L01: What is Machine learning and when to use ML?

Machine learning overview:

- Humans provide explicit instructions to computers, but in machine learning, the system learns from data and improves its performance over time.
- It focuses on the use of data and algorithms to imitate the way that humans learn.
- Data Driven: Gradually improving its accuracy over Data & Time.

model1 (Algo) $\rightarrow m_2 \rightarrow m_3 \rightarrow m_4$ (data-driven)

- ML tries to find the general feature/formula of the data (because sometimes we can't set the rules or conditions of data) to reach the target/prediction.
- The more diverse and accurate the training data, the easier it is for the machine to find patterns and the more accurate the outcome.

L02: Types of ML

Label:

- A label is a description that informs an ML model what a particular data represents so that it may learn from the example. For an image, this might be telling a model that there is a cat or dog or person or a tree. For an audio recording, an annotator writes the words that are being said.
- Data labeling/Data annotation: It is the process of adding labels to raw data to show the ML model the desired responses that it should be able to forecast.

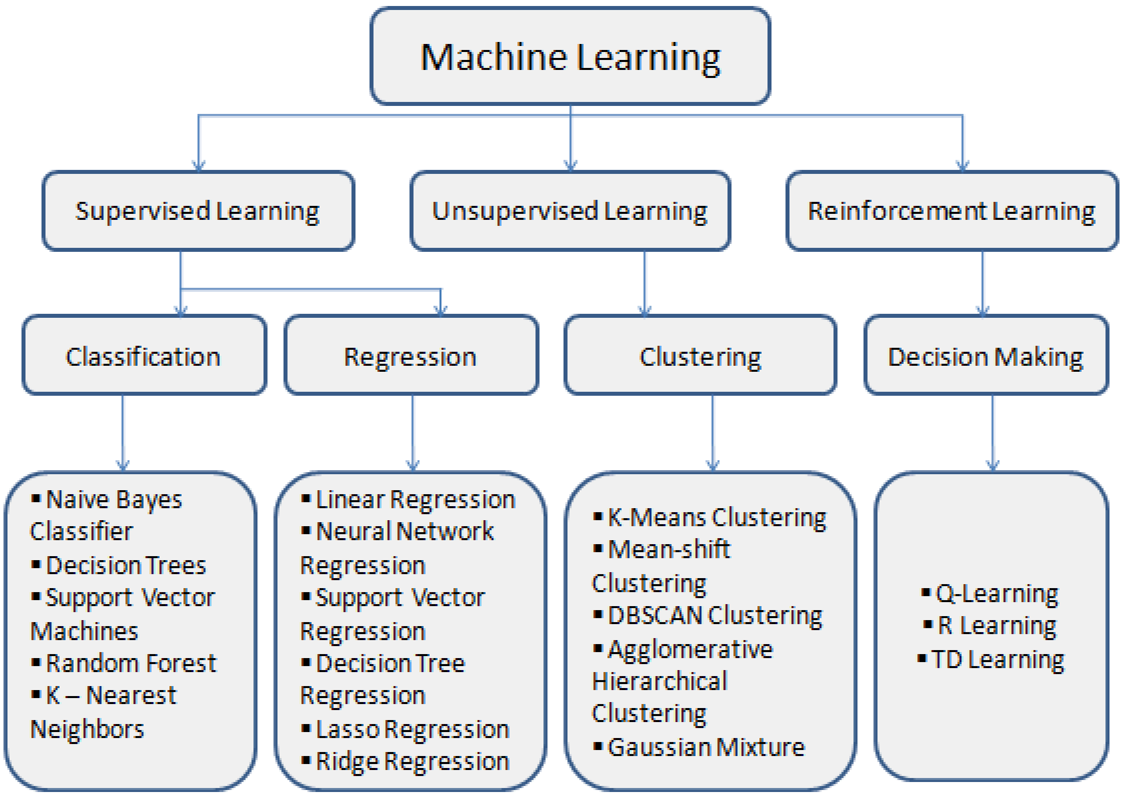
Feature:

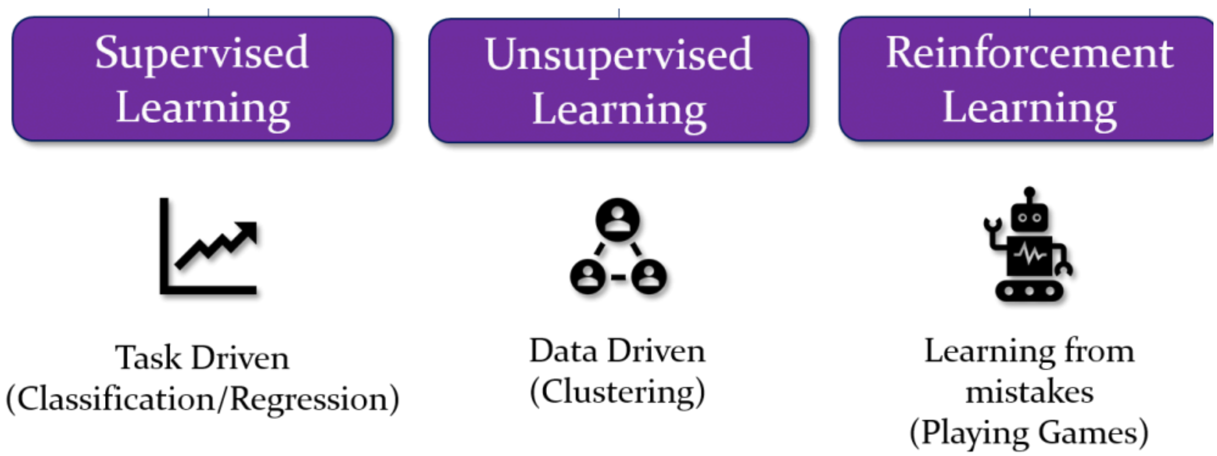
- A feature refers to an individual measurable characteristic or property of an object that is being observed from Data. Features distinguish one object from another from Data.
- For example, in image, features usually include patterns, colors, and shapes that are present in images.
- Dog corresponding breeds (labels) & Dog characteristics (features).
- A label represents an ***output*** value, while a feature is an ***input*** value that describes the *characteristics of such labels* in datasets.
- The more and better the labeled training dataset was, the better the model will predict labels by features.

Note:

- Input variable: It represents the input variable/feature and is usually denoted as x.

- Target: It is the final output which the model is trying to predict. It can be categorical or continuous.
- m is the number of training examples.
- (x, y) is a single training example.
- x^i, y^i represents the i^th training example.





There are generally 4 types of ML.

- Supervised learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

Supervised Learning:

- Classification: The goal is to predict categorical/discrete values, for example, {Yes/No}, {1/0}, {True/False}, {spam/not spam}. **Target known & Discrete.**
- Regression: The goal is to predict continuous values, e.g. home prices. **Target unknown & continuous.**

Applications	Input (X)		Output (Y)
Spam Filtering	Email	→	Spam (0/1)
Speech Recognition	Audio	→	Text transcripts
Machine Translation	English	→	Spanish
Online Advertising	Ad, User info	→	Click (0/1)
Self-driving Car	Image, radar info	→	Position of other cars
Visual Inspection	Image of phone	→	Defect (0/1)

Unsupervised Learning:

- Input data is not labeled and does not have a known result. Here, the ML might be able to teach us new things after it learns patterns in data, because in this type of dataset, human experts don't know what to look for in the data.
- There are no output categories or labels. Data only comes with inputs x, but not output labels y. Unsupervised learning is basically to find some structure in the data.
- Algorithms do not require a ***training data set***. That is, the data labeling is not necessary, the machine does not need a human to guide it, and it tries to find any patterns in the provided data on its own.

Clustering:

- Clustering involves arranging objects into relatively homogeneous groups. It separates data based on their characteristics that are similar to the machine.

Association:

- It is used to find sets of characteristics, and their values, which are frequently encountered in the feature descriptions of objects. Another name for this approach is called rule-finding - a machine analyzes a dataset and finds features that occur together frequently.

L03 + L04: Linear Regression (Part 01+Part 02)

Hypothesis: A hypothesis is a function that best describes the target in machine learning. It depends upon the data.

Loss function / error function:

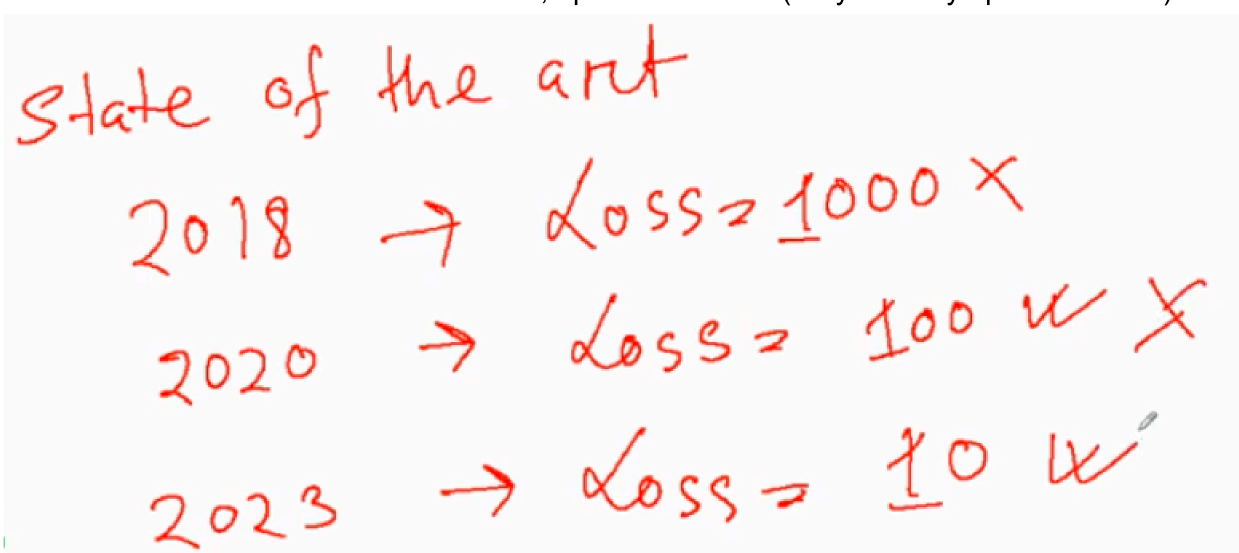
- The loss function is the function that computes the distance between the current output and the expected output of the algorithm.
- Loss function can be categorized into two groups. One for classification (discrete values, 0,1,2…) and the other for regression (continuous values).
- Loss function and cost function are used interchangeably.
- Loss low = Prediction high = Accuracy high = Good model.

Accuracy:

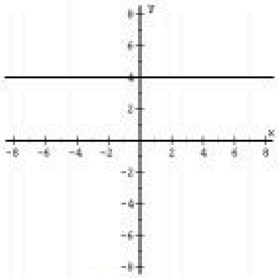
- Formula for Accuracy = Total Number of Predictions / Number of Correct Predictions
- Accuracy is 87.5% means that the model correctly predicted the class for 87.5% of the instances in the dataset.

STATE OF THE ART:

- The latest and highest or advanced stage of a technology or science. Something that's state of the art is the most modern, updated version. (Very recently updated model)

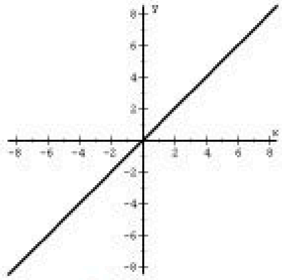


PARENT FUNCTIONS



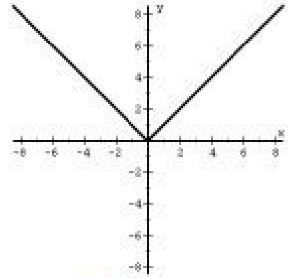
$f(x) = a$

Constant



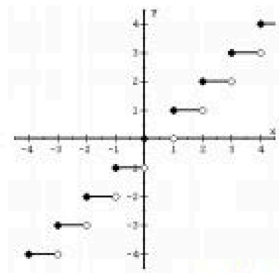
$f(x) = x$

Linear



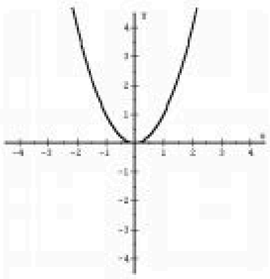
$f(x) = |x|$

Absolute Value



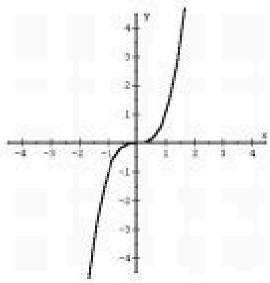
$f(x) = \operatorname{int}(x) = [x]$

Greatest Integer



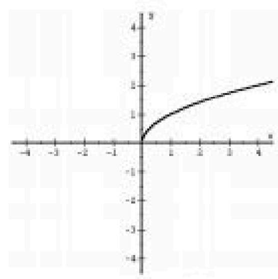
$f(x) = x^2$

Quadratic



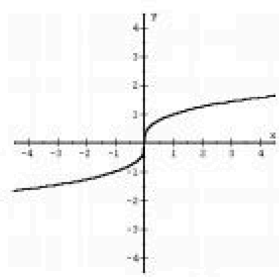
$f(x) = x^3$

Cubic



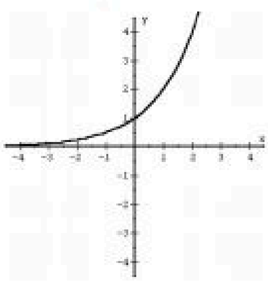
$f(x) = \sqrt{x}$

Square Root



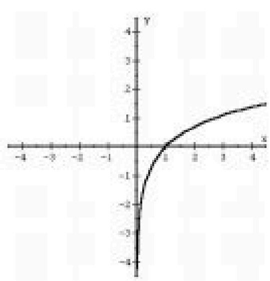
$f(x) = \sqrt[3]{x}$

Cube Root



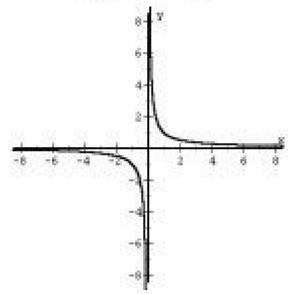
$f(x) = a^x$

Exponential



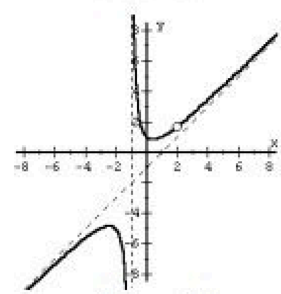
$f(x) = \log_a x$

Logarithmic



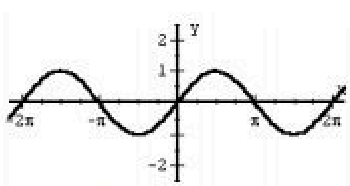
$f(x) = \dfrac{1}{x}$

Reciprocal

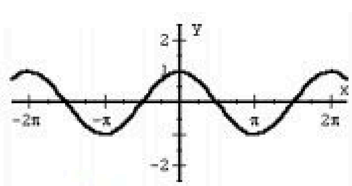


$f(x) = \dfrac{(x^2+1)(x-2)}{(x+1)(x-2)}$

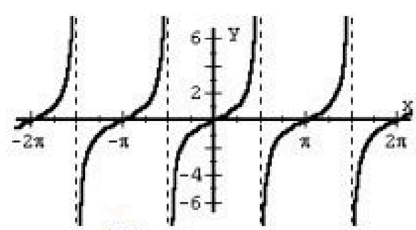
Rational



$f(x) = \sin x$



$f(x) = \cos x$



$f(x) = \tan x$

Trigonometric Functions

Q/A: When a function is linear or not?
Linear Function:

- A linear function is a polynomial function of degree 1.
- General form is: f(x)=ax+b like f(x)=2x+3
- The graph of a linear function is a straight line.

Quadratic Function:

- A quadratic function is a polynomial function of degree 2.
- General form is: $f(x) = ax\^{}2 + bx + c$
- The graph of a quadratic function is a parabola.

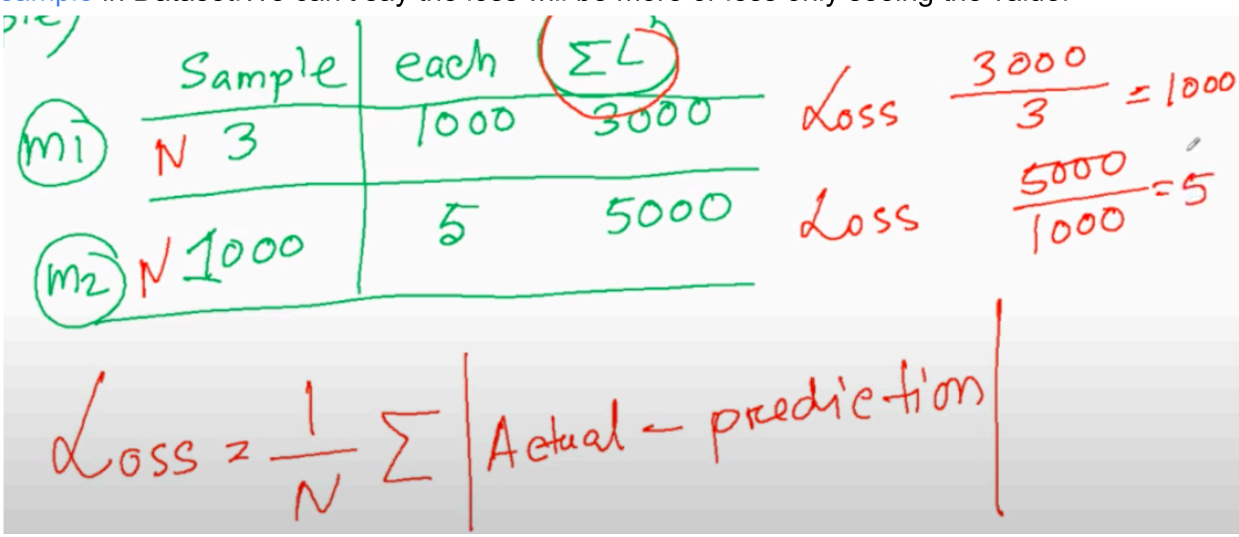
Data point, near the Hypothesis line
Case 01:
Case 02:

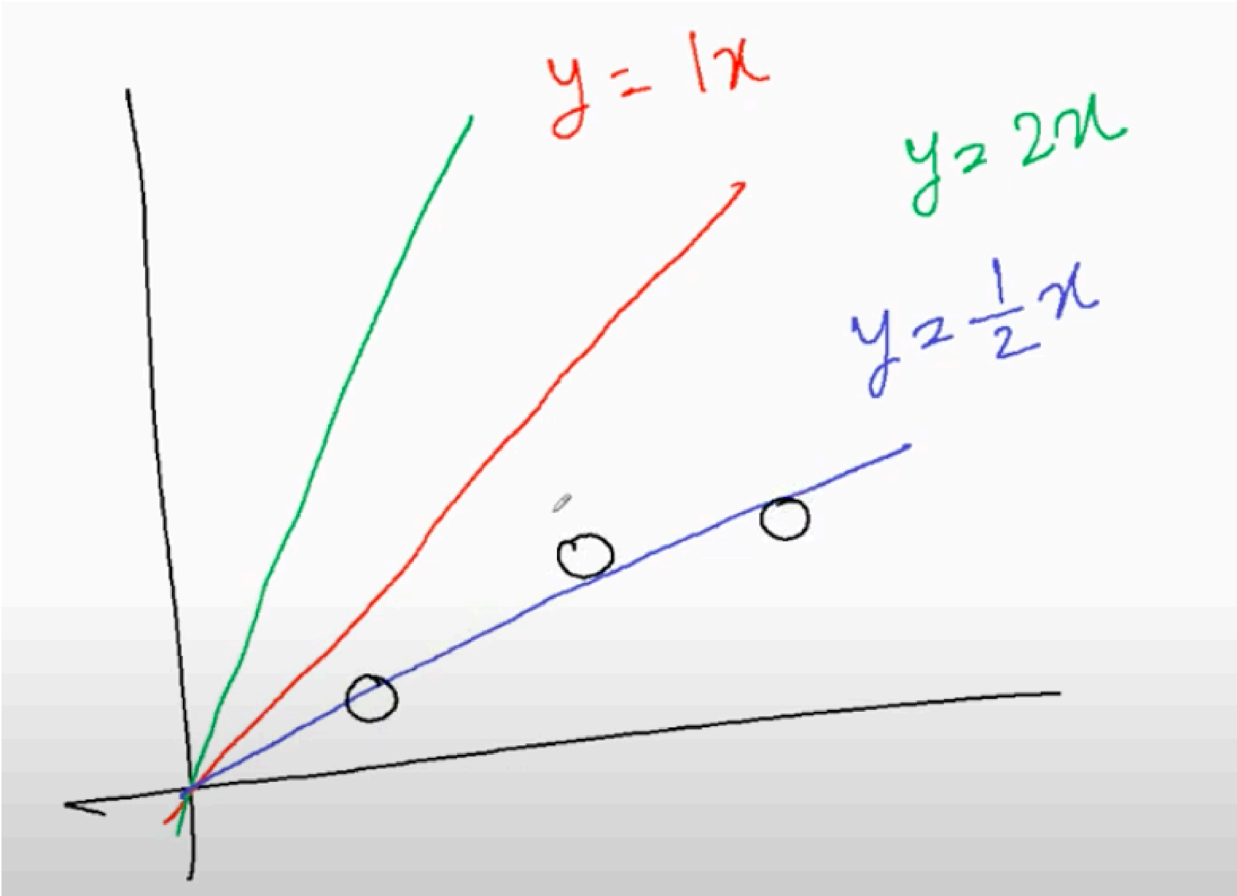
Loss function minimization:

To minimize the loss function in linear regression, we typically use an optimization algorithm such as gradient descent. The goal is to find the values of the model parameters (in this case, the weight w) that minimize a chosen loss function.

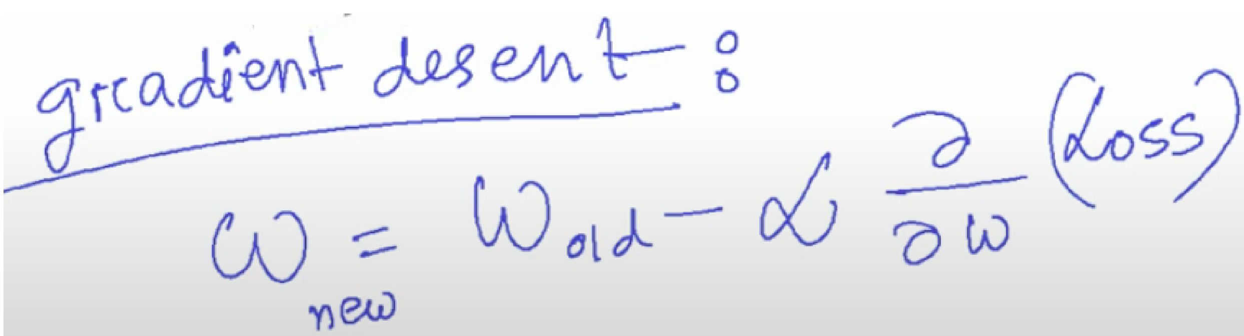
Q/A: y = wx (like R = 20S)
Here, y = target, w = weight, x = feature.
So, we can modify **only weighted values** because feature and target value are constant. So modify the weight value to minimize the loss function.

- Another thing is, loss function of a model depends on the number of samples and size of sample in Dataset.We can't say the loss will be more or less only seeing the value.





Gradient descent:
Gradient descent is an iterative optimization algorithm used to minimize a loss function for training models.



L05: Regression Evaluation Metrics

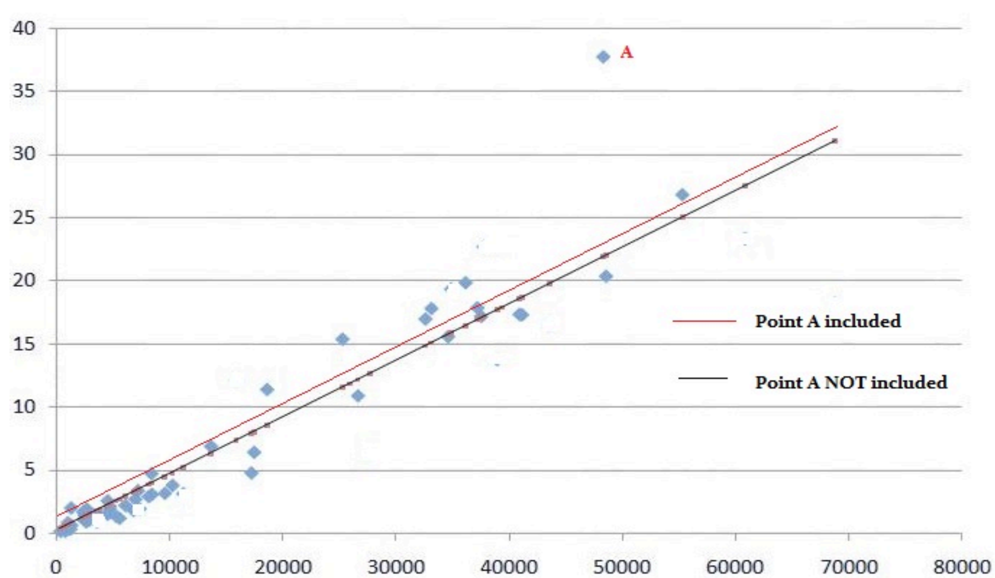
How well a model fits the actual data is typically assessed using various evaluation metrics. The choice of the metric depends on the type of task (regression or classification) and the specific goals of the model.

Regression Tasks:

1. Mean Absolute Error (MAE):
    - Measures the average absolute differences between predicted and actual values.
2. Mean Squared Error (MSE):
    - Measures the average squared differences between predicted and actual values.
    - The lower the MSE, the better the model predictive accuracy, and, the better the regression model is.
3. Root Mean Squared Error (RMSE):
    - The square root of MSE provides a measure of the average magnitude of errors.
4. R^2 (squared): Indicates the proportion of the variance in the target variable that is explained by the model.

Classification Tasks:

1. Accuracy:
    - Measures the ratio of correctly predicted instances to the total number of instances.
2. Precision, Recall, and F1 Score:
    - Useful for imbalanced datasets; precision focuses on the accuracy of positive predictions, recall on the ability to capture all positive instances, and F1 Score is the harmonic mean of precision and recall.



Mean Absolute Error (MAE)

Date : 17 02 2024

$$\boxed{MAE = \frac{1}{N} |y - \hat{y}| \quad \text{or} \quad \Sigma \frac{1}{N} |y - \hat{y}|}$$

Here, y = actual value (actual price)

$\hat{y}$ = predicted value (by model)

$\bar{y}$ = average of all actual value (avg. of all y)

Q/A:

Size	Price (y)	Prediction ($\hat{y}$)
1000	20000	21000
1200	25000	27000
1800	35000	31000

(i) $MAE = \Sigma \frac{1}{N} |y - \hat{y}|$

$$= \frac{1000 + 2000 + 4000}{3} \quad (\because N = 3, \text{ so 3 size})$$

$$= 2333.33$$

Mean Square Error (MSE)

Date: 17 02 2024

$$\boxed{MSE = \Sigma \frac{1}{N} (y - \hat{y})^2}$$

Q/A: For the following question,

$$MSE = \frac{(1000)^2 + (2000)^2 + (4000)^2}{3}$$

$$= 7000000$$

Note: For square, here we don't have to take any absolute operation, because neg. value will be vanished for square.

$(-2)^2 = 4$ or $|-2| = 2$

Note: Small value of metrics means, model's predictions are closer to the actual values, indicating better accuracy and performance.

Route Mean Square Error (RMSE)

Date : 17 02 2024

$$\boxed{\begin{aligned} RMSE &= \sqrt{MSE} \\ \text{or, } &\sqrt{\Sigma \frac{1}{N} (y-\hat{y})^2} \end{aligned}}$$

Q/A: For the following question,

$$RMSE = \sqrt{\frac{1000^2 + 2000^2 + 4000^2}{3}}$$

$$= 2645.75$$

Note: The value of MAE, MSE, RMSE, or any regression evaluation metrics is smaller, these indicates the ~~m~~ better performance of model. We want to minimize the value of MAE, MSE, RMSE or any metrics.

R^2 (Coefficient of determination)

Date: 17 02 24

$$\boxed{R^2 = 1 - \frac{\Sigma (y - \hat{y})^2}{(y - \bar{y})^2}}$$

y = actual value
$\hat{y}$ = prediction "
$\bar{y}$ = avg value

or

$$\boxed{R^2 = 1 - \frac{\text{sum of Squared residuals (SSR)}}{\text{Total sum of squares (TSS)}}}$$

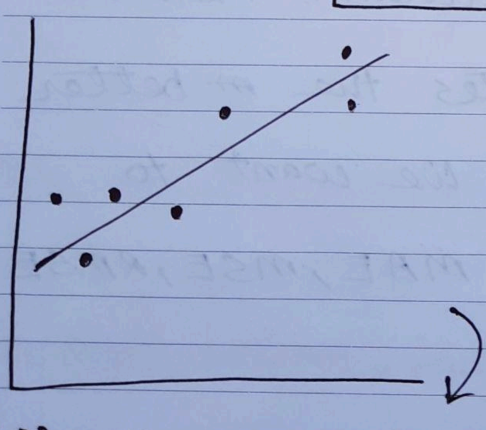
$\because \bar{y} = \frac{20000 + 25000 + 35000}{3}$

$= 26666.67$

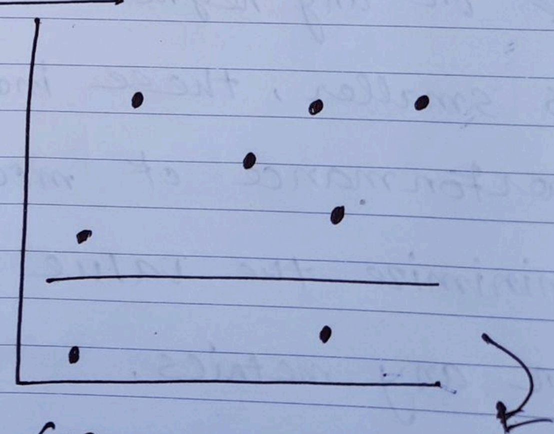
$R^2 = 0$; model doesn't explain the variability of target variable

$R^2 = 1$; model perfectly explain the variability of target variable.

$$\boxed{0 \leq R^2 \leq 1}$$



(R^2 value close to 1.)



(R^2 value close to zero)

Date : 17 02 2024

Q/A: what is value of model fitness? (via R^2)

Soln:

$\therefore \bar{y} = 20000 + 25000 + 35000/3$

$= 26666.67$

$\therefore (y-\bar{y})^2$ on Total sum of squares (TSS):

$= (20000 - 26666.67)^2 + \cancel{20000 - 25}$

$(25000 - 26666.67)^2 + (35000 - 26666.67)^2$

$= 116666666.67$

$\therefore \Sigma (y-\hat{y})^2$ on sum of squared residuals (SSR):

$(1000)^2 + (2000)^2 + (4000)^2$

$= 21000000$

$$\therefore R^2 = 1 - \frac{SSR}{TSS} = 1 - \frac{21000000}{116666666.67}$$

$$\boxed{= 0.82}$$

$R^2 = 0.82$ means the model explains about 82% of the variance in the target variable.

Suppose we have the same dataset:

Actual values (y): [10, 20, 30]

Predicted values ($\hat{y}$): [12, 18, 25]

Calculate Mean Absolute Error (MAE):

$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

$MAE = \frac{|10-12|+|20-18|+|30-25|}{3}$

$MAE = \frac{2+2+5}{3}$

$MAE = \frac{9}{3}$

$MAE = 3$

Calculate Mean Squared Error (MSE):

$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

$MSE = \frac{(10-12)^2+(20-18)^2+(30-25)^2}{3}$

$MSE = \frac{4+4+25}{3}$

$MSE = \frac{33}{3}$

$MSE = 11$

Calculate Root Mean Squared Error (RMSE):

$RMSE = \sqrt{MSE}$

$RMSE = \sqrt{11}$

$RMSE \approx 3.317$

Calculate R-squared (R^2):

$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$

$R^2 = 1 - \frac{33}{140}$

$R^2 \approx 0.764$

So, in this extended example:

- MAE is 3,
- MSE is 11,
- RMSE is approximately 3.317, and
- R^2 is approximately 0.764.

L06: Classification

These are many types of classification tasks. It depends on the specific problem. The choice of classification method and evaluation metrics may vary based on the characteristics of the data and the goals of the application.

ZeroR Classifier (Simplest Possible Classifier):

Classification problem! Date: 17 02 2024

Sr. no	Weather	Outlook	Play or not
1	Hot	sunny	Yes
2	Cold	Sunny	Yes
3	Cold	Rainy	No
4	Hot	sunny	Yes
5	Cold	sunny	Yes

ZeroR Classifier

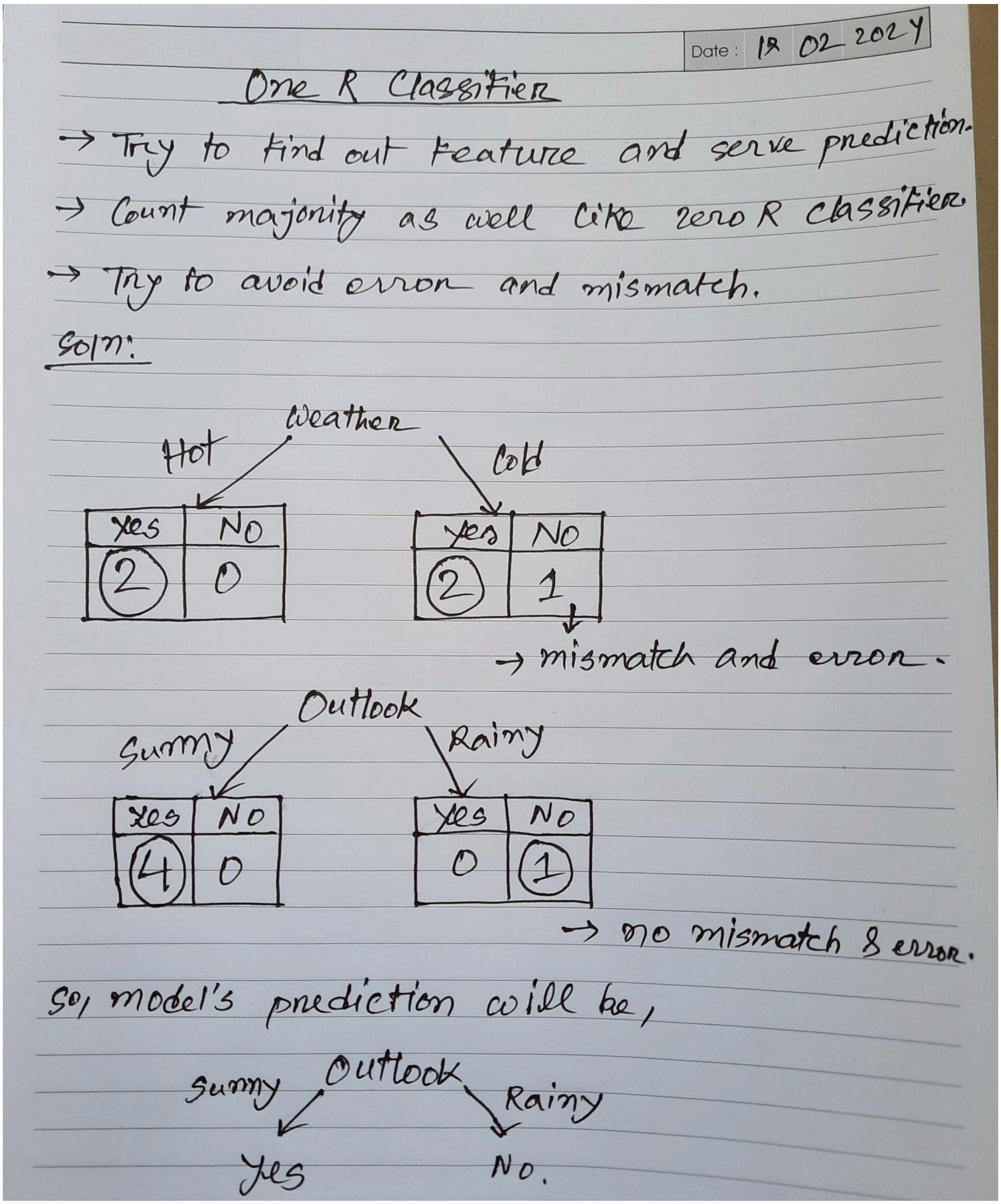
→ Only counts the majority

Soln: Yes → 4 NO → 1

∴ prediction = Yes.

One R Classifier:

- Try to find out a feature (only a single rule) and serve predictions based on this feature. (Generalize and make decisions)
- Count majority as well, like ZeroR Classifier.
- Try to avoid the error and mismatch.

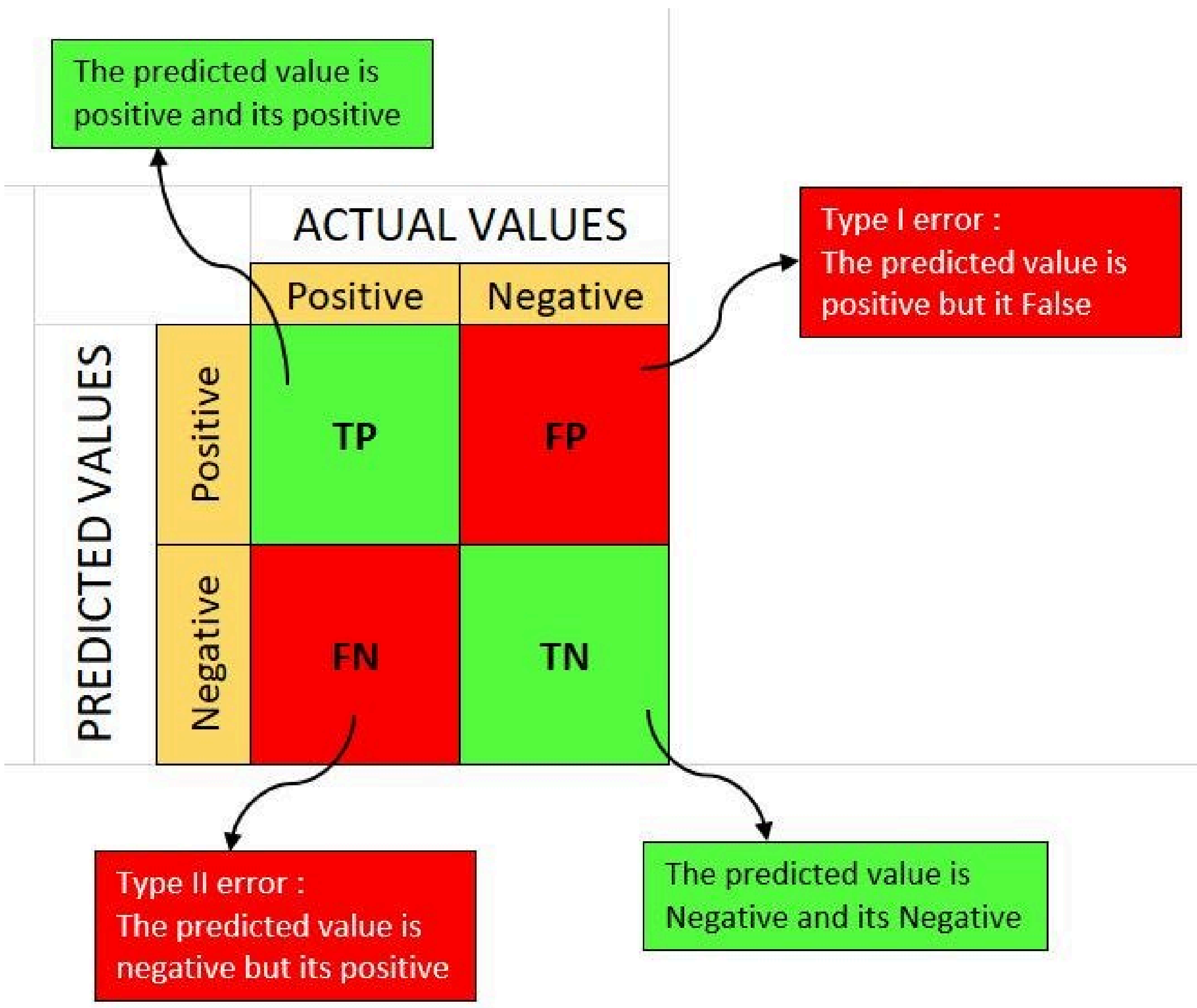


L07+L08: Classification evaluation part 1+2

- Based on only accuracy, we cannot evaluate the classification model's performance.

Confusion matrix:

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm. In machine learning, to measure the performance of the classification model (model's ***Recall, Precision, Specificity, Accuracy, most importantly AUC-ROC curves*** and overall effectiveness in class distinction), we use the confusion matrix.

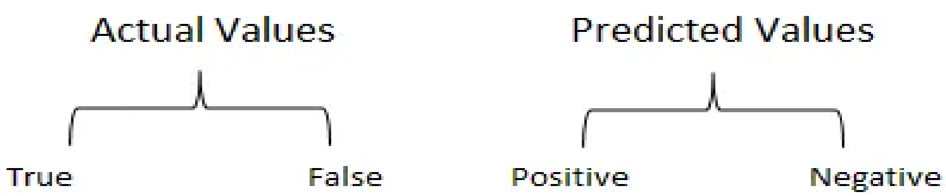


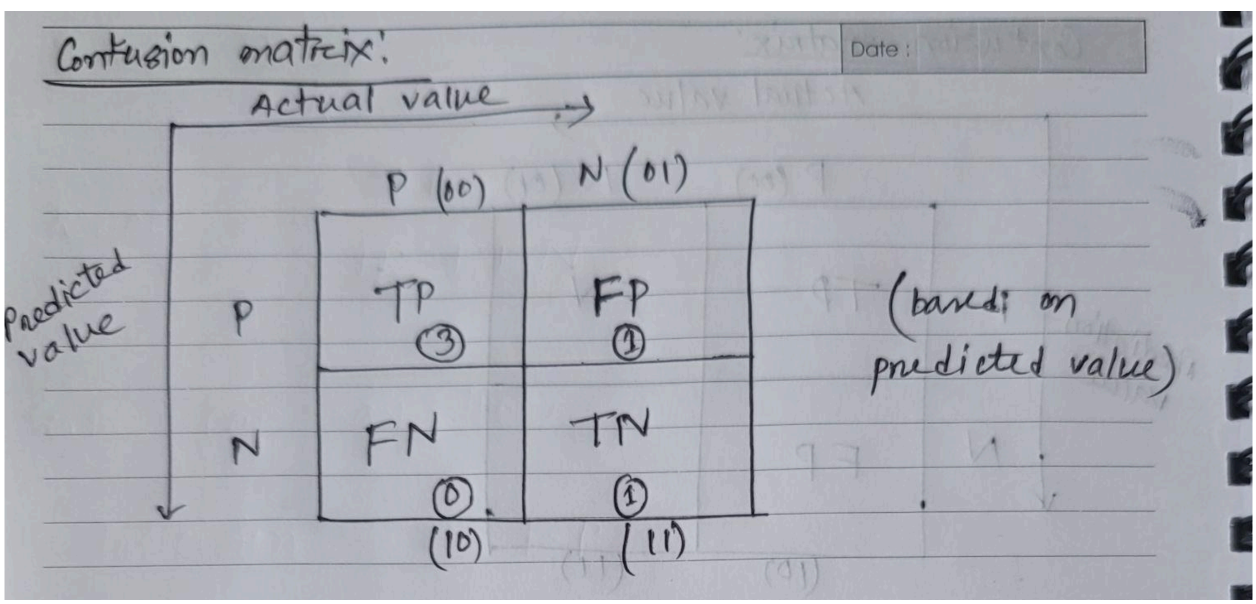
- True positives (TP): occur when the model accurately predicts a positive data point. Interpretation: You predicted positive and it's true. You predicted that a woman is pregnant and she actually is.
- True negatives (TN): occur when the model accurately predicts a negative data point.

Interpretation: You predicted negative and it’s true. You predicted that a man is not pregnant and he actually is not.

- False positives (FP) (Type I error): occur when the model predicts a positive data point incorrectly. Interpretation: You predicted positive and it’s false. You predicted that a man is pregnant but he actually is not.
- False negatives (FN) (Type II error): occur when the model mis-predicts a negative data point. Interpretation: You predicted negative and it’s false. You predicted that a woman is not pregnant but she actually is.

Note: Just remember, we describe actual values as True and False and predicted values as Positive and Negative.





1. **Accuracy:**

Accuracy is used to measure the performance of the model. It is the ratio of Total correct prediction to the total dataset. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{3+1}{5} = 4/5$$

$$= \cancel{80\%}\ 0.8$$

2. **Precision:**

Precision is a measure of how accurate a model's positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model.

Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

Here, $FP = 0$; precision 100%

$FP > 0$; precision < 100%

So, FP ↑ Precision ↓

FP ↓ Precision ↑

$$\therefore \text{precision} = \frac{TP}{TP+FP}$$

$$= \frac{3}{3+1} = \frac{3}{4} = 0.75$$

3. **Recall:**

Recall measures the effectiveness of a classification model in identifying all relevant instances from a dataset. It is the ratio of the number of true positive (TP) instances to the sum of true positive and false negative (FN) instances.

Recall

$$\text{Recall} = \frac{TP}{TP + FN}$$

So, we can say that,

Precision $\longrightarrow$ FP

Recall $\longrightarrow$ FN

$$\therefore \text{Recall} = \frac{TP}{TP + FN}$$

$$= \frac{3}{3+0} = 3/3 = \underline{\underline{1}}$$

Note: We use precision when we want to minimize false positives, crucial in scenarios like spam email detection where misclassifying a non-spam message as spam is costly. And we use recall when minimizing false negatives is essential, as in medical diagnoses, where identifying all actual positive cases is critical, even if it results in some false positives.

4. **F1-Score:**

F1-score is used to evaluate the overall performance of a classification model. It is the harmonic mean of precision and recall.

F_1 Score

$$F_1 \text{ score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

harmonic mean

Note: F_1 score always close to smaller values.

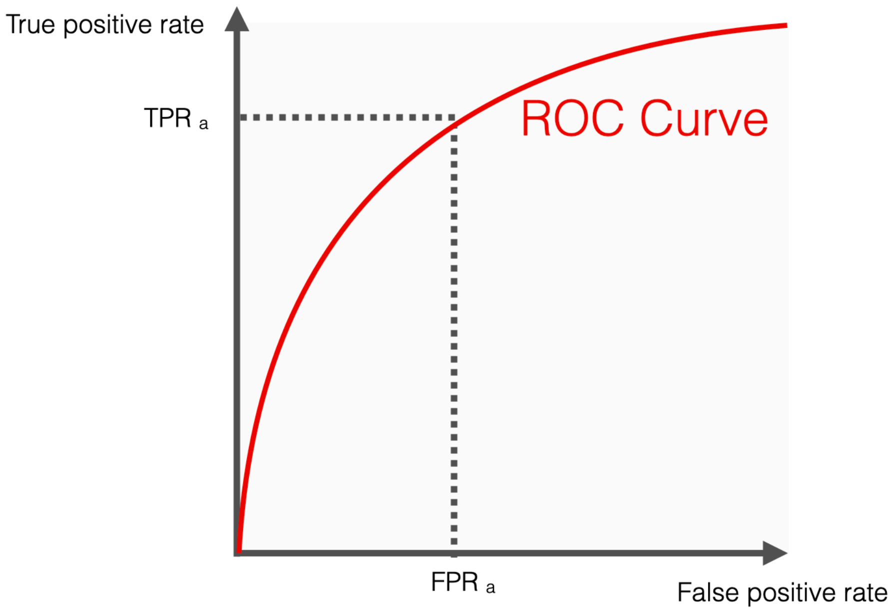
$$\therefore F1 \text{ score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$= \frac{2 \times 0.75 \times 1}{(0.75 + 1)} \doteq 0.85$$

We balance precision and recall with the F1-score when a trade-off between minimizing false positives and false negatives is necessary, such as in information retrieval systems.

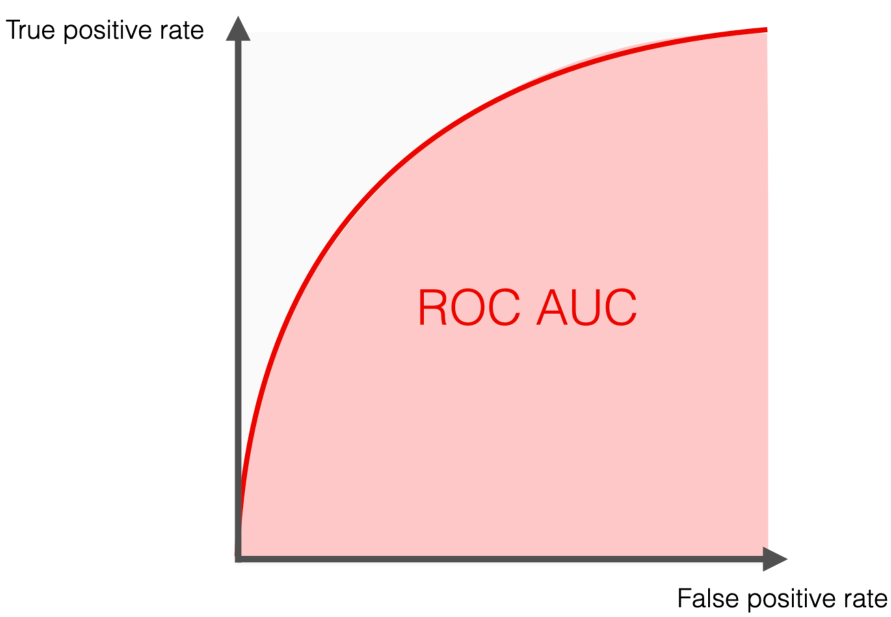
5. ROC curve:

- The curve plots the possible True Positive rates (TPR) against the False Positive rates (FPR).



6. ROC AUC score:

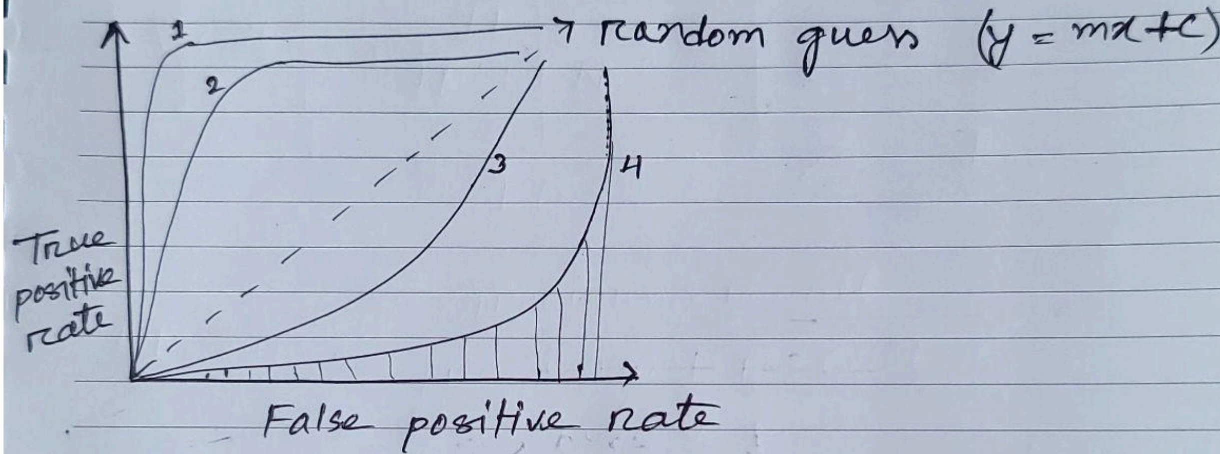
- ROC AUC stands for Receiver Operating Characteristic Area Under the Curve.
- The AUC is a numerical measure that quantifies the overall performance of a model based on its ROC curve. Specifically, AUC-ROC calculates the area under the ROC curve. ROC AUC score shows how well the classifier distinguishes positive and negative classes. It can take values from 0 to 1.
- A higher ROC AUC indicates better performance. A perfect model would have an AUC of 1, while a random model would have an AUC of 0.5.



Note: Check this link: https://www.evidentlyai.com/classification-metrics/explain-roc-curve

ROC curve (Receiver operating Characteristics)

ROC
↓
AUC – ROC (Area under the curve – ROC)



good → 1 > 2] model
worst → 4 > 3] model

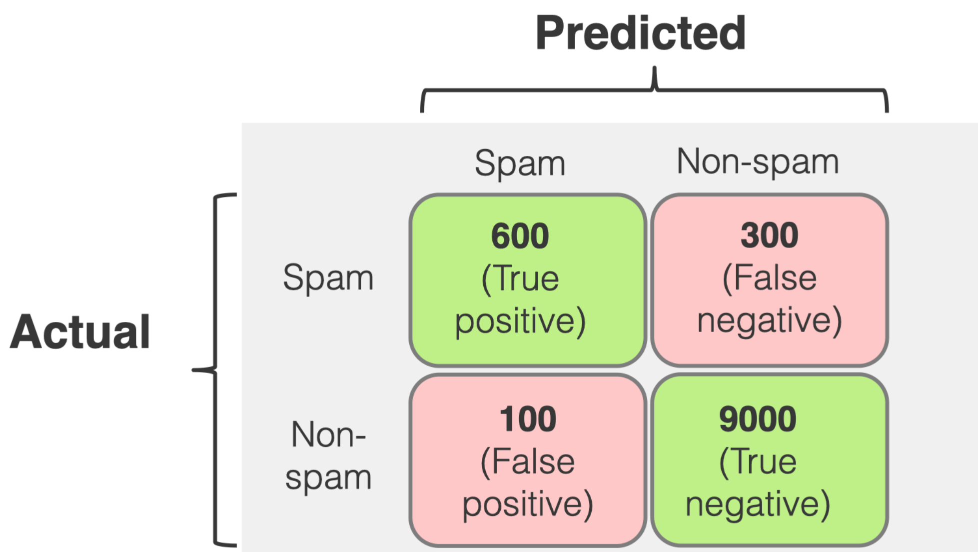
And: AUC

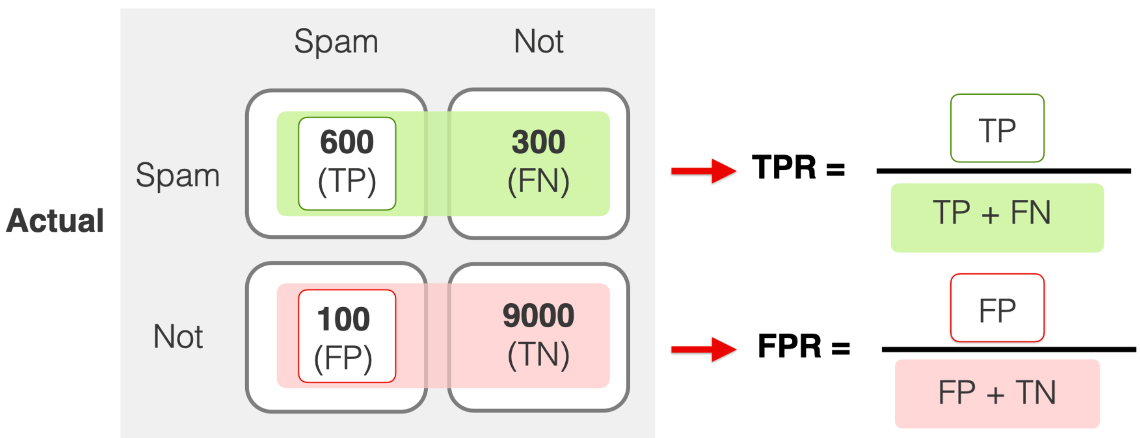
AUC = 90% (For model 1)

AUC = 10% (For model 4)

AUC ↑ → Model ↑

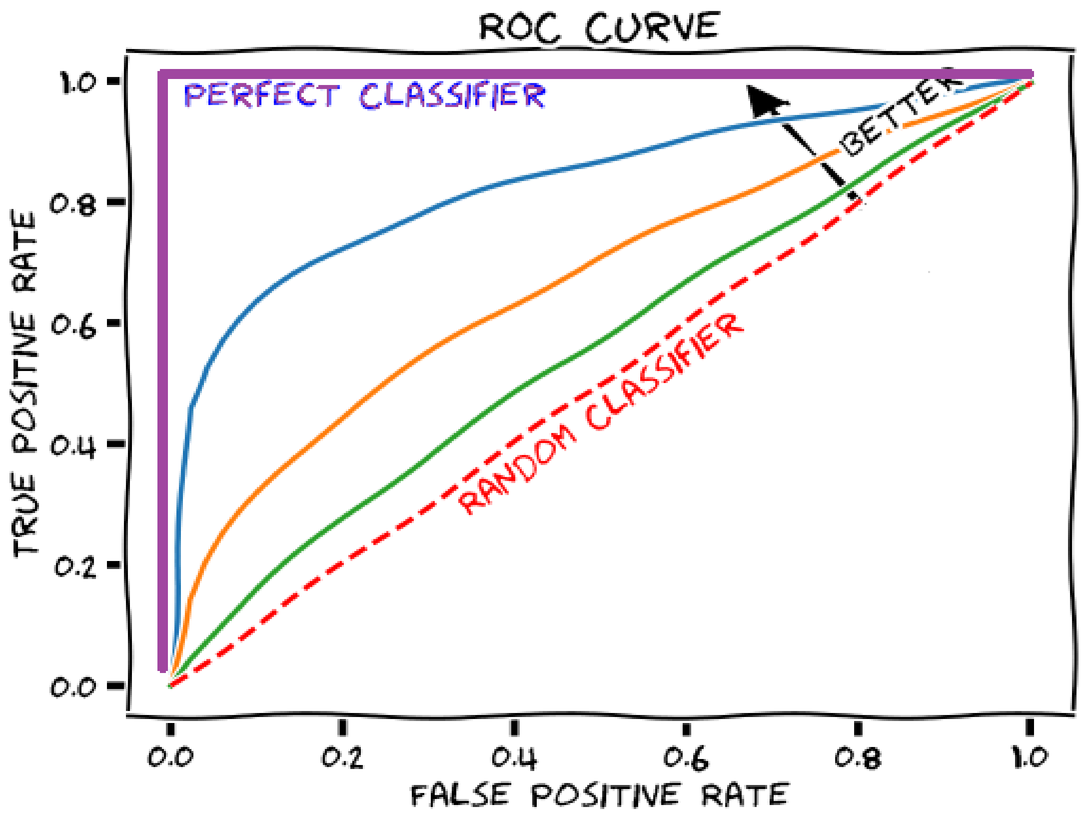
How to measure:



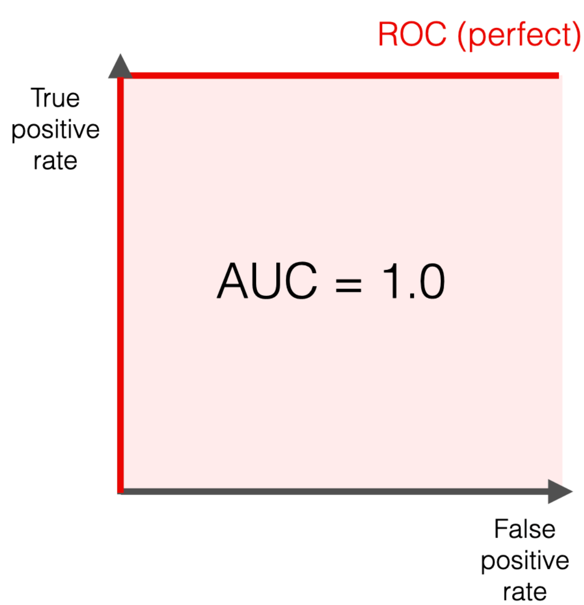


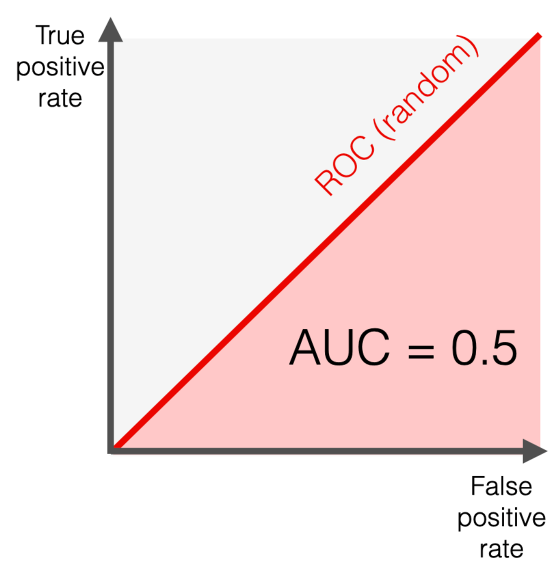
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$



Note: AUC-ROC is a measurement of the goodness of a classification problem. An ideal ROC curve will hug the top left corner, so the larger area under the (ROC) curve the AUC the better the classifier.





L09: Bias-variance overfitting and underfitting

Dataset dividation: For training and testing purposes of our model, we should have our data broken down into 03 distinct dataset splits.

1. **The Training set:**

- It is the set of data that is used to train and make the model learn the hidden features/patterns in the data. In each epoch, the same training data is fed to ***neural network architecture repeatedly***, so the model continues to learn features of the data.
- The training set should have a diversified set of inputs so that the model is trained in all scenarios and can predict any unseen data sample that may appear in the future.

2. **The Validation set:**

- The validation set is a set of data which is separate from the training set that is used to validate our model performance during training. This validation process gives information that helps us tune the ***model's hyperparameters and configurations accordingly***. It is like a critic telling us whether the training is moving in the right direction or not, like showing the right direction.
- The model is trained on the training set, and, simultaneously, the model evaluation is performed on the validation set after every epoch.
- The main idea of splitting the dataset into a validation set is to prevent our model from overfitting.

3. **The Test set:**

- The test set is a separate set of data used to test the model after completing the training.
- Test set must be unseen and different and untrained from the training dataset.

Case 01: If there are several hyperparameters to tune, the machine learning model requires a larger validation set to optimize the model performance. Similarly, if the model has fewer or no hyperparameters, it would be easy to validate the model using a small set of data.

Case 02: If a model use case is such that a false prediction can drastically hamper the model performance—like falsely predicting cancer—it's better to validate the model after each epoch to make the model learn varied scenarios.

Truth 01: There is no optimal split percentage. One has to come to a split percentage that suits the ***requirements and meets the model's needs***.

Truth 02: There are two major concerns while deciding on the optimum split:

- If there is less training data, the machine learning model will show high variance in training. The training set should not be too small; else, the model will not have enough data to learn.
- On the other hand, if the validation set is too small, then the evaluation metrics like accuracy, precision, recall, and F1 score will have large variance and will not lead to the

proper tuning of the model. In general, putting 80% of the data in the training set, 10% in the validation set, and 10% in the test set is a good split to start with.

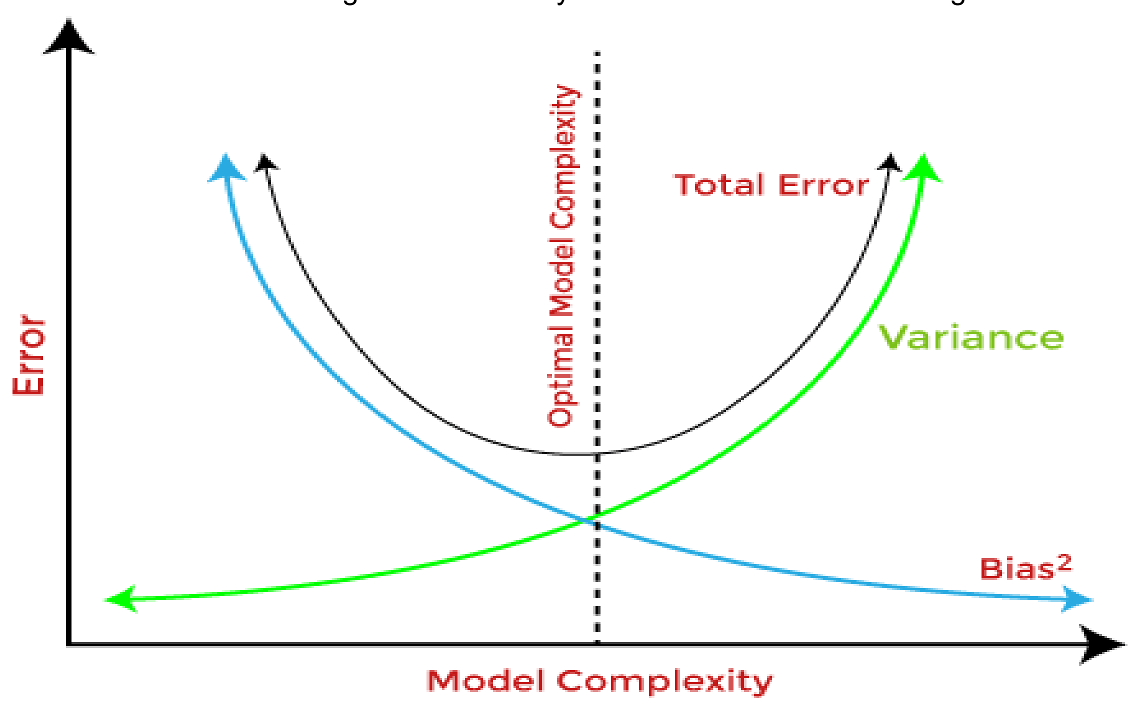
Bias (Inaccuracy):

Bias is simply defined as the inability of the model because of that there is some difference or error occurring between the model's predicted value and the actual value. These differences between actual values and predicted values are known as error or bias error.

- High Bias (Inaccuracy):
    - The model may be too simple, leading to systematic errors and inaccuracies.
    - The model may underfit the data.
- Low Bias (High Accuracy):
    - The model is more complex and can capture underlying patterns in the data.
    - The model tends to generalize well to new, unseen data.

Variance (Inconsistent):

- Inconsistent means there is no consistency, so automatically it refers to high variance.
- Low variance: Low variance means that the model is less sensitive to changes in the training data and can produce consistent estimates of the target function with different subsets of data from the same distribution. This is the case of underfitting when the model fails to generalize on both training and test data.
- High variance: High variance means that the model is very sensitive to changes in the training data and can result in significant changes in the estimate of the target function when trained on different subsets of data from the same distribution. This is the case of overfitting when the model performs well on the training data but poorly on new, unseen test data. It fits the training data too closely that it fails on the new training dataset.

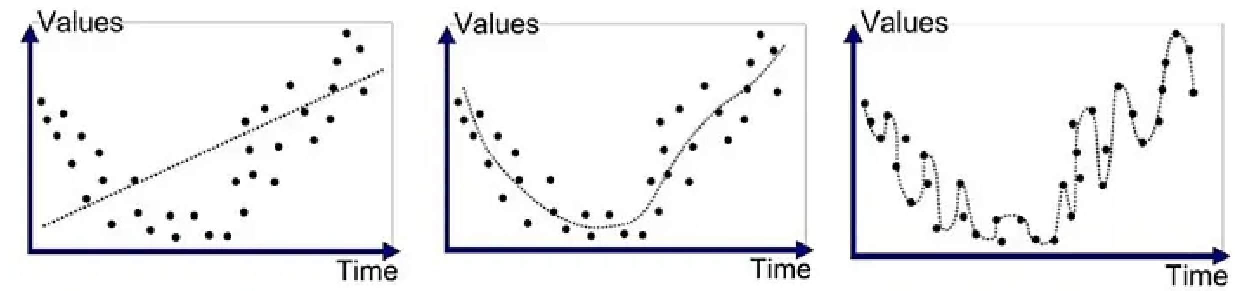


Underfitting:

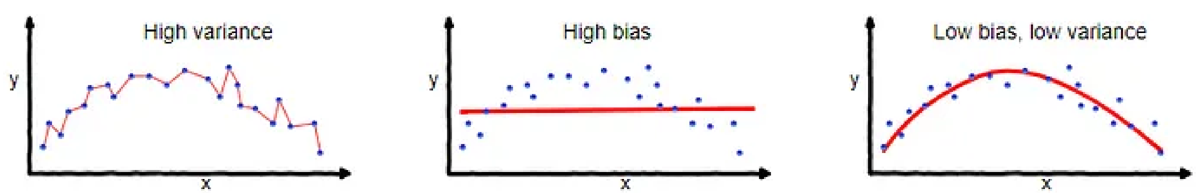
- Underfitting happens when a model is unable to capture the underlying pattern of the data. These models usually have ***high bias and low variance***.
- It happens when we have very little data to build an accurate model or when we try to build a linear model with non-linear data. Also, these kinds of models are very simple to capture the complex patterns in data like Linear and logistic regression.

Overfitting:

- Overfitting happens when our model captures the noise along with the underlying pattern in data. It happens when we train our model a lot over noisy datasets. These models have ***low bias and high variance***. These models are very complex like Decision trees which are prone to overfitting.



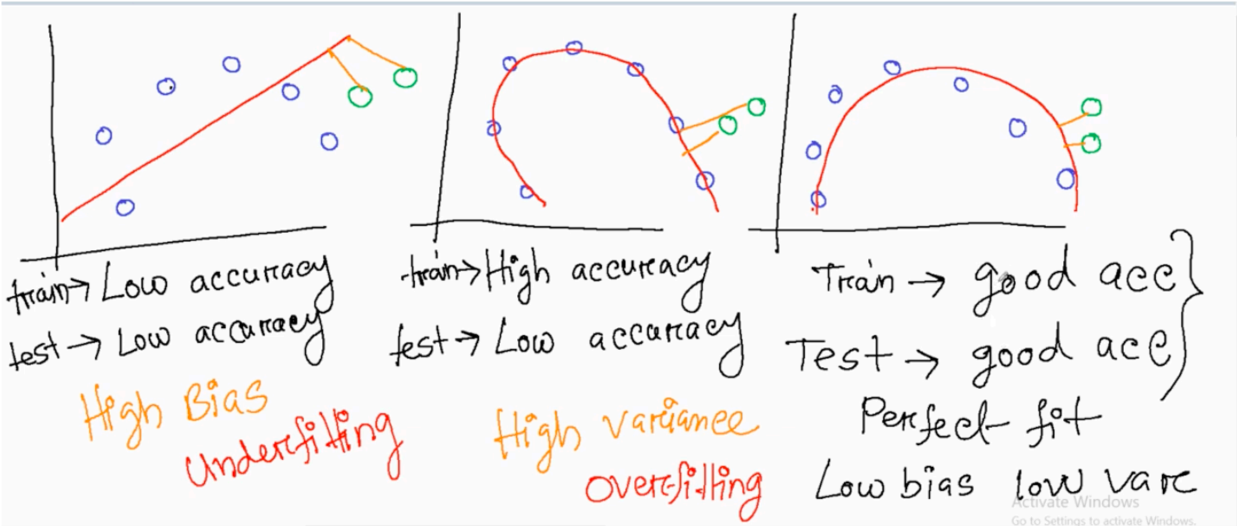
Underfitted　　Good Fit/Robust　　Overfitted



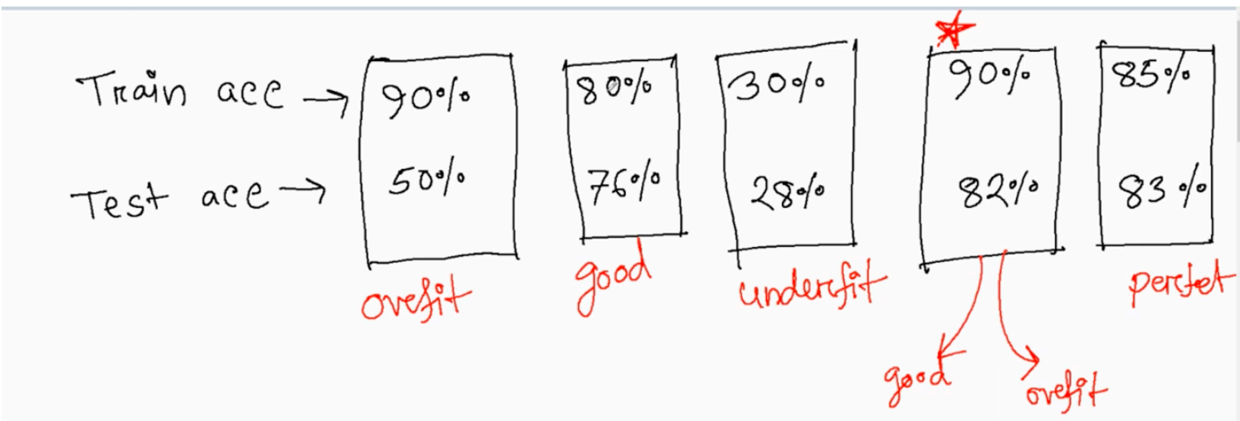
overfitting　　underfitting　　Good balance

Scenario:

- Here, in graph 01, we have low accuracy on both the training set and the test set, it suggests that the model is not performing well overall. It has high bias and low variance.
- And, graph 02, we observe high training accuracy but low test accuracy, it typically indicates a situation known as overfitting. It has low bias and high variance.
- And graph 03, we observe high/good training and test accuracy, it typically indicates a situation known as perfect/good fitting.



- There are some few examples that a model is good fit, underfit, overfit or not.



L10: Cross validation

Solve of Overfitting: **Cross-validation** and **Regularization** (Lasso-L1 Regularization, Ridge-L2 Regularization, and Elastic Net) techniques are common solutions/strategies of overfitting in machine learning models.