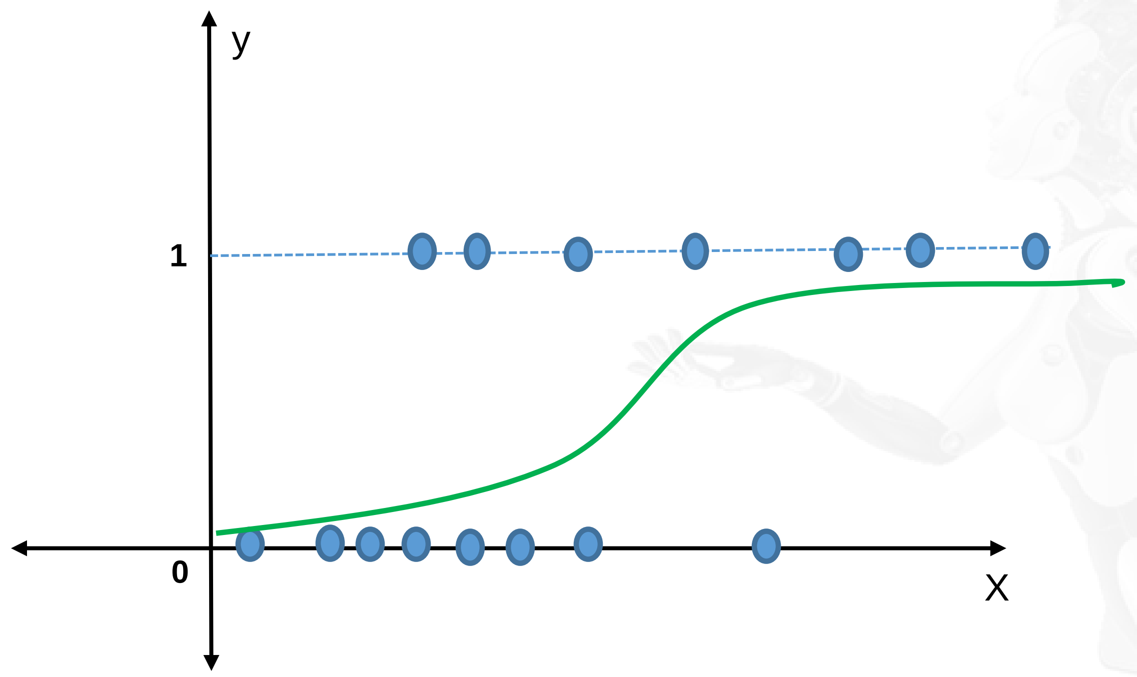
$$F(z) = \frac{e^z}{1+ez}$$

F(z) will return values always between 0 and 1. It is called as
Sigmoid Function.

$$\text{Prob}(y_i=1) = F(\beta_0 + \beta_1 X_i)$$

$$P(y_i=1) = \frac{e^{\beta 0 + \beta 1\ Xi}}{1+e^{\beta 0 + \beta 1\ Xi}}$$

$$\ln\left(\frac{P(yi=1)}{1-P(yi=1)}\right) = \ln(\text{ODDS}) = \beta 0 + \beta 1\ Xi$$



simplilearn

Maximum Likelihood Estimation

To establish a Logistic Regression model, the value of β_0 and β_1 needs to be determined that is done using the Maximum Likelihood Techniques.

This helps to find the parameters that make the observed values most likely to have occurred. i.e., it maximizes the probability of obtaining the samples at hand.

Probability of observing actual 'y' for any record = $P(y_i) = (P(y_i=1))^{yi} * (1- P(y_i=1))^{(1-yi)}$

Hence, we calculate β parameters for maximum value of

$$\mathbf{L(\beta) = \prod_{i=1}^{n} [(P(yi=1))^{yi} * (1- P(yi=1))^{(1-yi)}]}$$

Log-loss function and cost function

1. Log Loss for a Single Example

Log Loss (Binary Cross-Entropy Loss) for a single training example is:

$$L(y_i, \hat{y}_i) = -\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

where:

- y_i = actual class label (0 or 1)
- $\hat{y}_i$ = predicted probability of class 1 ($\hat{y}_i = \sigma(z_i)$)
- $\sigma(z) = \frac{1}{1+e^{-z}}$ (sigmoid function)

simplilearn

Log-loss function and cost function

2. Log Loss Cost Function (for the entire dataset)

For m **training examples**, the cost function (average log loss) is:

$$J(\beta_0, \beta_1) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

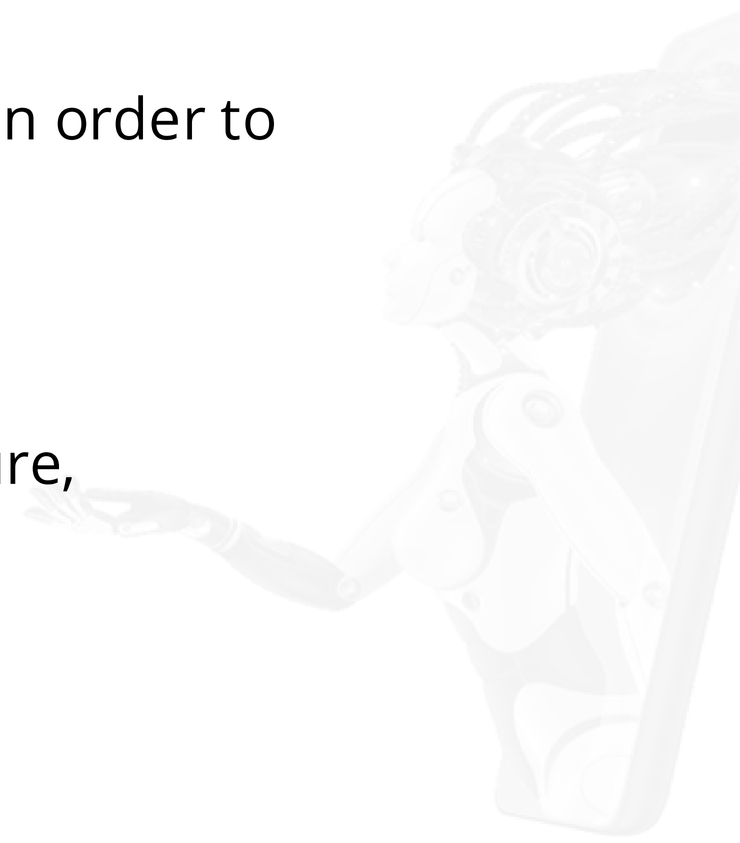
where:

- $J(\beta_0, \beta_1)$ = cost function to minimize
- $\hat{y}_i = \sigma(\beta_0 + \beta_1 x_i)$

Overcoming Overfitting

©Simplilearn. All rights reserved.

What is Regularization?



Regularization is the use of techniques to calibrate machine learning models in order to prevent overfitting or underfitting

In layman speak, we use regularization to make our model work harder when training/fitting, so that it generalizes better (likely to perform better on future, incoming, and yet unknown data)

Types

- Lasso Regression (L1 Regularization)
- Ridge Regression (L2 Regularization)
- ElasticNet Regression (Combination of both L1 and L2 Regularization)

Lasso Regression

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda \underbrace{\sum_{j=1}^{p}|\beta_j|}_{\ell^1-penalisation}$$

λ (Lambda), here is a parameter we arbitrarily choose to increase or decrease the strength/difficulty of the additional penalty (L1 regularization)

Ridge

- Ridge regression adds the “squared magnitude” of the coefficient (gradient of each X variable) as the penalty term to the loss function.

$$\underbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2}_{\text{residual sum of squares}} + \lambda \underbrace{\sum_{j=1}^{p}\beta_j^2}_{\ell_2\text{−}penalisation}$$

λ (Lambda), here, is again a parameter we arbitrarily choose to increase or decrease the strength/difficulty of the additional penalty (L2 regularization)

λ for Lasso and Ridge (and ElasticNet) can range from 0 to any number, though typically we increase values by order of 10, such as having it take on values of [1, 10, 100, 1000], one at a time to generate predictive models and
compare their performance

Elastic Net

- Combines both the penalties from both the lasso and ridge techniques to regularize regression models

$$\sum_{i=1}^{n}(y_i-\hat{y}_i)+\lambda\left((1-\alpha)\sum_{j=1}^{p}\beta_j^2+\alpha\sum_{j=1}^{p}|\beta_j|\right)$$

α can range from 0 to 1, and is typically a decimal
It will be the proportion/weightage of the penalty we chose to depend on Lasso
Ridge thus will have (1- α) as its fractional weightage of the penalty

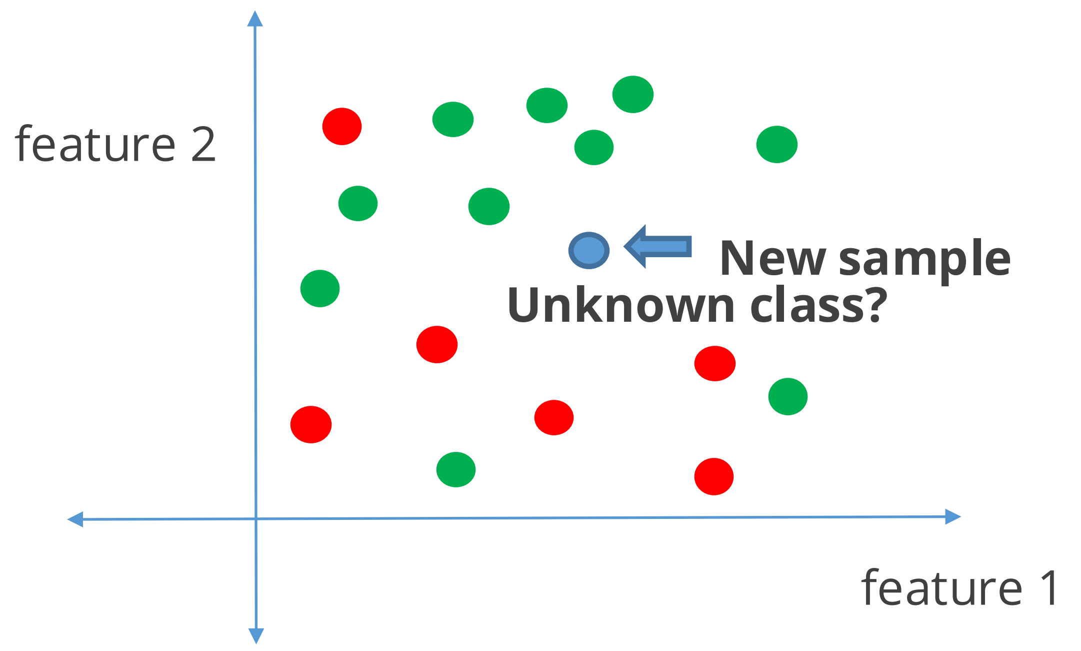
k-Nearest Neighbors(kNN)

©Simplilearn. All rights reserved.

k-Nearest Neighbors

KNN algorithm is a parametric technique that classifies new data points based on similarity that can be measured using Euclidean distance.

KNN algorithm can be used for Regression as well as for Classification but generally it is used to solve Classification problems.



Euclidean distance between 2 points A(x_1, y_1) and B(x_2, y_2)

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

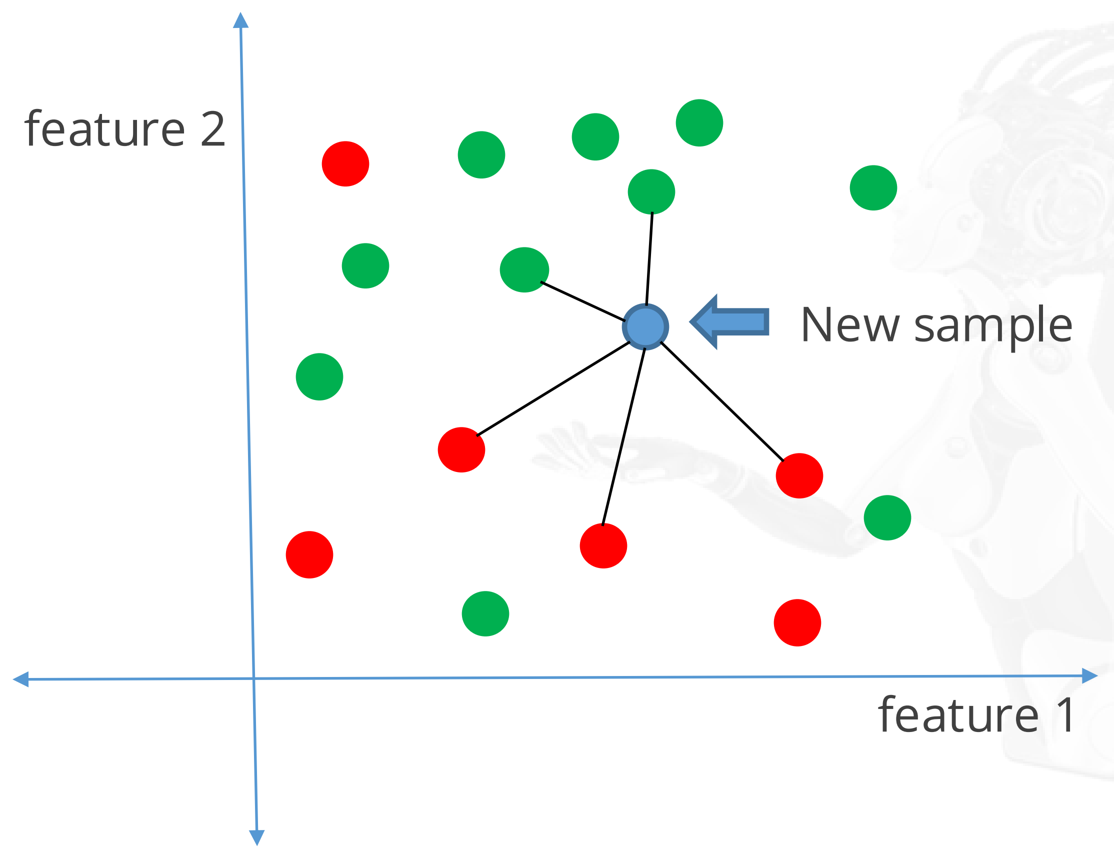
kNN Methodology

Step 1: select the value of k (no of neighbors).

Step 2: Take k neighbors based on the distance measurement.

Step 3: Count the number of samples of each class in the neighborhood.

Step 4: Assign the majority class to the new sample.

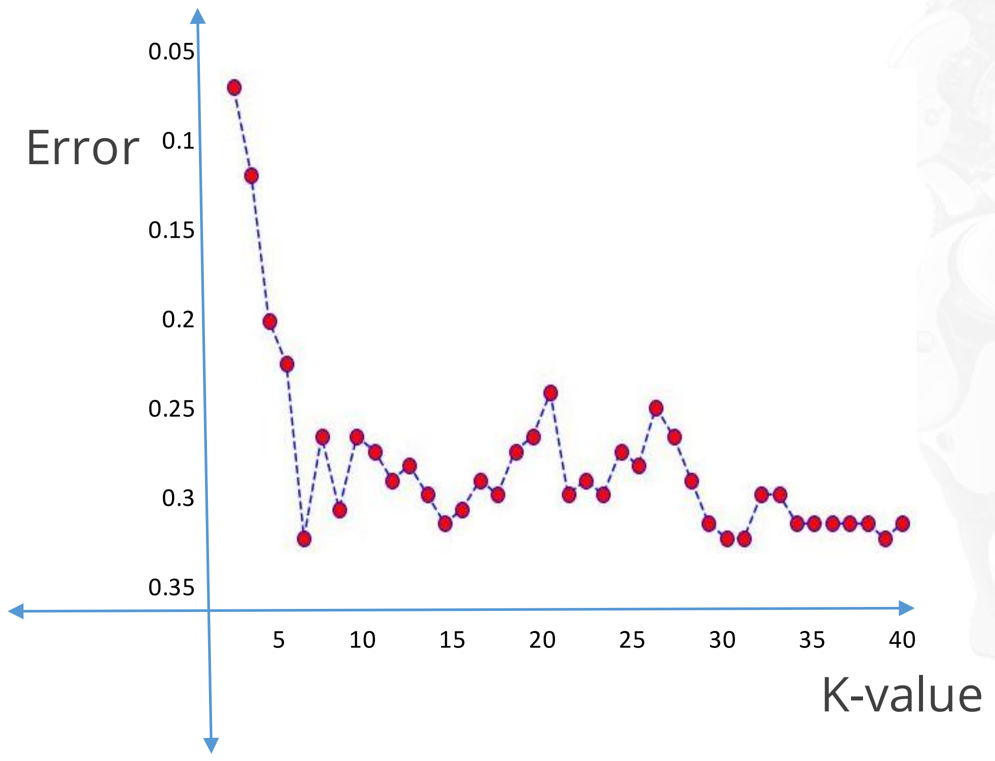


Deciding the Value of k

Value of k is decided based on performance metric like error/accuracy.

✓ K should not be taken as a very small value to ensure generalization.

✓ K should not be taken as a big value. This will create problems specially in case of unbalanced classes.



simplilearn

DATA AND
ARTIFICIAL INTELLIGENCE

Decision Tree

©Simplilearn. All rights reserved.

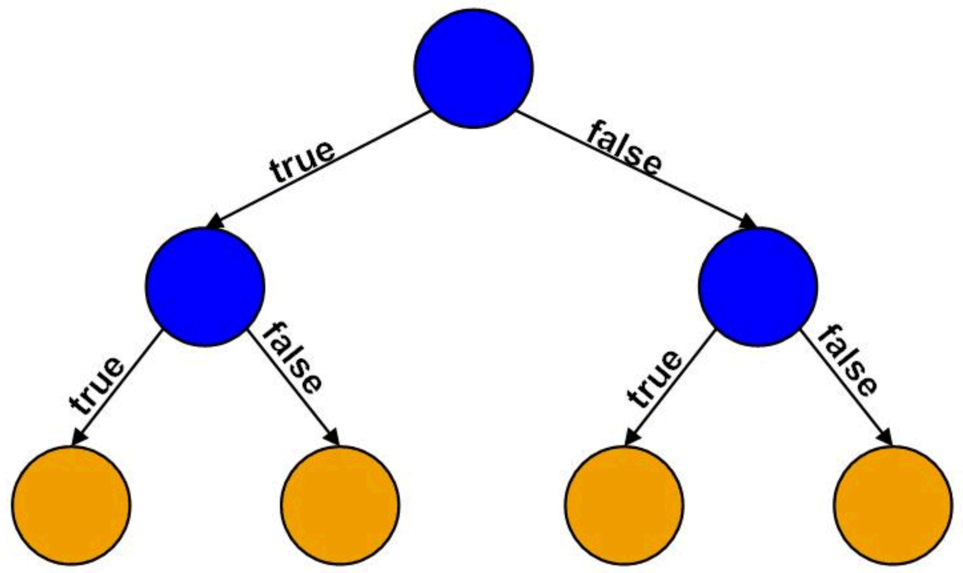
Identifying Possible Loan-Defaulter

A loan applicant has applied for a loan of $40K. He is earning $1250 per month. If his credit score is 720, can he be a possible loan defaulter?

Identifying Possible Loan-Defaulter

Solution: Step 1: Using customer data, create a classification model like decision tree.

Decision tree is used to display an algorithm that contains conditional control statements. It is a representation of patterns identified from observations(samples) in the form of a tree.



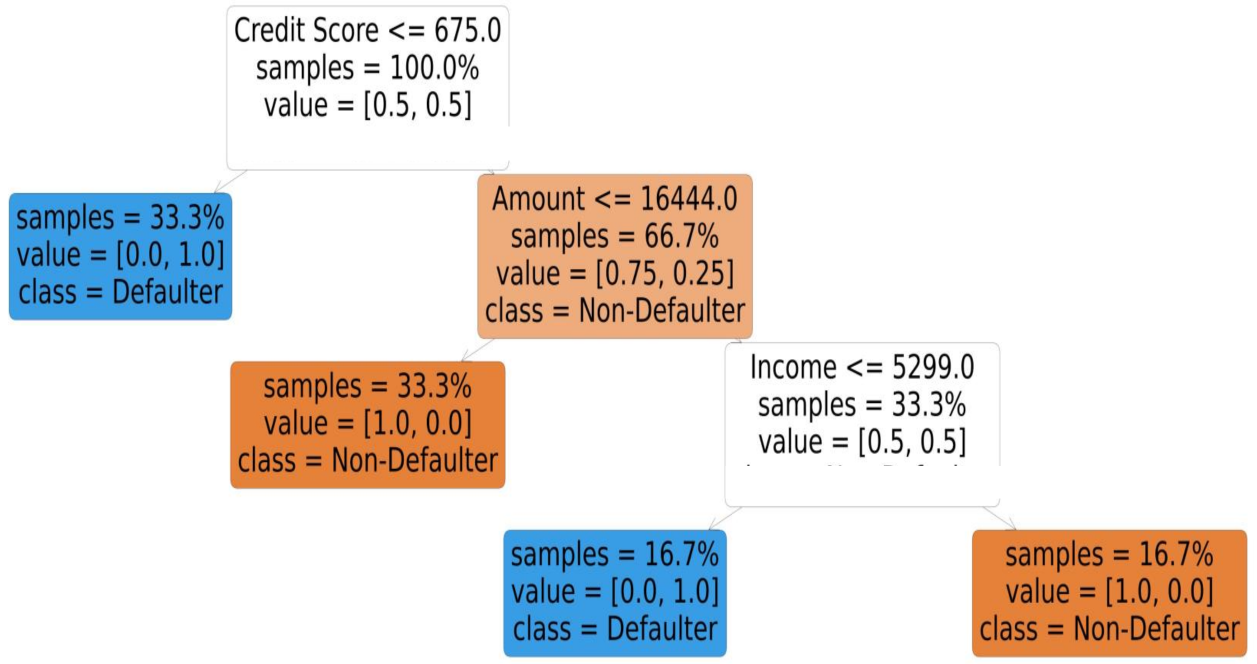
Identifying Possible Loan-Defaulter

Step 1: Using customer data, create a classification model like decision tree.

Customer Data of a Bank

ID	Amount	Credit Score	Income
1	$14000	750	$1000
2	$2000	720	$1200
3	$2000	730	$1250
4	$12000	700	$5500
5	$70000	570	$5000
6	$60000	850	$12000
7	$55000	600	$1190
8	$30000	750	$5398
9	$30000	620	$15000
10	$18888	750	$5200
11	$18000	650	$1380
12	$25000	700	$1280

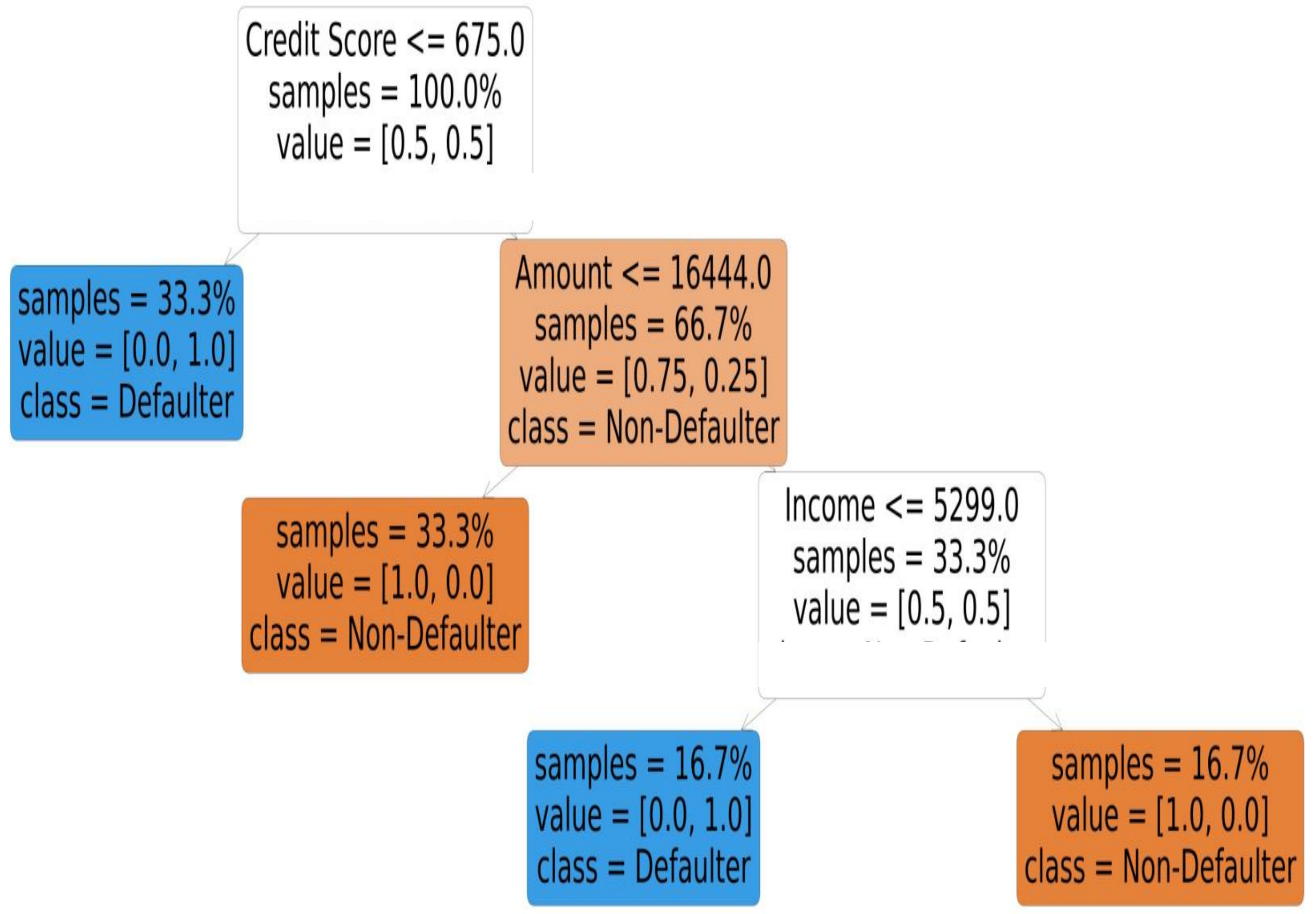
Decision Tree for the data



Identifying Possible Loan-Defaulter

Step 2: Check for the relevant pattern/rule to decide the probable class.

From the customer data, using the decision tree, you can derive that if Amount > 16444 and income > 5299 then it is highly probable that the customer may turn out to be a defaulter.



Popular Techniques

Chi-square Automatic Interaction Detector (CHAID)

Classification and Regression Trees (CART)

C4.5

Splitting Criterion

CART uses Gini index as a measure of impurity.

Gini at a node = GINI = 1 - $\sum p_j^2$

N = 15
C_1 = 10
C_2 = 5

Gini = 1 – $((10/15)^2 + (5/15)^2)$ = 0.44

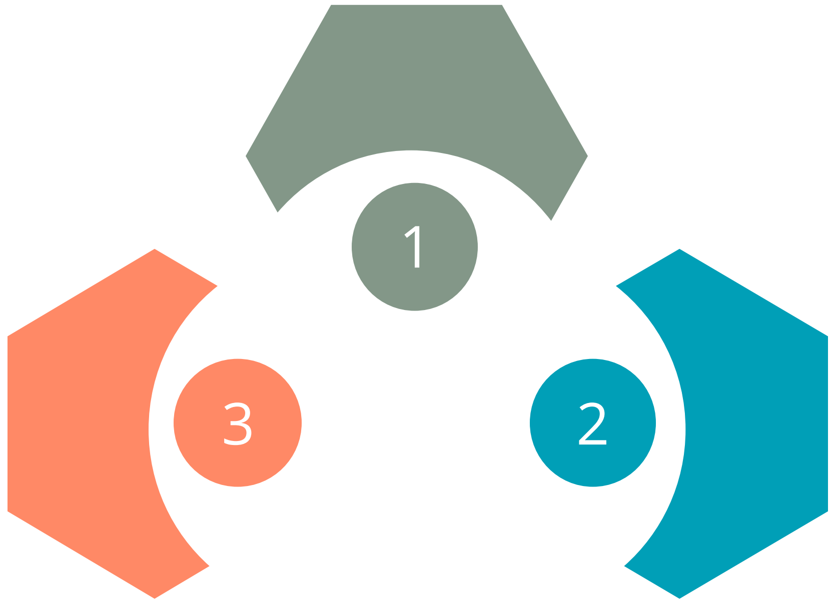
Higher the purity, lower the Gini.

N = 15
C_1 = 15
C_2 = 0

Gini = 1 – $((15/15)^2 + (0/15)^2)$ = 0

Advantages and Disadvantages

Very intuitive and easy to explain



Handles both categorical and numerical data easily

Resistant to outliers

Decision trees are prone to overfitting.

simplilearn

Random Forest

©Simplilearn. All rights reserved.

simplilearn

Bootstrap Aggregation

Bootstrap Aggregation or bagging is a class of ensemble learning. The essential idea in bagging is to average many noisy but approximately unbiased models, and hence reduce the variance. Trees are ideal candidates for bagging since they can capture complex interactions.

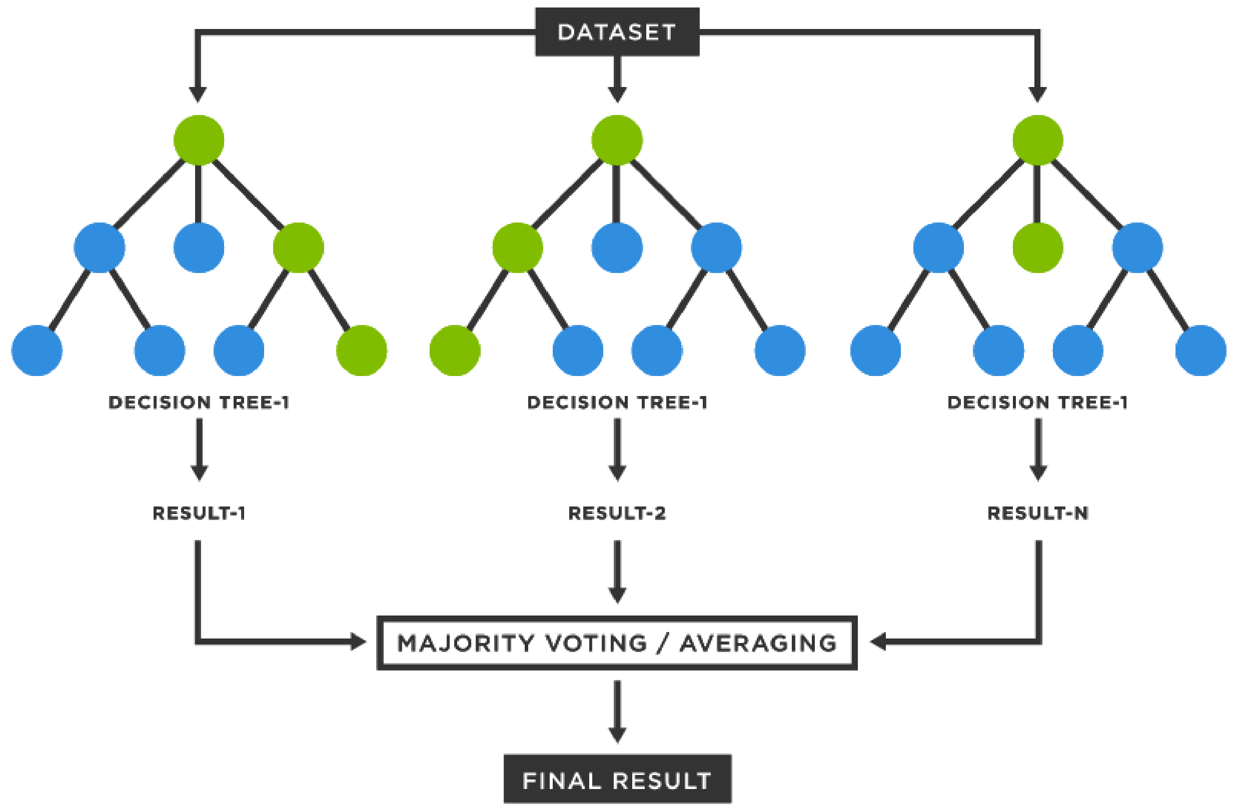
Ensemble refers to the use of multiple models to achieve better performance than could be expected by any of the constituent models.

simplilearn

Random Forest

Random forest is a bagging technique that combines multiple decision trees for the model building.

It helps to overcome the issue of overfitting. This helps to achieve great accuracy in the performance of the model.



Steps in Random Forest

Step 1: N Bootstrapped samples are created. Here random sampling with replacement takes place.

Customer Data

ID	Amount	Credit Score	Income
1	$14000	750	$1000
2	$2000	720	$1200
3	$2000	730	$1250
4	$12000	700	$5500
5	$70000	570	$5000
6	$60000	850	$12000
7	$55000	600	$1190
8	$30000	750	$5398
9	$30000	620	$15000
10	$18888	750	$5200
11	$18000	650	$1380
12	$25000	700	$1280

n = 12

Bootstrapped Data – Sample 1

ID	Amount	Credit Score	Income
1	$14000	750	$1000
2	$2000	720	$1200
3	$2000	730	$1250
4	$12000	700	$5500
5	$70000	570	$5000
6	$60000	850	$12000
7	$55000	600	$1190
8	$30000	750	$5398
9	$30000	620	$15000
10	$18888	750	$5200
11	$18000	650	$1380
12	$25000	700	$1280

n = 12

Steps in Random Forest

Step 2: Create ‘N’ decision trees for ‘N’ bootstrapped samples. In a decision, only a random subset of ‘m’ variables out of total ‘p' variables are used at each node to decide the splitting of each node.

Step 3: Each record is classified based on N decision trees. The final class of each record is decided based on the majority vote.

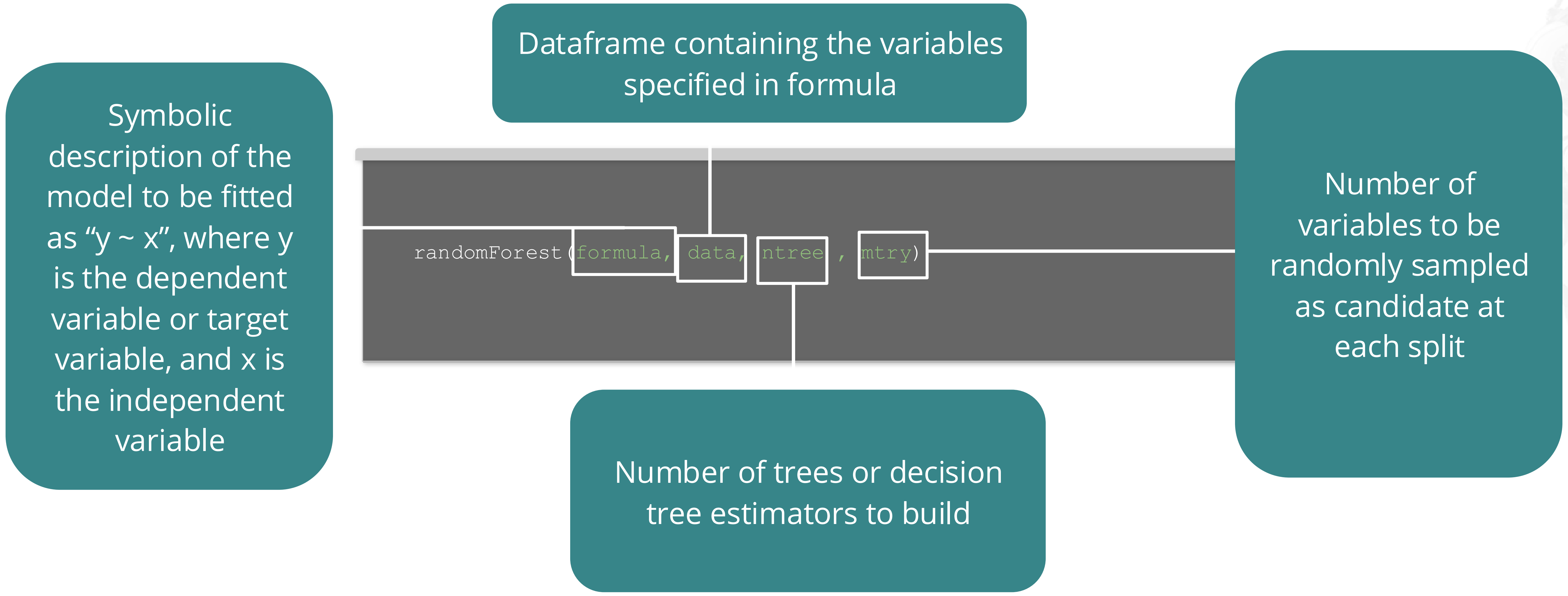
OOB Samples

Records left out in kth bootstrapped sample(bag) are ‘Out of Bag’ samples.

For each observation z_i, construct its Random Forest predictor by averaging only those trees corresponding to bootstrap samples in which z_i did not appear.

Random Forest in R

Random Forest Model can be implemented in R using randomForest () function of the "randomForest" package.



```
randomForest(formula, data, ntree , mtry)
```

Symbolic description of the model to be fitted as "y ~ x", where y is the dependent variable or target variable, and x is the independent variable

Dataframe containing the variables specified in formula

Number of trees or decision tree estimators to build

Number of variables to be randomly sampled as candidate at each split

DATA AND
ARTIFICIAL INTELLIGENCE

Naïve Baye’s Classification

©Simplilearn. All rights reserved.

Bayes Theorem

Bayes Theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

$$P\,(Class_j\,|\,x) = \frac{P(x|Class_j) * P(Classj)}{\sum_{j=1}^{c} P(x|Class_j) * P(Classj)}$$

Where,

- P (Classj $|x$) - Posterior Probability of class
- $P(x|\,Classj)$ - Likelihood, probability of predictor given class
- $P(Classj)$ - Class Prior Probability
- $P(x)$ - Predictor Prior Probability

simplilearn

Bayes Theorem: Simplified

Bayes theorem can be simplified using probability theory:

$$P\,(Class_j \mid x) = \frac{P(x|Class_j) * P(Classj)}{\sum_{j=1}^{c} P(x|Class_j) * P(Classj)}$$

where c is the number of classes.

Using the assumption of independence, the numerator for predictors x_1, x_2.. can be expanded.

$$P(x|Class_j) * P(Classj) = [P(x_1|Class_j) * P(x_2|Class_j) * P(x_3|Class_j) * P(x_4|Class_j) \ldots] * P(Classj)$$

Note

A denominator is effectively a constant for a given data. In practice, the numerator is compared.

Baye’s Theorem: Example

Given the previous patients doctor has seen, should he believe that the patient with the following symptoms has the flu?

Chills	runny nose	headache	fever	flu
Y	N	Mild	Y	N
Y	Y	No	N	Y
Y	N	Strong	Y	Y
N	Y	Mild	Y	Y
N	N	No	N	N
N	Y	Strong	Y	Y
N	Y	Strong	N	N
Y	Y	Mild	Y	Y

Chills	runny nose	headache	fever	flu
Y	N	Mild	Y	?

Find:

$P\,(Class_1 | x) =$ **P (flu=Y | x)** and

$P\,(Class_1 | x) =$ **P (flu=N | x)**

to solve this problem.

simplilearn

Calculations

Calculate likelihood of x in the given class:

$(x|Class_j) = P(x_1|Class_j) * P(x_2|Class_j) * P(x_3|Class_j) * P(x_4|Class_j)$

P (x |flu=Y) =

$P(chills = Y|$ flu=Y) * $P(\text{runny nose} = N|$ flu=Y) * $P(\text{headache} = mild|$ flu=Y) *
$P(\text{fever} = Y|$ flu=Y)

$= \frac{3}{5} * \frac{1}{5} * \frac{2}{5} * \frac{4}{5}$ $\Rightarrow$ P (x |flu=Y) = .0384

Similarly $\Rightarrow$ P (x |flu=N) = .0246

Calculations

Using the data, these can be found:

P (flu=Y) = 5/8 = 0.625 and P (flu=N) = 3/8 = 0.38

Hence,

- Numerator of ***P*** **(flu=Y | x)** = .0384 * .625 = .024 and
- Numerator of ***P*** **(flu=N | x)** = .024 * .38 = .009 using Baye's theorem
- Since denominator remains constant ***P*** **(flu=Y | x)** > ***P*** **(flu=N | x)** And therefore, the doctor should believe that the patient with the given symptoms has the flu.

Model Evaluation

©Simplilearn. All rights reserved.

Evaluating Classification Models

A confusion matrix is used to evaluate the performance of a classification model on a set of test datapoints for which the actual values are known.

ID	Actual class	Predicted class	Result
1	0	0	TN
2	0	0	TN
3	1	0	FN
4	0	0	TN
5	1	1	TP
6	1	0	FN
7	0	1	FP
8	1	0	FN
9	1	1	TP
10	1	1	TP

1 is positive class

Actual Values

Predicted Values

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Important Metrics

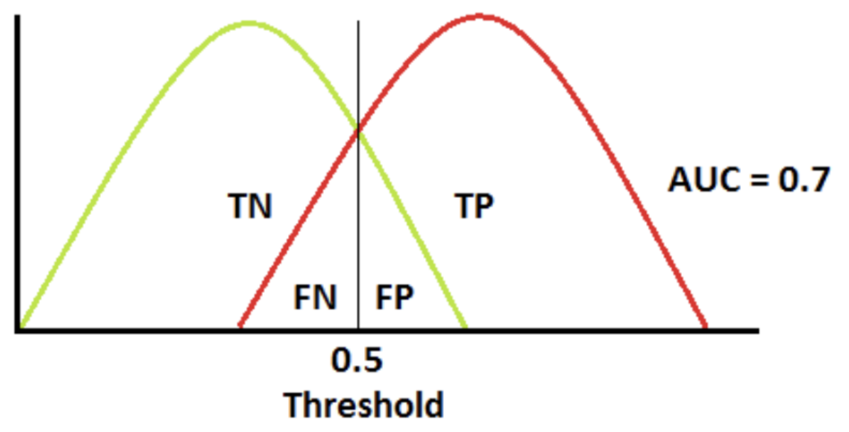
Accuracy = $\frac{(TP+TN)}{(P+N)}$	Positive (P)	Negative (N)	
Positive (PP) (Predicted Positive)	True Positive	False Positive	Precision = $\frac{TP}{PP}$
Negative (PN) (Predicted Negative)	False Negative	True Negative	False omission rate $= \frac{FN}{PN}$
	Sensitivity/recall $= \frac{TP}{P}$	False positive rate $= \frac{FP}{N}$	

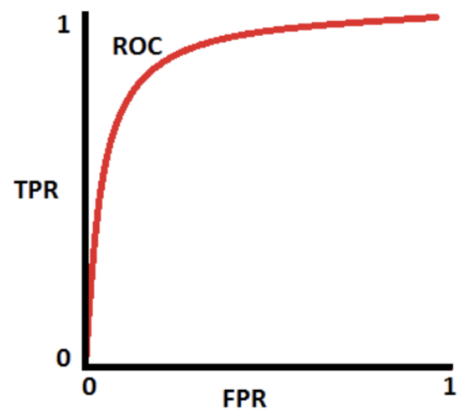
ROC Curve

ROC curve is a performance measurement metric for the classification problems.

ROC curve is obtained by plotting a graph between TPR and FPR at different thresholds for predicted probabilities.

AUC is the area under ROC curve and helps to measure the distinguishing power of the model. Means, higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.





simplilearn