Sourav Kapil
Data Scientist
Email : souravkapil2000@gmail.com
Contact : 7060442742

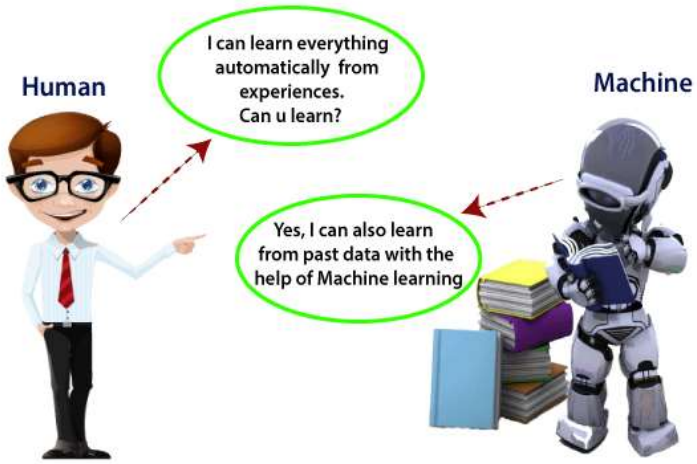
Introduction to Machine Learning

- What is Machine Learning?
- Difference Between AI, ML and DL
- Supervised / Unsupervised / Semi Supervised / Reinforcement
- Applications of Machine Learning
- Parametric vs NonParametric

What is Machine Learning?

- Machine learning is a way for computers to learn and make predictions or decisions without being explicitly programmed for each specific task. It's like teaching computers to think and make choices based on patterns they discover in data.

- In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of **Machine Learning.**



- Machine Learning is said to be a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own.

- Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

Key Components of Machine Learning:

- **Data:** At the heart of machine learning are data sets. These are collections of information that computers can learn from. Data can include text, numbers, images, videos, and more.

- **Algorithm:** An algorithm is a set of rules and instructions that guide the machine learning process. It's like a recipe for the computer to follow in order to learn from the data.

- **Model:** The model is the result of the machine learning process. It's like the brain of the computer that has learned from the data. This model can be used to make predictions or decisions about new, unseen data.

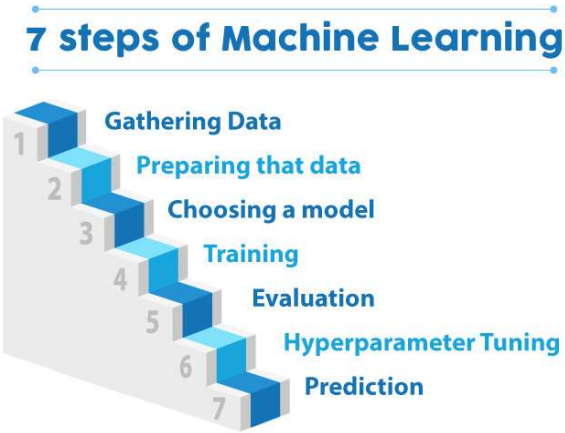
How Does Machine Learning Work?

- **Training:** To teach a machine learning model, you provide it with a lot of data that's already labeled or categorized. For example, if you want to teach a computer to recognize cats, you show it many pictures of cats and tell it that those are indeed cats.

- **Learning:** The algorithm processes the labeled data and tries to find patterns or relationships in the data. It adjusts its internal parameters to capture these patterns.

- **Testing:** Once the model has learned from the training data, you test it with new, unlabeled data to see how well it can make predictions or classifications. If it can correctly identify new cats it hasn't seen before, the model is successful.

- **Iteration:** If the model doesn't perform well, you refine the algorithm and provide more training data. This iterative process continues until the model becomes accurate enough for its intended purpose.



Artificial Intelligence vs Machine Learning vs Deep Learning

Let's break down the concepts of Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) -

---

Artificial Intelligence (AI):

Artificial Intelligence refers to the simulation of human intelligence processes by machines, especially computer systems. The goal of AI is to create machines that can mimic human thinking and decision-making. It's about making computers "smart" so that they can perform tasks that normally require human intelligence.

Examples of AI:

- **Chatbots:** These are AI programs that can have conversations with people, like virtual assistants on websites.
- **Game Playing AI:** AI that can play games like chess or Go at a very high level.
- **Autonomous Robots:** Robots that can navigate and perform tasks without human intervention.

Machine Learning (ML):

Machine Learning is a subset of AI that focuses on enabling machines to learn from data. Instead of being explicitly programmed for a specific task, ML algorithms can learn patterns and make decisions based on the data they've been provided. It's like teaching computers to learn and improve from experience.

Examples of Machine Learning:

- **Spam Filters:** Email programs use ML to learn what constitutes spam based on user behaviour.
- **Recommendation Systems:** Services like Netflix use ML to suggest movies based on your watching history.
- **Credit Scoring:** ML can analyze credit data to predict the likelihood of a person defaulting on a loan.

Deep Learning (DL):

Deep Learning is a subset of machine learning that focuses on using artificial neural networks to process and learn from data. These networks are inspired by the structure of the human brain and consist of interconnected layers of nodes that process information. Deep Learning has gained a lot of attention due to its ability to handle complex tasks, especially in areas like image and speech recognition.

Examples of Deep Learning:

- **Image Recognition:** DL algorithms can identify objects, people, or animals in images.
- **Speech Recognition:** Services like voice assistants use DL to understand and respond to spoken commands.
- **Language Translation:** DL models can translate text from one language to another.

Summary:

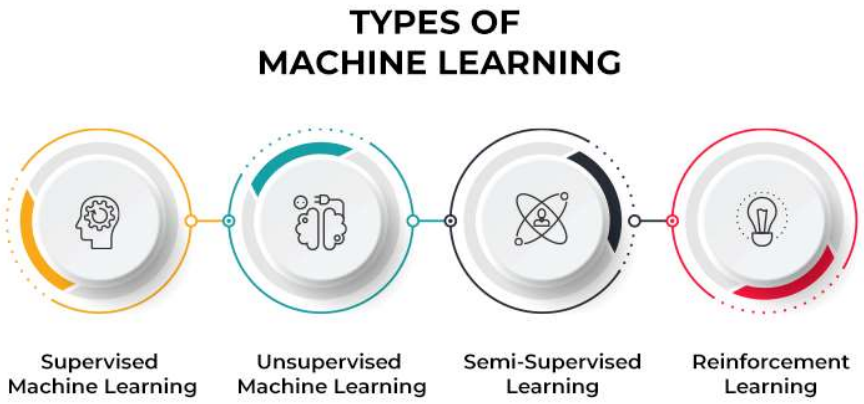
★ AI is the broader concept of making machines intelligent, allowing them to perform tasks that would typically require human intelligence.

★ ML is a subset of AI that involves teaching machines to learn from data and make predictions or decisions based on patterns.

★ DL is a subset of ML that uses neural networks to process and learn from complex data, particularly suited for tasks like image and speech recognition.

Types of Machine Learning:

Let’s discuss 4 different types of machine learning -

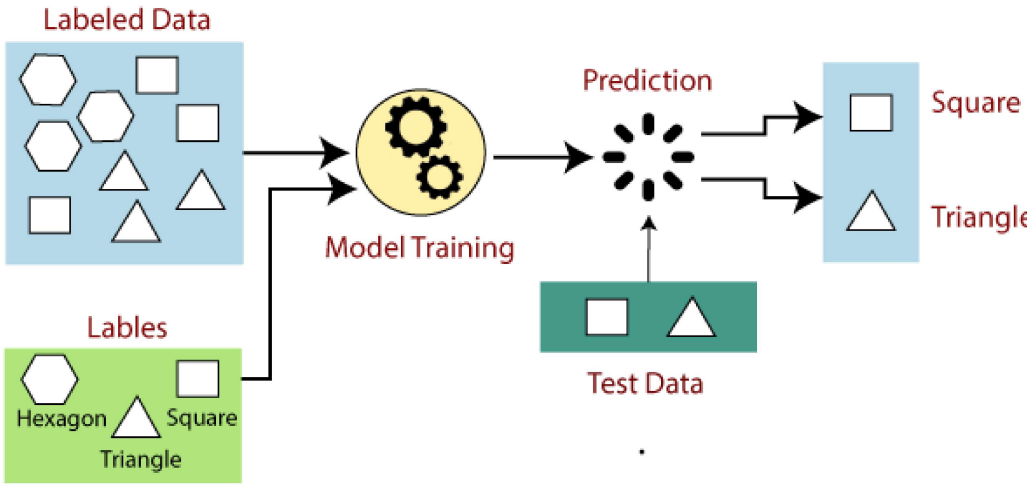


★ Supervised Machine Learning

- Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

- In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

- Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

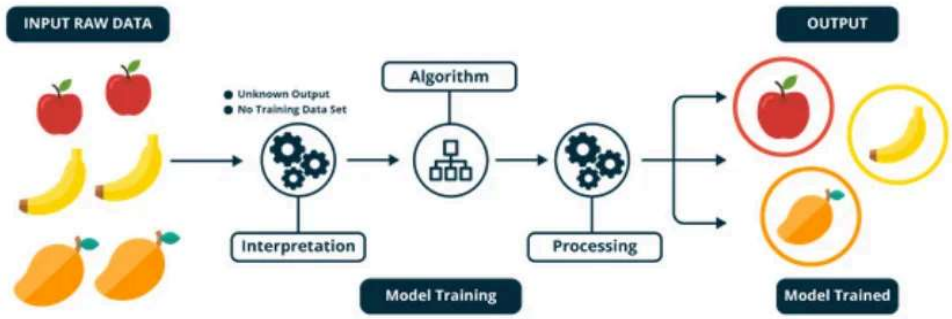


★ Unsupervised Machine Learning

- As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.
  It can be defined as:

  "Unsupervised learning is a type of machine learning in which models are trained using an unlabeled dataset and are allowed to act on that data without any supervision."

- Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of the dataset, group that data according to similarities, and represent that dataset in a compressed format.

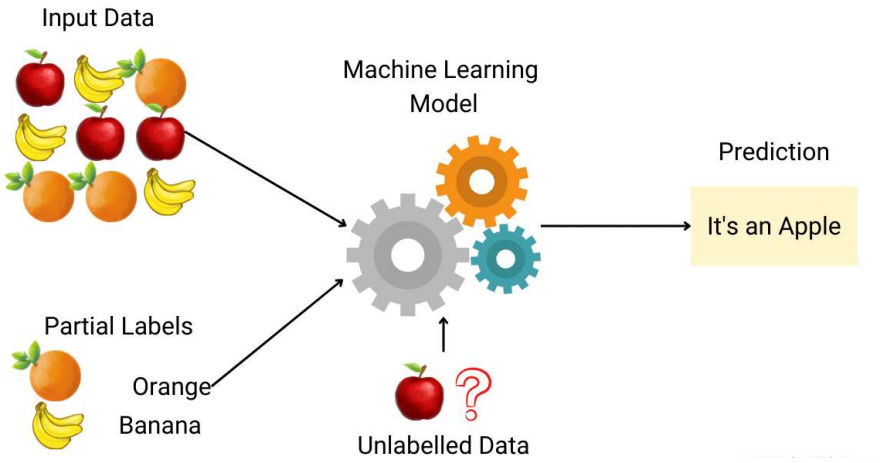


★ Semi-Supervised Learning

- Semi-supervised learning is a combination of supervised and unsupervised learning. It involves using a small amount of labeled data along with a larger amount of unlabeled data.

- The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data.

- Semi-supervised learning is particularly useful when there is a large amount of unlabeled data available, but it's too expensive or difficult to label all of it.

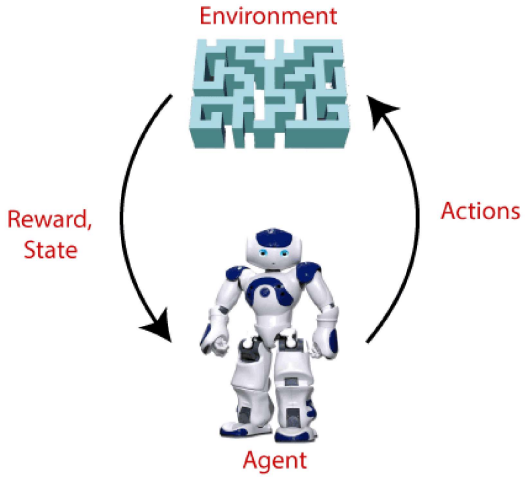


★ Reinforcement Learning

- Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.

- In Reinforcement Learning, the agent learns automatically using feedback without any labeled data, unlike supervised learning. Since there is no labeled data, the agent is bound to learn by its experience only. RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as **game-playing, robotics, etc.**

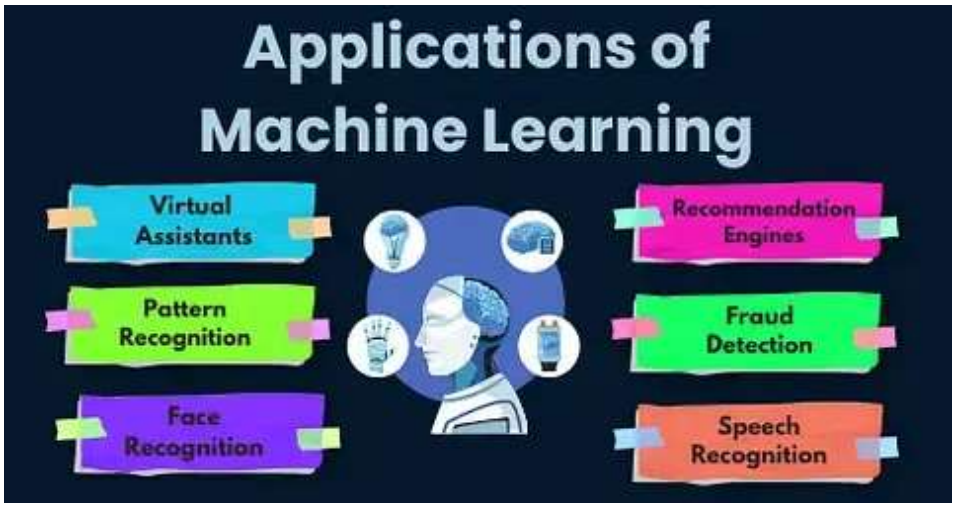
- The agent learns with the process of hit and trial, and based on the experience, it learns to perform the task in a better way. Hence, we can say that "Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that."



Applications of Machine Learning

Real-World Examples :-

- **Image Recognition:** Models can be trained to recognize objects in images, like telling apart cats from dogs.
- **Recommendation Systems:** Think of platforms suggesting movies, songs, or products based on your previous choices - that's machine learning at work.
- **Language Translation:** Machine learning helps translate languages by learning the patterns and nuances of language use.
- **Self-Driving Cars:** These use machine learning to analyze the environment, predict the behaviour of other vehicles, and make driving decisions.
- **Sentiment Analysis:** ML determines the sentiment behind social media posts and reviews.



In essence, machine learning is like teaching computers to learn from data and make informed decisions. It's a powerful tool with applications in various fields, from healthcare to entertainment to transportation.

Parametric vs NonParametric

Parametric Methods:

Parametric methods assume a specific functional form for the relationship between input variables (features) and the output variable (target). The model has a fixed number of parameters, and the goal is to estimate these parameters from the training data.

Characteristics:

- **Assumption:** Parametric methods make assumptions about the underlying distribution of the data.
- **Simplicity:** These methods are often simpler and require fewer training data points.
- **Faster Training:** Because the model structure is predefined, training is usually faster.
- **Risk:** If the assumed function doesn't match the true relationship in the data, the model's performance might be limited.

Example: Linear Regression is a classic parametric method. It assumes a linear relationship between input features and the target variable. The model tries to find the best-fit line that minimizes the error between predicted and actual values.

Imagine you have a dataset of houses with their respective sizes and prices. Linear regression assumes that the relationship between house size (input) and price (output) can be modelled as a straight line (a linear function). The model's goal is to find the slope and intercept of that line that best fits the data points.

How It Works:

The linear regression model will learn the equation of the line that best represents the data:

Y = mx + c
Price = Slope * Size + Intercept

Once the slope and intercept are learned from the training data, the model can predict the price of a new house based on its size.

Pros:

- ➢ **Simplicity:** Easy to understand and interpret.
- ➢ **Efficiency:** Faster training and prediction.

Cons:

Assumption: If the relationship is not linear, the model might not perform well.

Non-Parametric Methods:

Non-parametric methods do not assume a fixed functional form for the relationship between input and output variables. Instead, they rely on flexible models that adapt to the data patterns.

Characteristics:

- **Flexibility:** Non-parametric methods can capture complex relationships without being tied to specific assumptions.
- **No Distribution Assumptions:** They don't require assumptions about the underlying data distribution.
- **More Data:** As the model is more flexible, it might require more training data to generalize well.
- **Slower Training:** The flexibility can lead to longer training times, especially for large datasets.

Example: k-Nearest Neighbors (k-NN) is a non-parametric method. It classifies a new data point based on the majority class of its k-nearest neighbors in the training data. The decision boundary is determined by the distribution of the training data, without assuming a fixed function.

Suppose you have a dataset of flowers with their petal lengths and widths, and you want to classify them into different species. k-NN doesn't assume a specific function. Instead, it looks at the k nearest neighbors to a new flower and assigns the species based on the majority class among those neighbors.

How It Works:

If you have a new flower and you set k = 3, the model will find the three nearest flowers in the training data. If two of them belong to species A and one to species B, the new flower will be classified as species A.

Pros:

- ➢ **Flexibility:** Can capture complex relationships and patterns.
- ➢ **No Assumption:** Doesn't require assumptions about data distribution.

Cons:

- ➢ **Complexity:** Slower training and prediction.
- ➢ **More Data:** Might require more data to generalize well.

Summary:

In essence, parametric methods make specific assumptions about the functional form of the relationship between variables. They are simpler and faster, but might not capture complex patterns if the assumptions are incorrect. Non-parametric methods, on the other hand, are more flexible and can capture complex relationships without making strong assumptions, but they might require more data and training time.

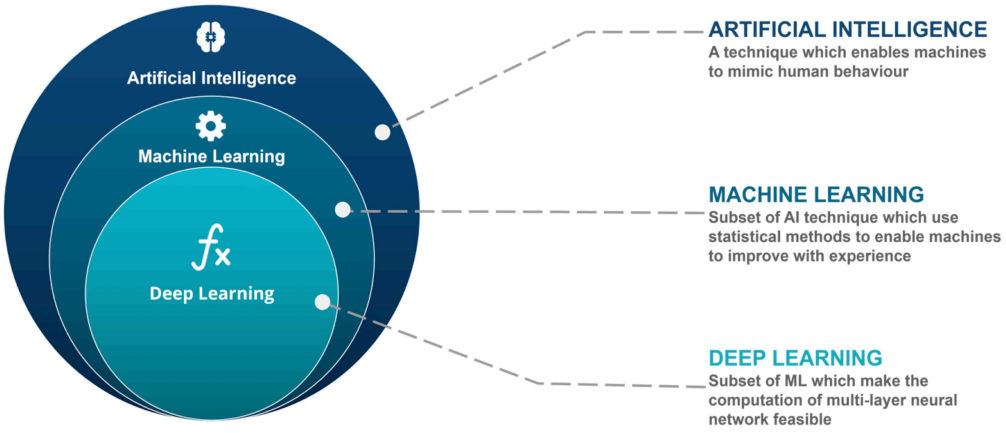
ML Basic Concepts

Machine Learning vs Deep Learning

★ Machine learning and deep learning are both types of AI. In short, machine learning is AI that can automatically adapt with minimal human interference.

★ Machine Learning allows the computers to learn from the experiences on its own, use statistical methods to improve the performance and predict the output without being explicitly programmed.

★ Deep learning is a subset of machine learning that uses artificial neural networks to mimic the learning process of the human brain.



Machine learning

- A subset of AI
- Can train on smaller data sets
- Requires more human intervention to correct and learn
- Shorter training and lower accuracy
- Makes simple, linear correlations
- Can train on a CPU (central processing unit)

Deep learning

- A subset of machine learning
- Requires large amounts of data
- Learns on its own from environment and past mistakes
- Longer training and higher accuracy
- Makes non-linear, complex correlations
- Needs a specialised GPU (graphics processing unit) to train

Supervised vs Unsupervised Learning

Supervised Learning	Unsupervised Learning
• Supervised learning algorithms are trained using labelled data.	Unsupervised learning algorithms are trained using unlabeled data.
• Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
• Supervised learning model predicts the output.	Unsupervised learning models find the hidden patterns in data.
• In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
• The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
• Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
• Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.
• Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.

Bias-Variance Tradeoff

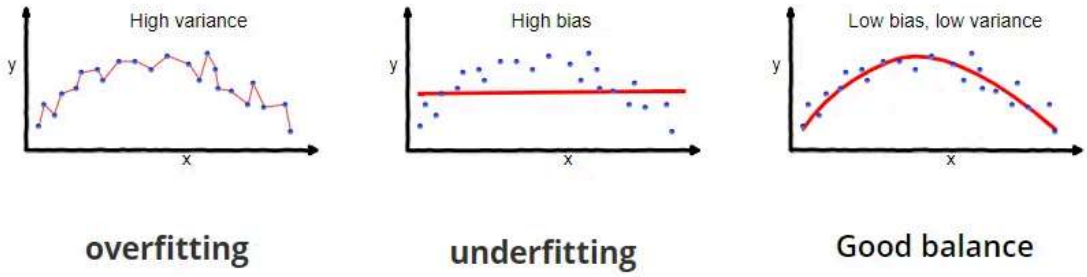
Whenever we discuss model prediction, it's important to understand prediction errors (bias and variance). There is a tradeoff between a model's ability to minimize bias and variance. Gaining a proper understanding of these errors would help us not only to build accurate models but also to avoid the mistake of overfitting and underfitting.

What is bias?

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

What is variance?

Variance is the variability of model prediction for a given data point or a value which tells us the spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but have high error rates on test data.

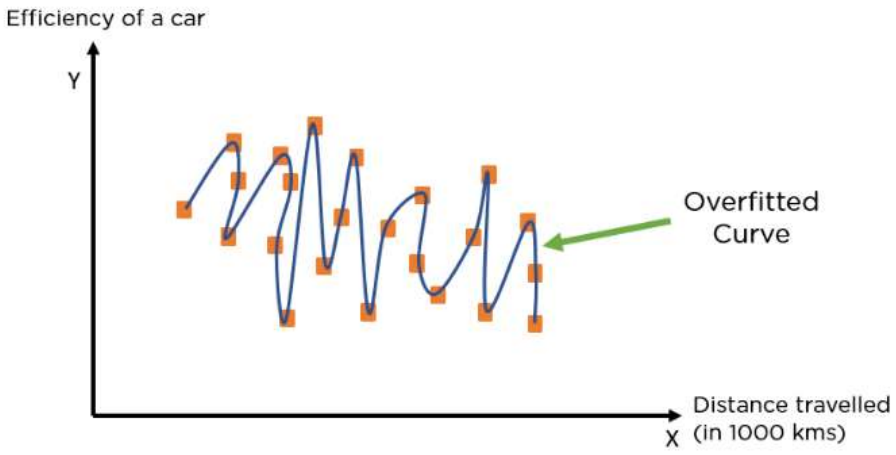


To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.

Overfitting and Underfitting

Overfitting

When a model performs very well for training data but has poor performance with test data (new data), it is known as overfitting. In this case, the machine learning model learns the details and noise in the training data such that it negatively affects the performance of the model on test data. Overfitting can happen due to **low bias and high variance.**



Reasons for Overfitting

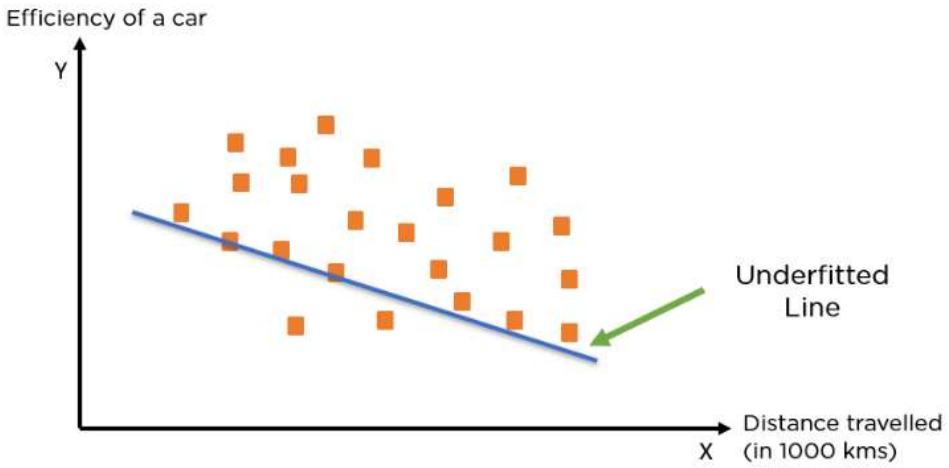
- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high variance
- The size of the training dataset used is not enough
- The model is too complex.

Ways to Tackle Overfitting

- Using K-fold cross-validation
- Using Regularization techniques such as Lasso and Ridge
- Training model with sufficient data
- Adopting ensembling techniques

Underfitting

When a model has not learned the patterns in the training data well and is unable to generalize well on the new data, it is known as underfitting. An underfit model has poor performance on the training data and will result in unreliable predictions. Underfitting occurs due to **high bias and low variance.**



Reasons for Underfitting

- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high bias
- The size of the training dataset used is not enough
- The model is too simple

Ways to Tackle Underfitting

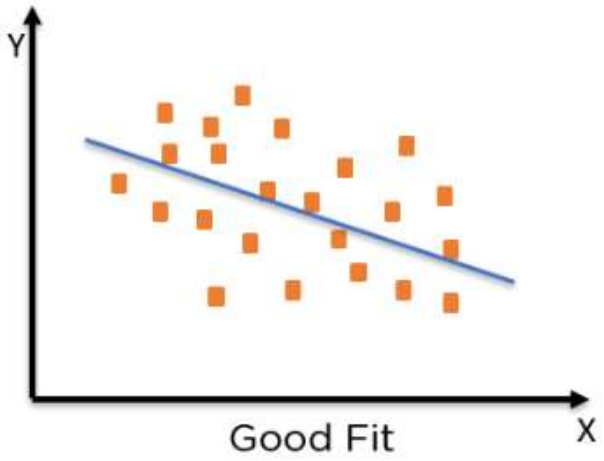
- Increase the number of features in the dataset
- Increase model complexity
- Reduce noise in the data
- Increase the duration of training the data

Now that you have understood what **overfitting and underfitting** are, let's see what is a good fit model in this tutorial on overfitting and underfitting in machine learning.

Goodness of Fit

As far as a machine learning algorithm is concerned, a good fit is when both the training data error and the test data are minimal. As the algorithm learns, the mistake in the training data for the model is decreasing over time, and so is the error on the test dataset.

In order to achieve a good fit, you need to stop training at a point where the error starts to increase.



Imbalance Data

- A classification data set with skewed class proportions is called imbalanced.
- Data imbalance usually reflects an unequal distribution of classes within a dataset.
- Tree based algorithms perform well with imbalance datasets.
- Boosting algorithms (e.g AdaBoost, XGBoost,…) are ideal for imbalanced datasets because higher weight is given to the minority class at each successive iteration.

For example, in a credit card fraud detection dataset, most of the credit card transactions are not fraud and a very few classes are fraud transactions.

Handling Imbalanced Data

- The main two methods that are used to tackle the class imbalance are -
    - ★ **Upsampling/Oversampling**
    - ★ **Downsampling/Undersampling**
- The sampling process is applied only to the training set and no changes are made to the validation and testing data.

Upsampling

Upsampling is a procedure where synthetically generated data points (corresponding to minority class) are injected into the dataset. After this process, the counts of both labels are almost the same.

Techniques to perform Upsampling -

- SMOTE
- DataDuplication

SMOTE (Synthetic Minority Oversampling Technique)

- ➢ SMOTE is basically used to create synthetic class samples of minority classes to balance the distribution.
- ➢ It works based on the KNearestNeighbours algorithm, synthetically generating data points that fall in the proximity of the already existing outnumbered group.
- ➢ The input records should not contain any null values when applying this approach.

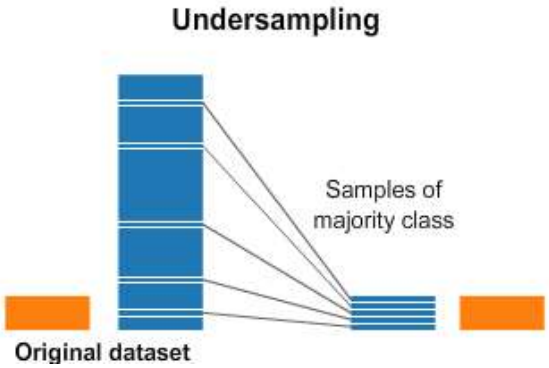
- **SMOTENC:** SMOTE variant for continuous and categorical features.
- **SMOTEN:** SMOTE variant for data with only categorical features.

Data Duplication

- ➢ In this approach, the existing data points corresponding to the outvoted labels are randomly selected and duplicated.

Downsampling

Downsampling is a mechanism that reduces the count of training samples falling under the majority class. As it helps to even up the counts of target categories. By removing the collected data, we tend to lose so much valuable information.



Balanced Class Weight:

- ➢ The undersampling technique removes the majority class data points which results in data loss, whereas upsampling creates artificial data points of the minority class.
- ➢ During the training of machine learning, one can use **class_weight** parameters to handle the imbalance in the dataset.
- ➢ Scikit-learn comes with the **class_weight** parameters for all the machine learning algorithms.

- **class weight — balanced:** The class weight is inversely proportional to class frequencies in the input data.

- **{class_label: weight}:** Let's say, target class labels are 0 and 1. Passing input as class_weight={0:3, 1:1} means class 0 has weight 3 and class 1 has weight 1.

-

Evaluation Metrics for ML Models

(Loss Functions)

Why Do We Require Evaluation Metrics?

Most beginners and practitioners most of the time do not bother about the model performance. The talk is about building a well-generalized model, Machine learning model cannot have 100 percent efficiency otherwise the model is known as a biased model. which further includes the concept of overfitting and underfitting.

- ➢ Accuracy (e.g. classification accuracy) is a measure for classification, not regression.
- ➢ We cannot calculate accuracy for a regression model.
- ➢ The skill or performance of a regression model must be reported as an error in those predictions.

Metrics for Regression

- ➢ In this section, we will take a closer look at the popular metrics for regression models and how to calculate them for your predictive modeling project.

- ➢ There are three error metrics that are commonly used for evaluating and reporting the performance of a regression model; they are:

    - Mean Squared Error (MSE).
    - Root Mean Squared Error (RMSE).
    - Mean Absolute Error (MAE)
    - R2 squared
    - Adjusted R2 Squares

Mean Squared Error

- ➢ Mean Squared Error, or MSE for short, is a popular error metric for regression problems.
- ➢ It is also an important loss function for algorithms fit or optimized using the least squares framing of a regression problem. Here “least squares” refers to minimizing the mean squared error between predictions and real values.
- ➢ The MSE is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

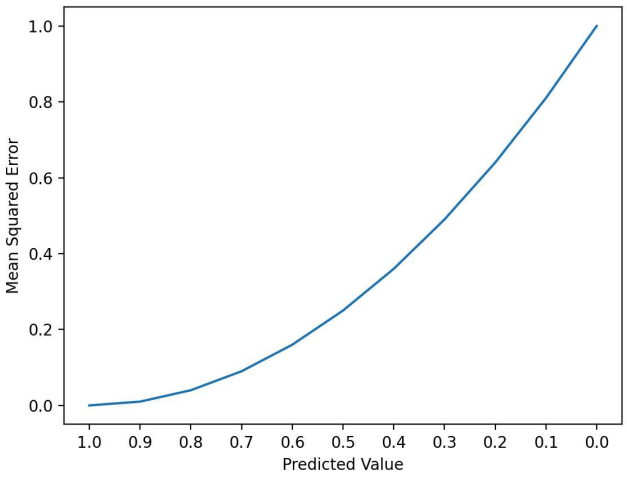
Advantages of MSE

The graph of MSE is differentiable, so you can easily use it as a loss function.

Disadvantages of MSE

The value you get after calculating MSE is a squared unit of output. For example, the output variable is in meter(m) then after calculating MSE the output we get is in meter squared.

If you have outliers in the dataset then it penalizes the outliers most and the calculated MSE is bigger. So, in short, It is not Robust to outliers which were an advantage in MAE.



Root Mean Squared Error

- The Root Mean Squared Error, or RMSE, is an extension of the mean squared error.
- Importantly, the square root of the error is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted.
- For example, if your target variable has the units “dollars,” then the RMSE error score will also have the unit “dollars” and not “squared dollars” like the MSE.
- As such, it may be common to use MSE loss to train a regression predictive model, and to use RMSE to evaluate and report its performance

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

Advantages of RMSE

The output value you get is in the same unit as the required output variable which makes interpretation of loss easy.

Disadvantages of RMSE

It is not that robust to outliers as compared to MAE.
For performing RMSE we have to use the NumPy square root function over MSE.

Mean Absolute Error

- Mean Absolute Error, or MAE, is a popular metric because, like RMSE, the units of the error score match the units of the target value that is being predicted.
- Unlike the RMSE, the changes in MAE are linear and therefore intuitive.
- As its name suggests, the MAE score is calculated as the average of the absolute error values. Absolute or abs() is a mathematical function that simply makes a number positive.
- Therefore, the difference between an actual and predicted value may be positive or negative and is forced to be positive when calculating the MAE.

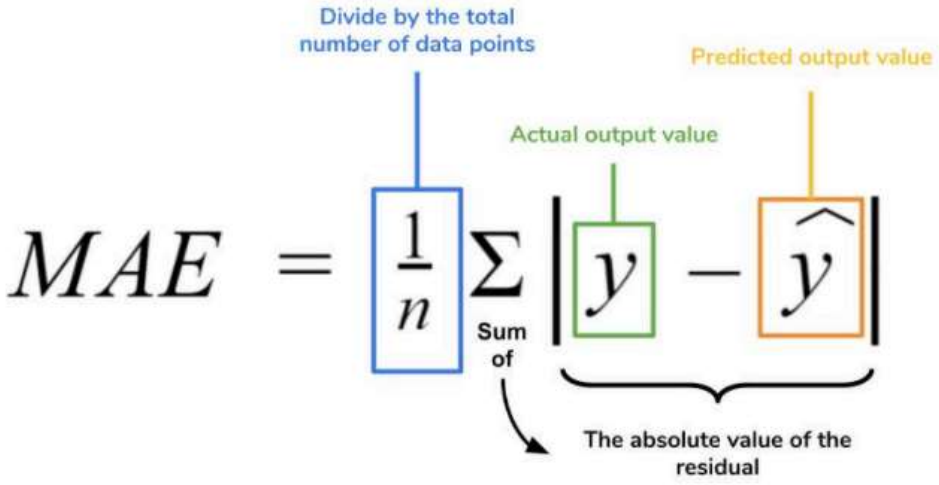
Advantages of MAE

The MAE you get is in the same unit as the output variable.
It is most Robust to outliers.

.

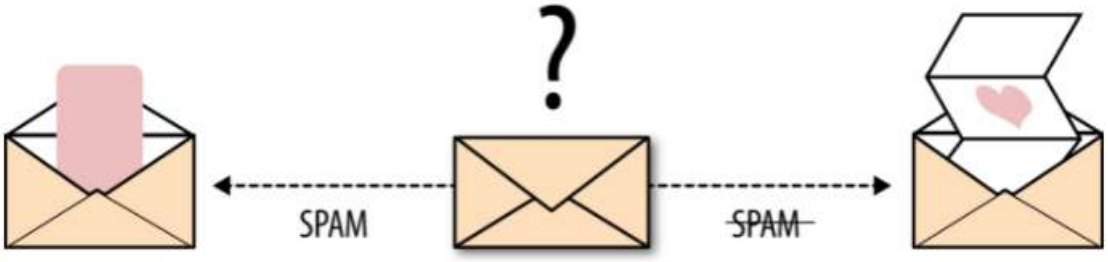
Disadvantages of MAE

The graph of MAE is not differentiable so we have to apply various optimizers like Gradient descent which can be differentiable.



Metrics for Classification

Classification is about predicting the class labels given in input data. A very common example of binary classification is spam detection, where the input data could include the email text and metadata (sender, sending time), and the output label is either “spam” or “not spam.” Sometimes, people use some other names also for the two classes: “positive” and “negative,” or “class 1” and “class 0.”



There are many ways for measuring classification performance. Accuracy, confusion matrix, log-loss, and AUC-ROC are some of the most popular metrics. Precision-recall is a widely used metric for classification problems.

Confusion Matrix

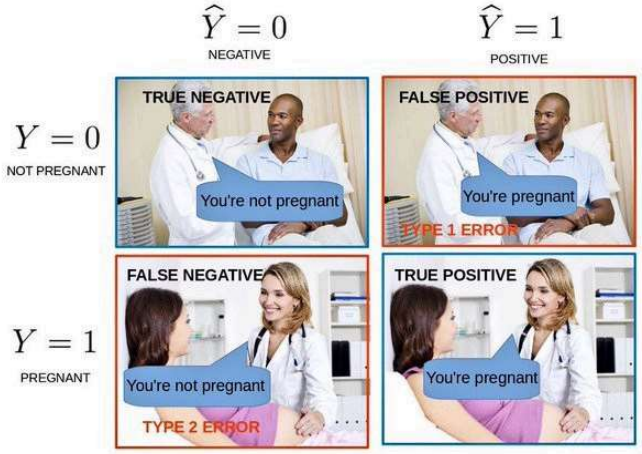
Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values.

“A confusion matrix is defined as the table that is often used to describe the performance of a classification model on a set of the test data for which the true values are known.”

		Predicted	
		Negative (**N**) -	Positive (**P**) +
Actual	Negative -	True Negatives (TN)	False Positives (FP) **Type I error**
	Positive +	False Negatives (FN) **Type II error**	True Positives (TP)

- It is extremely useful for measuring the **Recall, Precision, Accuracy, and AUC-ROC curves.**

- Let’s try to understand **TP, FP, FN, TN** with an example of pregnancy analogy.



- **True Positive:** We predicted positive and it's true. In the image, we predicted that a woman is pregnant and she actually is.

- **True Negative:** We predicted negative and it's true. In the image, we predicted that a man is not pregnant and he actually is not.

- **False Positive (Type 1 Error):** We predicted positive and it's false. In the image, we predicted that a man is pregnant but he actually is not.

- **False Negative (Type 2 Error):** We predicted negative and it's false. In the image, we predicted that a woman is not pregnant but she actually is.

Accuracy

Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions.

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

When any model gives an accuracy rate of 99%, you might think that model is performing very well but this is not always true and can be misleading in some situations.

We discussed Accuracy, now let's discuss some other metrics of the confusion matrix

Precision

- Precision explains how many of the correctly predicted cases actually turned out to be positive.
- Precision is useful in the cases where False Positive is a higher concern than False Negatives.
- The importance of Precision is in music or video recommendation systems, e-commerce websites, etc. where wrong results could lead to customer churn and this could be harmful to the business.
- Precision for a label is defined as the number of true positives divided by the number of predicted positives.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Recall (Sensitivity)

- ➢ Recall explains how many of the actual positive cases we were not able to predict correctly with our model.
- ➢ It is a useful metric in cases where False Negative is of higher concern than False Positive.
- ➢ It is important in medical cases where it doesn't matter whether we raise a false alarm but the actual positive cases should not go undetected!
- ➢ Recall for a label is defined as the number of true positives divided by the total number of actual positives.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

F1 Score

- ➢ It gives a combined idea about Precision and Recall metrics. It is maximum when Precision is equal to Recall.
- ➢ F1 Score is the harmonic mean of precision and recall.
- ➢ The F1 score punishes extreme values more. F1 Score could be an effective evaluation metric in the following cases:

    - When FP and FN are equally costly.
    - Adding more data doesn't effectively change the outcome
    - True Negative is high

$$F1 = 2.\frac{Precision \times Recall}{Precision + Recall}$$

AUC - ROC Curve

- AUC - (Area Under Curve)
- ROC - (Receiver Operating Characteristics)

The AUC ROC curve is a graph which shows the performance of a classification model at all thresholds. ROC is a probability curve and AUC represents degree of separability. ROC plots the following parameters:-

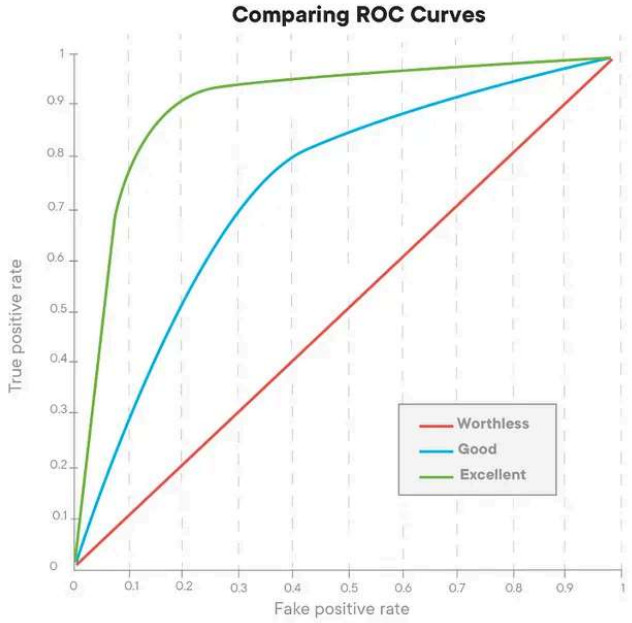
- **True Positive Rate (TPR),** also known as recall or sensitivity, which was explained in the previous section.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

- **False Positive Rate (FPR),** also known as Fall-out, the ratio of the false positive predictions compared to all values that are actually negative.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

★ Both the TPR and FPR are within the range [0, 1]. The curve is the FPR vs TPR at different points in the range [0, 1].
★ The best performing classification models will have a curve similar to the green line in the graph below.
★ The green line has the largest Area Under the Curve. The higher the AUC, the better your model is performing.
★ A classifier with only 50–50 accuracy is realistically no better than randomly guessing, which makes the model worthless (red line).



Log Loss (Binary CrossEntropy Loss)

Log loss (Logistic loss) or Cross-Entropy Loss is one of the major metrics to assess the performance of a classification problem.

For a single sample with true label y∈{0,1} and a probability estimate p=Pr(y=1), the log loss is:

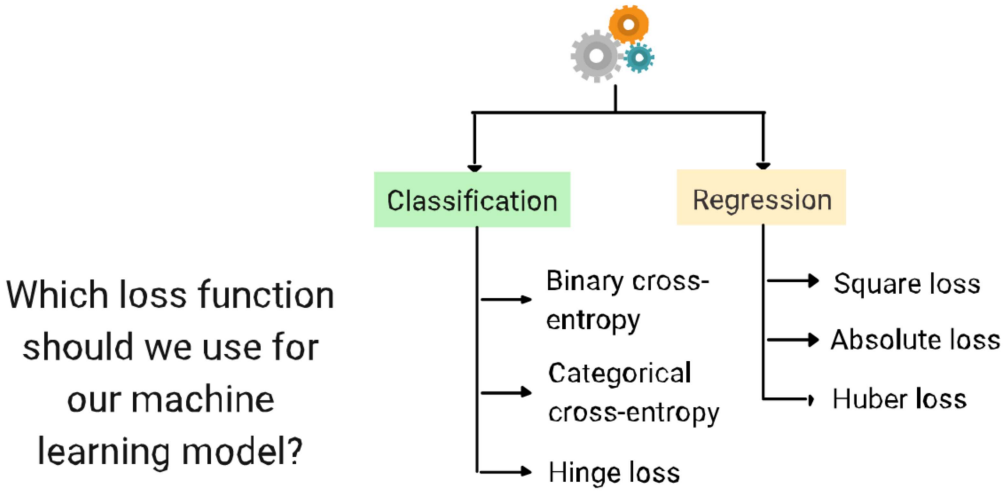
$$logloss_{(\mathrm{N}=1)} = y\log(p) + (1-y)\log(1-p)$$

Loss Function and Cost Function

In the context of machine learning and optimization, both loss and cost functions are used to quantify how well a model performs in terms of its predictions compared to the actual or desired outcomes.

Difference between Loss and Cost function

In other words, the loss function is to capture the difference between the actual and predicted values for a single record whereas cost functions aggregate the difference for the entire training dataset. The Most commonly used loss functions are Mean-squared error and Hinge loss.



Gradient Descent: Minimizing the cost function

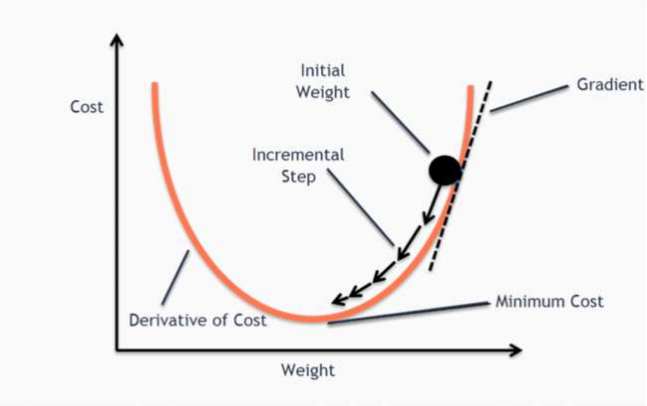
- As we discussed in the above section, the cost function tells how wrong your model is? And each machine learning model tries to minimize the cost function in order to give the best results. Here comes the role of Gradient descent.

- Gradient Descent is an optimization algorithm which is used for optimizing the cost function or error in the model.

- It enables the models to take the gradient or direction to reduce the errors by reaching the least possible error. Here direction refers to how model parameters should be corrected to further reduce the cost function.

- The error in your model can be different at different points, and you have to find the quickest way to minimize it, to prevent resource wastage.

- Gradient descent is an iterative process where the model gradually converges towards a minimum value, and if the model iterates further than this point, it produces little or zero changes in the loss. This point is known as convergence, and at this point, the error is least, and the cost function is optimized.



- Below is the equation for gradient descent in linear regression:

$$X = X - lr * \frac{d}{dX} f(X)$$

Where,

X = *input*

$F(X)$ = *output based on X*

lr = *learning rate*

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

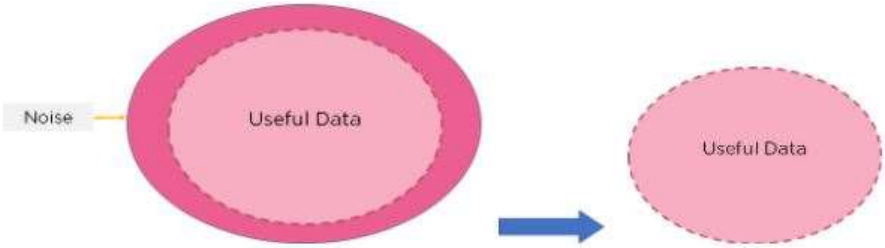
- In the gradient descent equation, alpha is known as the learning rate. This parameter decides how fast you should move down to the slope. For large alpha, take big steps, and for small alpha value, you need to take small steps.

Feature Selection

- Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data.

- It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve. We do this by including or excluding important features without changing them.

- It helps in cutting down the noise in our data and reducing the size of our input data.

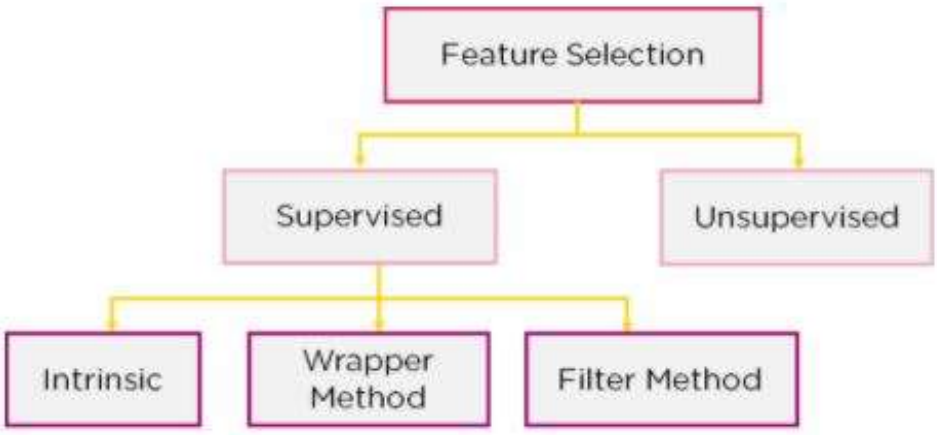


Feature Selection Models

Feature selection models are of two types:

1. **Supervised Models:** Supervised feature selection refers to the method which uses the output label class for feature selection. They use the target variables to identify the variables which can increase the efficiency of the model.

2. **Unsupervised Models:** Unsupervised feature selection refers to the method which does not need the output label class for feature selection. We use them for unlabelled data.

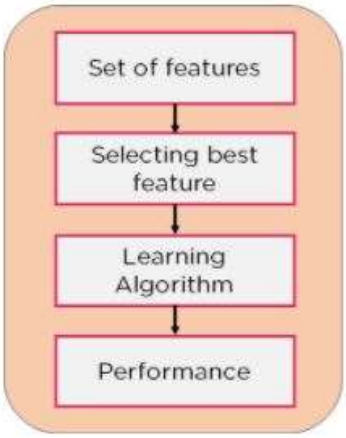


We can further divide the supervised models into three :

- ★ **Filter Method**
- ★ **Wrapper Method**
- ★ **Intrinsic Method**

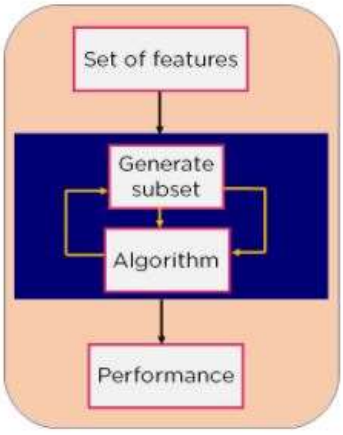
1. Filter Method:

In this method, features are dropped based on their relation to the output, or how they are correlating to the output. We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly.
Eg: Information Gain, Chi-Square Test, Correlation Matrix etc.



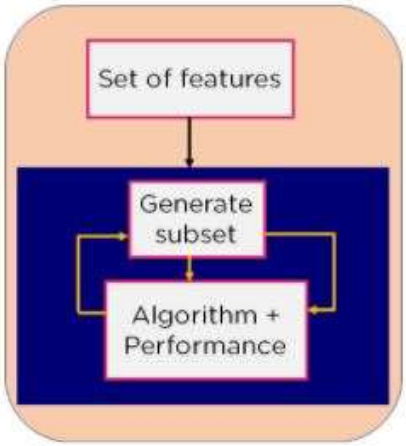
2. Wrapper Method:

We split our data into subsets and train a model using this. Based on the output of the model, we add and subtract features and train the model again. It forms the subsets using a greedy approach and evaluates the accuracy of all the possible combinations of features.
Eg: Forward Selection, Backwards Elimination, etc.



3. Intrinsic Method:

This method combines the qualities of both the Filter and Wrapper method to create the best subset. This method takes care of the machine training iterative process while maintaining the computation cost to be minimum.
Eg: Lasso and Ridge Regression.

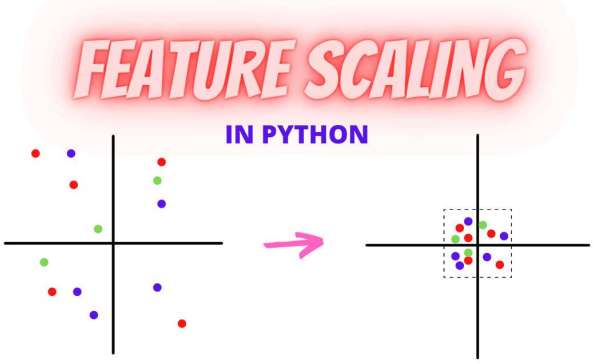


Feature Scaling

- Feature scaling in machine learning is one of the most critical steps during the pre-processing of data before creating a machine learning model. Scaling can make a difference between a weak machine learning model and a better one.

- Feature scaling is the process of normalising the range of features in a dataset.

- Real-world datasets often contain features that are varying in degrees of magnitude, range and units. Therefore, in order for machine learning models to interpret these features on the same scale, we need to perform feature scaling.



- The most common techniques of feature scaling are **Normalization** and **Standardization -**

    - **Normalisation,** also known as **min-max scaling**, is a scaling technique whereby the values in a column are shifted so that they are bounded between a fixed range of **0 and 1.**

    - **MinMaxScaler** is the Scikit-learn function for normalisation. Normalization is good to use when the distribution of data does not follow a Gaussian distribution. It can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbors. In Neural Networks algorithms that require data on a 0–1 scale, normalization is an essential pre-processing step.

    - Normalization is used when we want to bound our values between two numbers, typically, between [0,1] or [-1,1].

Standardisation

- On the other hand, **Standardisation** or **Z-score** normalisation is another scaling technique whereby the values in a column are rescaled so that they demonstrate the properties of a standard Gaussian distribution, that is mean = 0 and Std = 1.

- **StandardScaler** is the Scikit-learn function for standardisation.

- Standardization is useful when your data has varying scales and the algorithm you are using does make assumptions about your data having a Gaussian distribution, such as linear regression, logistic regression, and linear discriminant analysis.

Feature Engineering

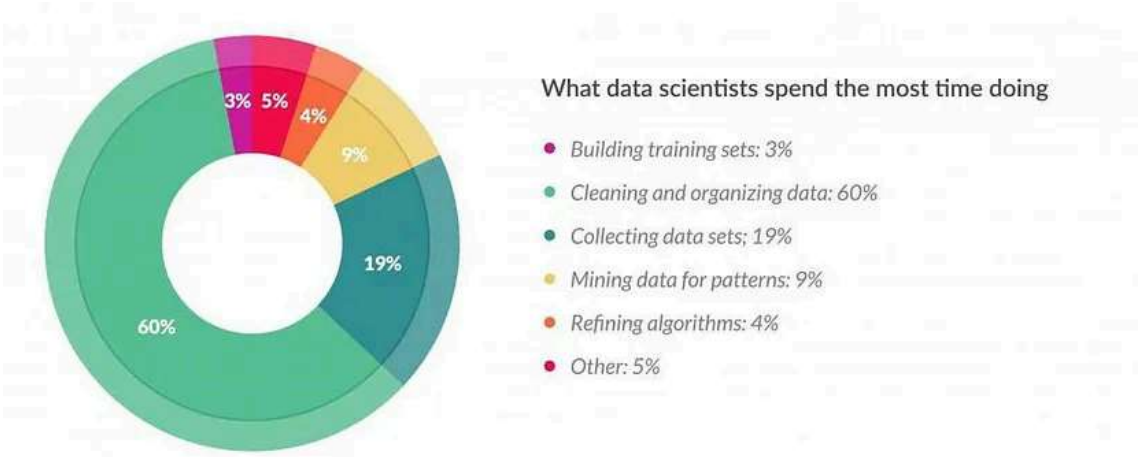
- Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work.

- Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive. The need for manual feature engineering can be obviated by automated feature learning.

- Feature engineering is an informal topic, but it is considered essential in applied machine learning.

Why is Feature Engineering important?

➢ If feature engineering is done correctly, it increases the predictive power of machine learning algorithms by creating features from raw data that help facilitate the machine learning process.



➢ Data scientists spend 80% of their time on Data Preparation:

Encoding Techniques

In many practical data science activities, the data set will contain categorical variables. These variables are typically stored as text values. Since machine learning is based on mathematical equations, it would cause a problem when we keep categorical variables as is.

There are 3 types of encoding techniques -

Nominal Encoding

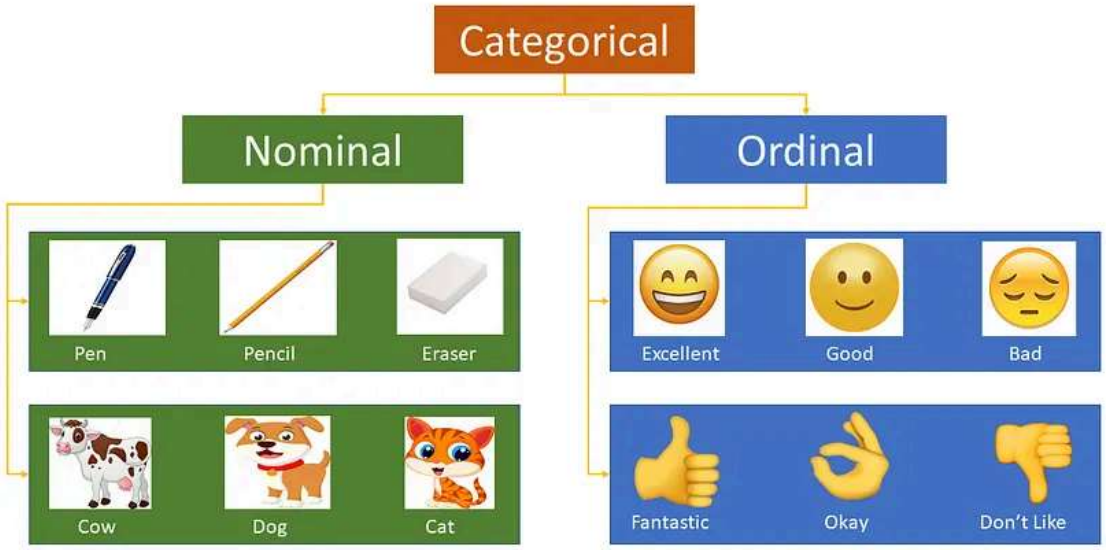
- One Hot Encoding
- Mean Encoding or Target Encoding

Ordinal Encoding

- Label Encoding
- Target Guided Ordinal Encoding

Binary Encoding

- Binary Encoding



Nominal Encoding

One-Hot Encoding

- We use this categorical data encoding technique when the features are nominal**(do not have any order).** In one hot encoding, for each level of a categorical feature, we create a new variable.

- Each category is mapped with a binary variable containing either 0 or 1. Here, 0 represents the absence, and 1 represents the presence of that category.

- These newly created binary features are known as Dummy variables. The number of dummy variables depends on the levels present in the categorical variable.

- **Example -** Suppose we have a dataset with a category animal, having different animals like Dog, Cat, Sheep, Cow, Lion. Now we have to one-hot encode this data.

Index	Animal
0	Dog
1	Cat
2	Sheep
3	Horse
4	Lion

One-Hot code

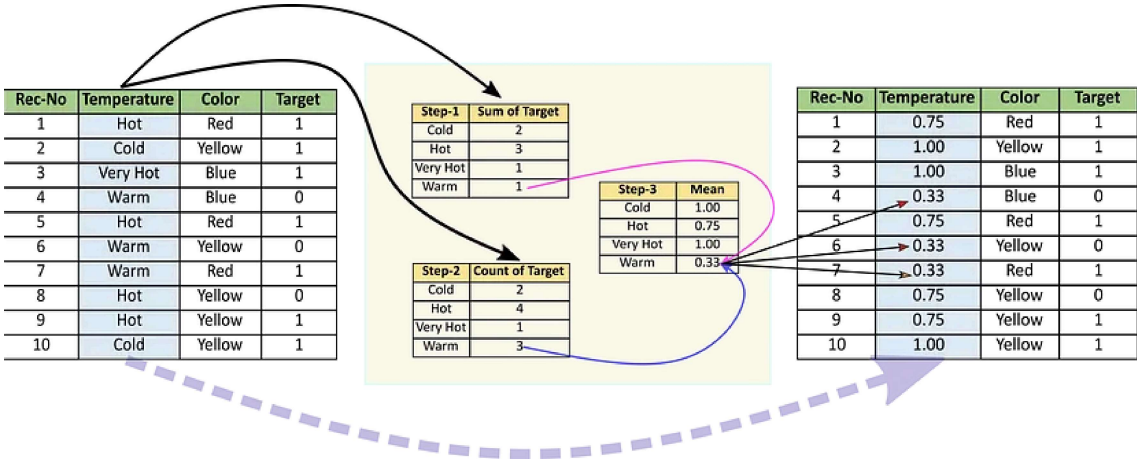
Index	Dog	Cat	Sheep	Lion	Horse
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

Mean Encoding OR Target Encoding

- Mean Encoding or Target Encoding is one very popular encoding approach followed by Kagglers. Mean encoding is similar to label encoding, except here labels are correlated directly with the target.

- For example, mean target encoding for each category in the feature label is decided with the mean value of the target variable on training data.

- The advantages of the mean target encoding are that it does not affect the volume of the data and helps in faster learning.



Rec-No	Temperature	Color	Target
1	Hot	Red	1
2	Cold	Yellow	1
3	Very Hot	Blue	1
4	Warm	Blue	0
5	Hot	Red	1
6	Warm	Yellow	0
7	Warm	Red	1
8	Hot	Yellow	0
9	Hot	Yellow	1
10	Cold	Yellow	1

Step-1	Sum of Target
Cold	2
Hot	3
Very Hot	1
Warm	1

Step-2	Count of Target
Cold	2
Hot	4
Very Hot	1
Warm	3

Step-3	Mean
Cold	1.00
Hot	0.75
Very Hot	1.00
Warm	0.33

Rec-No	Temperature	Color	Target
1	0.75	Red	1
2	1.00	Yellow	1
3	1.00	Blue	1
4	0.33	Blue	0
5	0.75	Red	1
6	0.33	Yellow	0
7	0.33	Red	1
8	0.75	Yellow	0
9	0.75	Yellow	1
10	1.00	Yellow	1

Ordinal Encoding

Label Encoding

- In this encoding each category is assigned a value from 0 through N (here N is the number of categories for the feature). It may look like (Car<Bus<Truck ….0<1 < 2).

- Categories that have some ties or are close to each other lose some information after encoding.

CAT73	CAT73 label_encoded
A	1
A	1
C	3
B	2
A	1
C	3
B	2

Binary Encoding

- Binary encoding is a combination of Hash encoding and one-hot encoding. In this encoding scheme, the categorical feature is first converted into numerical using an ordinal encoder.

- Then the numbers are transformed into binary numbers. After that, the binary value is split into different columns.

- Binary encoding works really well when there are a high number of categories. For example the cities in a country where a company supplies its products.

	City
0	Delhi
1	Mumbai
2	Hyderabad
3	Chennai
4	Bangalore
5	Delhi
6	Hyderabad
7	Mumbai
8	Agra

	City_0	City_1	City_2	City_3
0	0	0	0	1
1	0	0	1	0
2	0	0	1	1
3	0	1	0	0
4	0	1	0	1
5	0	0	0	1
6	0	0	1	1
7	0	0	1	0
8	0	1	1	0

Model Selection and Model Assessment

- Model selection is the process of selecting one final machine learning model from among a collection of candidate machine learning models for a training dataset.

- Model selection is a process that can be applied both across different types of models (e.g. logistic regression, SVM, KNN, etc.)

- Model selection is the process of choosing one of the models as the final model that addresses the problem. Model selection is different from model assessment.

➢ The process of evaluating a model's performance is known as model assessment, whereas the process of selecting the proper level of flexibility for a model is known as model selection.

Cross-Validation

➢ Cross-Validation is a technique that is used to train and evaluate our model on a portion of our data, before re-portioning our dataset and evaluating it on the new portions.

➢ This means that instead of splitting our dataset into two parts, one to train on and another to test on, we split our dataset into multiple portions, train on some of these and use the rest to test on.

➢ We then use a different portion to train and test our model on. This ensures that our model is training and testing on new data at every new step.

Cross-Validation Techniques

There are many cross-validation techniques, some are given below:

★ **Hold Out method**

★ **K-Fold cross-validation**

★ **Stratified K-Fold cross-validation**

1. Hold Out method

- This is the simplest evaluation method and is widely used in Machine Learning projects. Here the entire dataset(population) is divided into 2 sets – train set and test set.

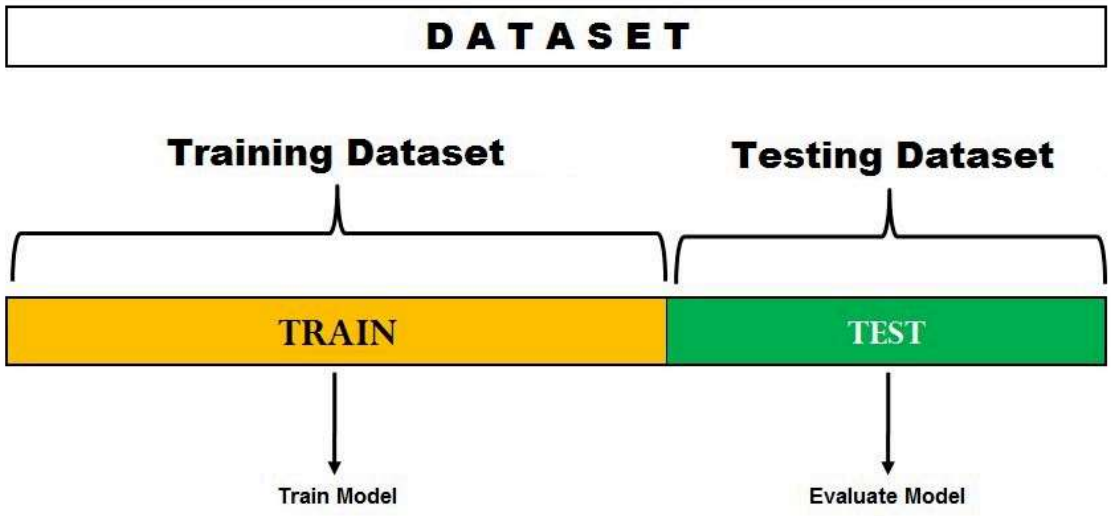
- The data can be divided into 70-30 or 60-40, 75-25 or 80-20, or even 50-50 depending on the use case. As a rule, the proportion of training data has to be larger than the test data.

- There are some drawbacks to this method:

- In the Hold out method, the test error rates are highly variable (high variance) and it totally depends on which observations end up in the training set and test set

- Only a part of the data is used to train the model (high bias) which is not a very good idea when data is not huge and this will lead to overestimation of test error.

- One of the major advantages of this method is that it is computationally inexpensive compared to other cross-validation techniques.

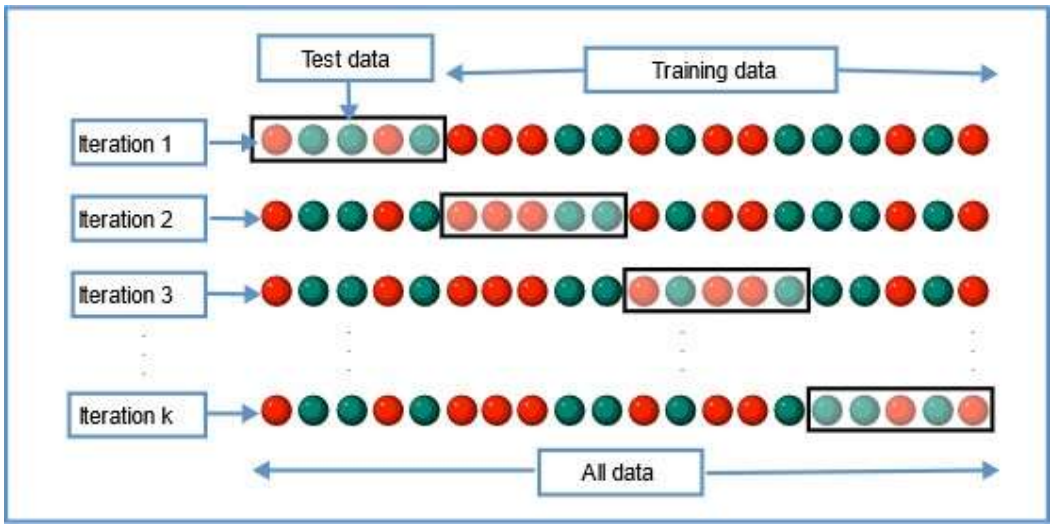


2. K-Fold cross-validation

- In K Fold cross validation, the dataset is split into k portions one section is for testing and the rest for training.

- Another section will be chosen for testing and the remaining section will be for training. This will continue K number of times until all sections have been used as a testing set once.

- Interchanging the training and test sets also adds to the effectiveness of this method. As a general rule and empirical evidence, K = 5 or 10 is generally preferred, but nothing's fixed and it can take any value.



---

3. Stratified K-Fold cross-validation

- This is slightly different from K-Fold Cross Validation, which uses **'stratified sampling'** instead of 'random sampling.'

- Suppose your data contains reviews for a cosmetic product used by both the male and female population. When we perform random sampling to split the data into train and test sets, there is a possibility that most of the data representing males is not represented in training data but might end up in test data. When we train the model on sample training data that is not a correct representation of the actual population, the model will not predict the test data with good accuracy.

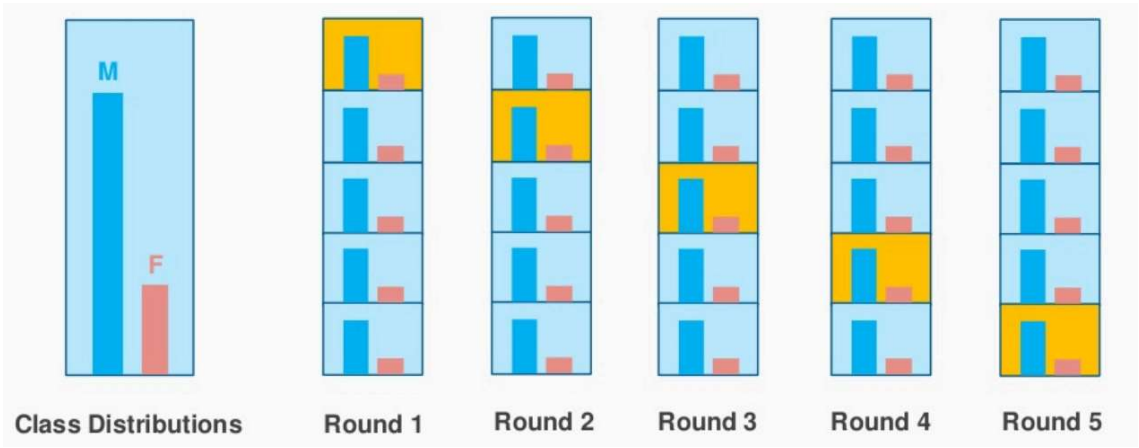
- This is where Stratified Sampling comes to the rescue. Here the data is split in such a way that it represents all the classes from the population.

Example:

- Let's consider the above example which has a cosmetic product review of 1000 customers out of which **60% is female and 40% is male**. I want to split the data into train and test data in proportion (80:20).

- 80% of 1000 customers will be 800 which will be chosen in such a way that there are 480 reviews associated with the female population and 320 representing the male population. In a similar fashion, 20% of 1000 customers will be chosen for the test data (with the same female and male representation).

- This is exactly what stratified K-Fold CV does and it will create K-Folds by preserving the percentage of sample for each class.


M
F
Class Distributions
Round 1
Round 2
Round 3
Round 4
Round 5