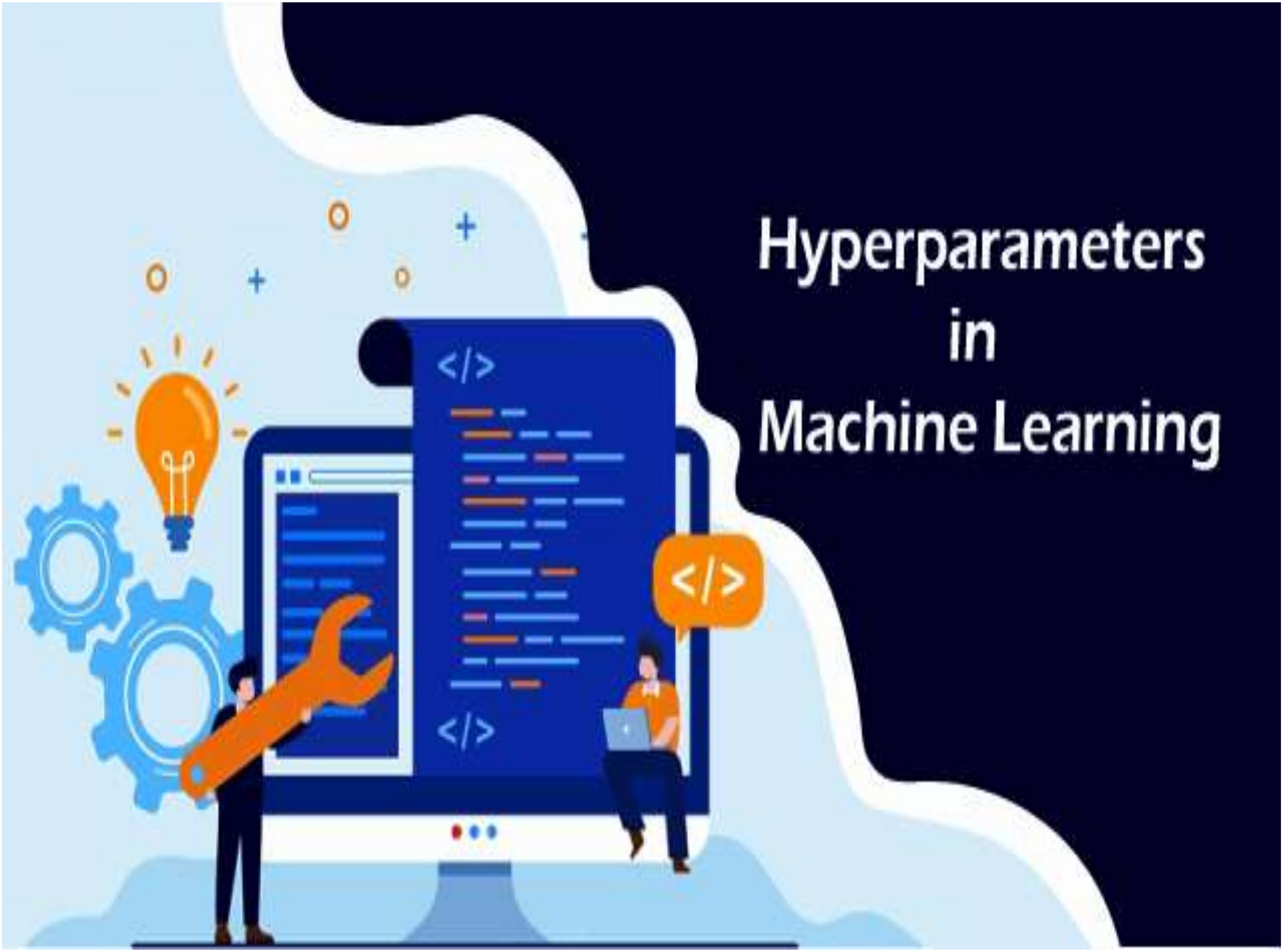

Hyperparameters
in
Machine Learning

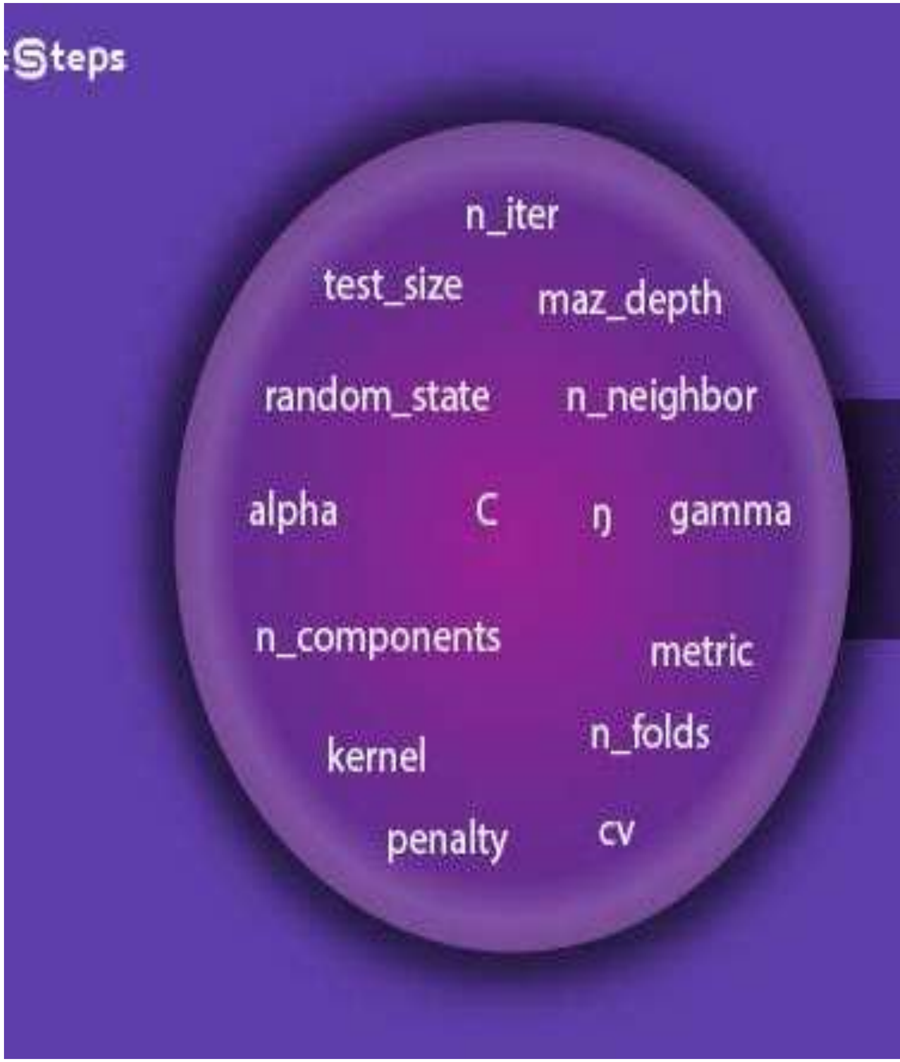
Hyper parameters

"Hyperparameters in Machine learning are those parameters that are explicitly defined by the user to control the learning process. "

- These hyperparameters are used to improve the learning of the model, and their values are set before starting the learning process of the model.
- These are external to the model, and their values cannot be changed during the training process

Examples of Hyperparameters in Machine Learning

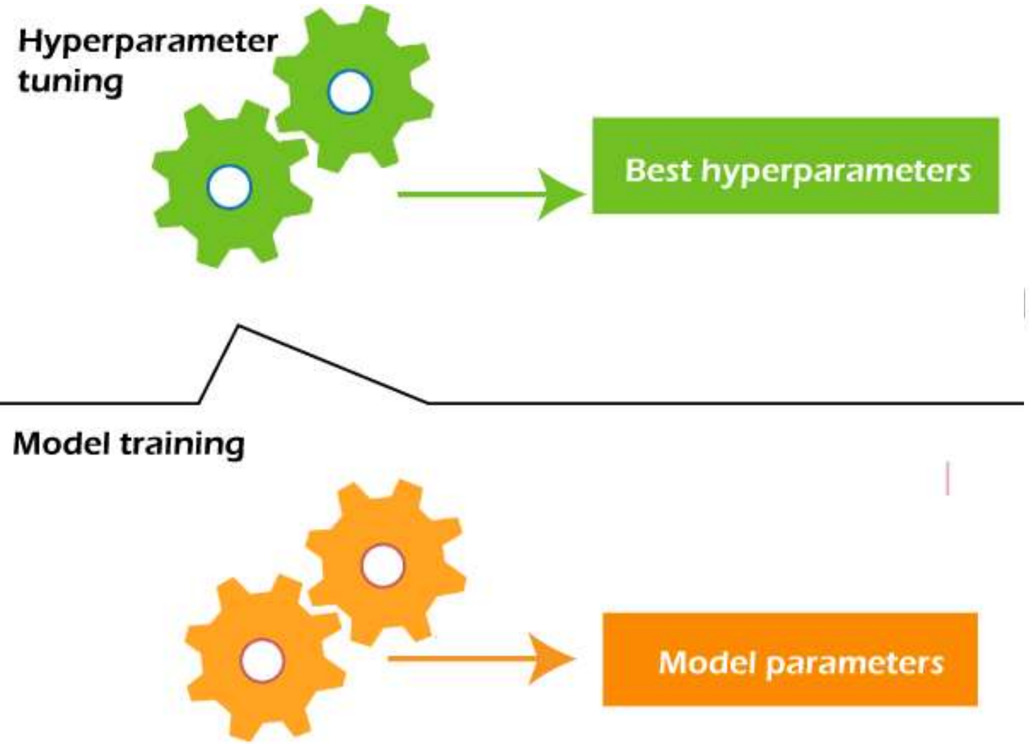
- The k in kNN or K-Nearest Neighbour algorithm
- Learning rate for training a neural network
- Train-test split ratio
- Batch Size
- Number of Epochs
- Branches in Decision Tree
- Number of clusters in Clustering Algorithm

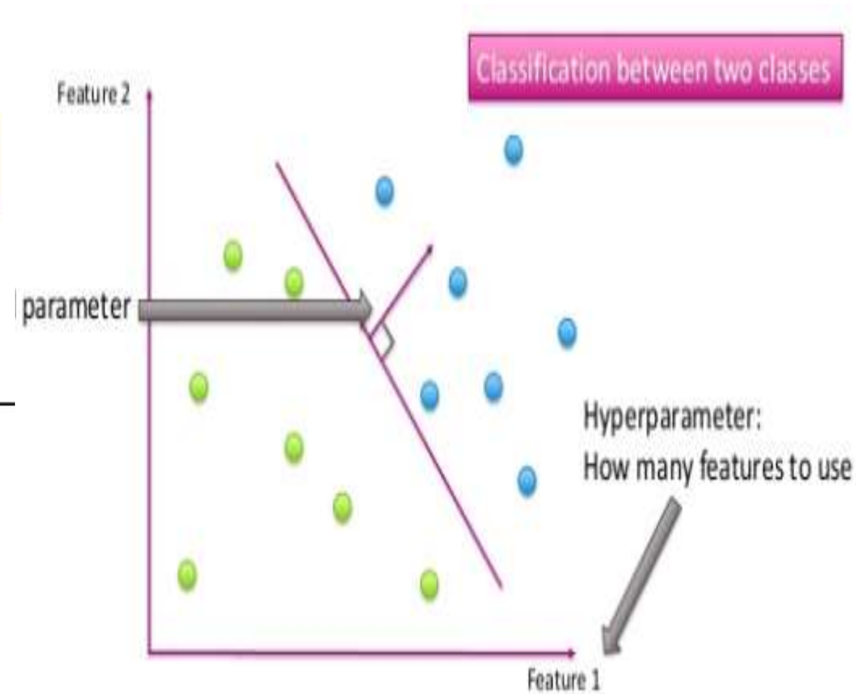


Comparison between Parameters and Hyperparameters

Parameters	Hyperparameters
Parameters are the configuration model, which are internal to the model.	Hyperparameters are the explicitly specified parameters that control the training process.
Parameters are essential for making predictions.	Hyperparameters are essential for optimizing the model.
These are specified or estimated while training the model.	These are set before the beginning of the training of the model.
It is internal to the model.	These are external to the model.
These are learned & set by the model by itself.	These are set manually by a machine learning engineer/practitioner.
These are dependent on the dataset, which is used for training.	These are independent of the dataset.
The values of parameters can be estimated by the optimization algorithms, such as Gradient Descent.	The values of hyperparameters can be estimated by hyperparameter tuning.
The final parameters estimated after training decide the model performance on unseen data.	The selected or fine-tuned hyperparameters decide the quality of the model.
Some examples of model parameters are Weights in an ANN, Support vectors in SVM, Coefficients in Linear Regression or Logistic Regression.	Some examples of model hyperparameters are the learning rate for training a neural network, K in the KNN algorithm, etc.

Parameters vs Hyperparameters





Categories of Hyperparameters

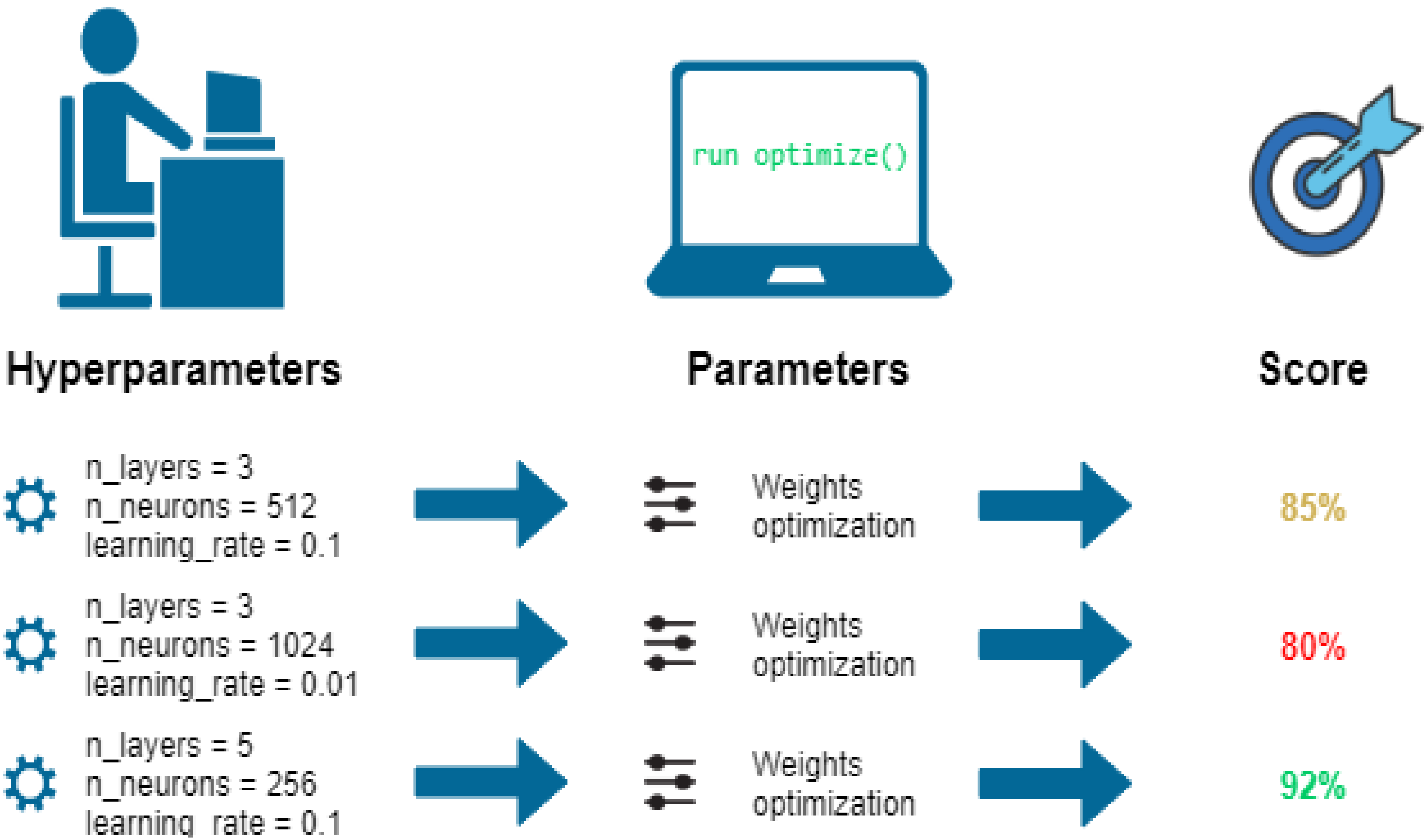
1. **Hyperparameter for Optimization:** The process of selecting the best hyperparameters to use is known as **Hyperparameter tuning**, and the tuning process is also known as hyperparameter optimization. Optimization parameters are used for optimizing the model.
2. **Hyperparameter for Specific Models:** Hyperparameters that are involved in the structure of the model are known as hyperparameters for specific models

Hyperparameter for Optimization

Some of the popular optimization parameters are given below:

- **Learning Rate:** The learning rate is the hyperparameter in optimization algorithms that controls how much the model needs to change in response to the estimated error for each time when the model's weights are updated.
- **Batch Size:** To enhance the speed of the learning process, the training set is divided into different subsets, which are known as a batch.
- **Number of Epochs:** An epoch can be defined as the complete cycle for training the machine learning model. Epoch represents an iterative learning process.

ML Optimization Workflow

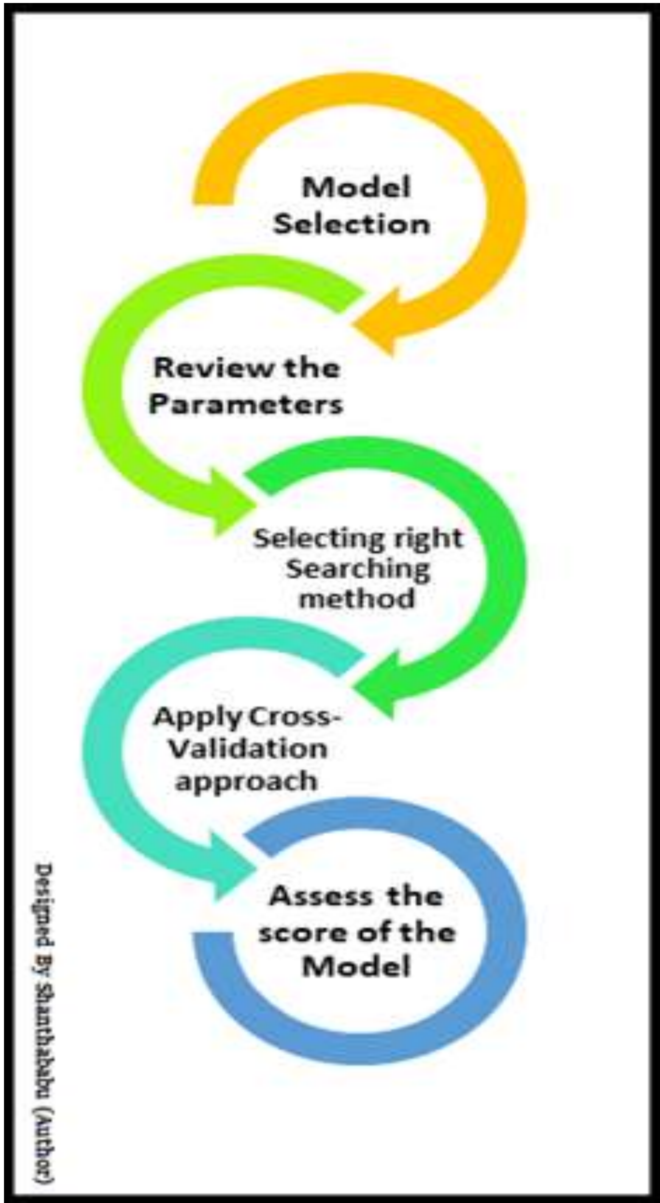


Hyperparameter for Specific Models

- **A number of Hidden Units:** Hidden units are part of neural networks, which refer to the components comprising the layers of processors between input and output units in a neural network.
- **Number of Layers:** A neural network is made up of vertically arranged components, which are called layers. There are mainly **input layers, hidden layers, and output layers**. A 3-layered neural network gives a better performance than a 2-layered network. For a Convolutional Neural network, a greater number of layers make a better model.

Steps to perform hyperparameter tuning

- Select the right type of model.
- Review the list of parameters of the model and build the HP space
- Finding the methods for searching the hyperparameter space
- Applying the cross-validation scheme approach
- Assess the model score to evaluate the model

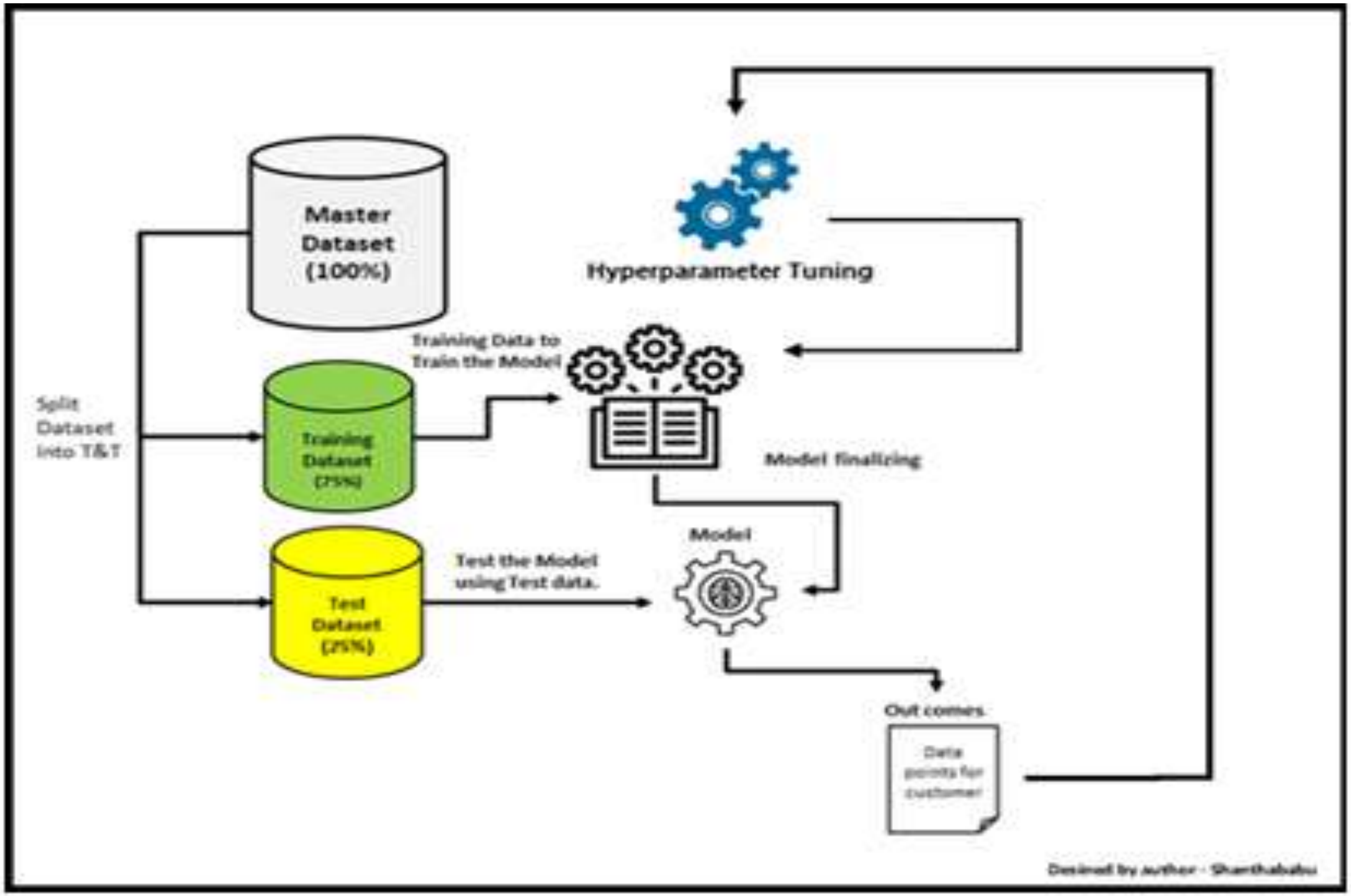


Influencing On Models

Overall, Hyperparameters are influencing the below factors while designing your model.

- Linear Model
  - What degree of polynomial features should use?
- Decision Tree
  - What is the maximum allowed depth?
- What is the minimum number of samples required at a leaf node in the decision tree?
  - Random forest
- How many trees we should include?
  - Neural Network
  - How many neurons we should keep in a layer?
- How many layers, should keep in a layer?
  - Gradient Descent
  - What learning rate should we?

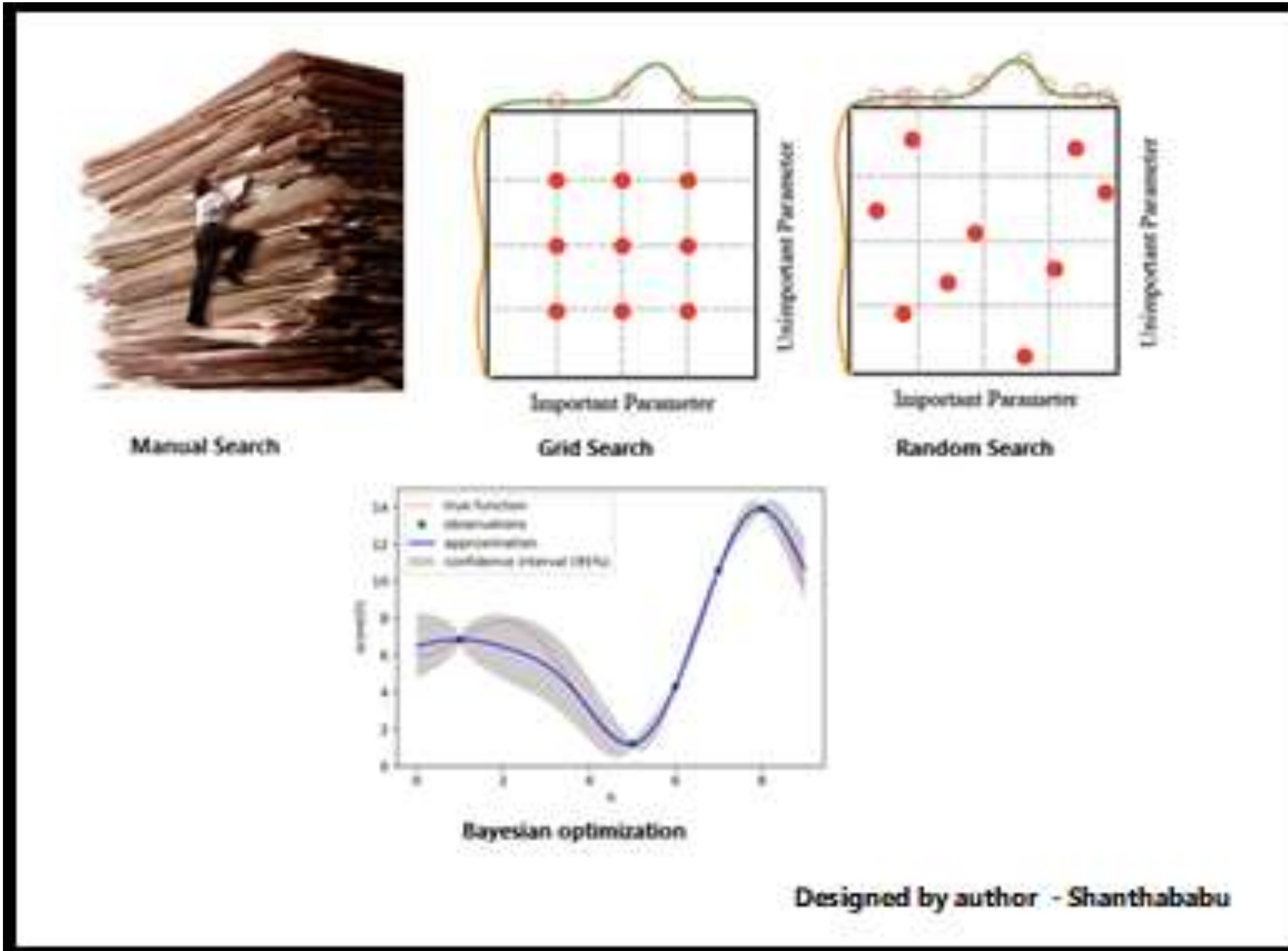
once we started thinking about introducing the hyperparameters in our model then the overall architecture model would be like below.



Hyperparameter Optimization Techniques

In the ML world, there are many Hyperparameter optimization techniques are available.

- Manual Search
- Random Search
- Grid Search
- Halving
  - Grid Search
  - Randomized Search
- Automated Hyperparameter tuning
  - Bayesian Optimization
  - Gradient Descent and
  - Evolutionary Algorithms
- Artificial Neural Networks Tuning
- HyperOpt-Sklearn
- Bayes Search



Designed by author - Shanthababu

- **Manual Search:**

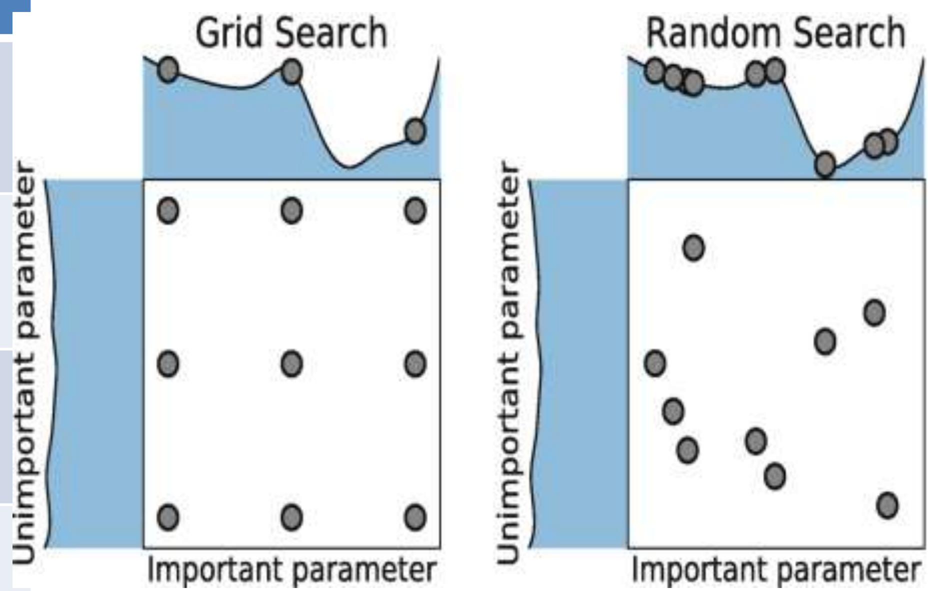
  The name itself is self-explanatory that the data scientist can do the experiment with different combinations of hyperparameters and its values for the selected model perform the training and pick up the best model with the best performance and go for testing and move on to production deployment. this method will consume immense effort.

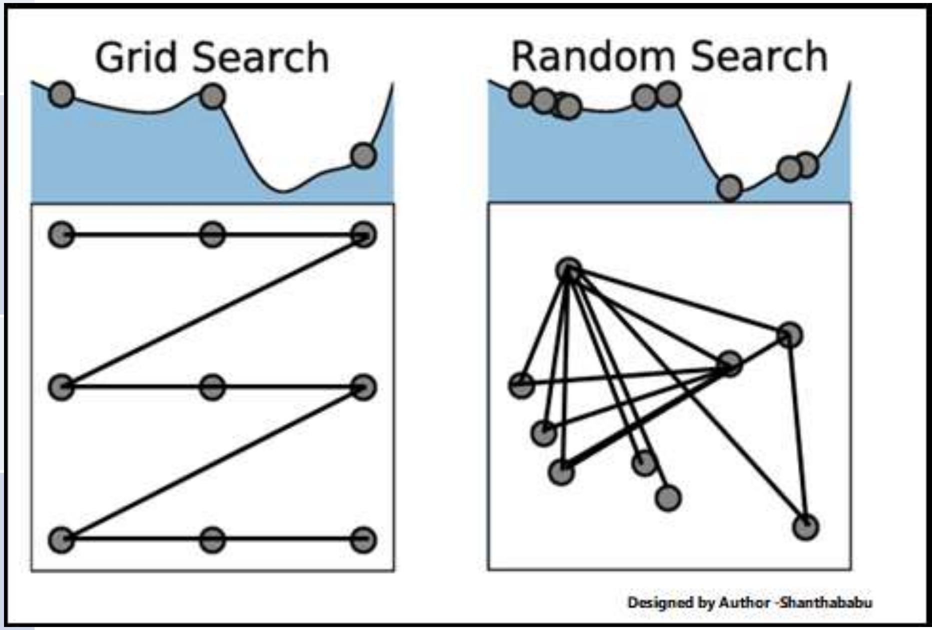
- **Grid-Search:** To implement the Grid-Search, we have a Scikit-Learn library called GridSearchCV. The computational time would be long, but it would reduce the manual efforts by avoiding the ‘n’ number of lines of code. Library itself perform the search operations and returns the performing model and its score. it searches for best set of hyperparameters from a grid of hyperparameters values.

- **Random Search:** The **Grid Search** one that we have discussed above usually increases the complexity in terms of the computation flow, So sometimes GS is considered inefficient since it attempts all the combinations of given hyperparameters. But the **Randomized Search** is used to train the models based on random hyperparameters and combinations. obviously, the number of training models are small column than grid search.

“In simple terms, In Random Search, in a given grid, the list of hyperparameters are trained and test our model on a random combination of given hyperparameters.”

GridSearchCV	RandomSearshCV
Grid is well-defined	Random is not well defined
Discrete values for HP-params	Continuos values and Statistical distribution
Defined size for Hyperparameter space	No such a restriction
Picks of the best combination from HP-Space	Picks up the samples from HP-Space
Samples are not created	Samples are created and specified by the range and n_iter
Low performance than RSCV	Better performance and result
Guided flow to search for the best combination	The name itself says that, no guidance.





- **Automated Hyperparameter Tuning :**When using Automated Hyperparameter Tuning, the model hyperparameters to use are identified using techniques such as:

- Bayesian Optimization
- Gradient Descent and
- Evolutionary Algorithms.

- Bayesian optimization:

- Bayesian optimization uses probability to find the minimum of a function. The final aim is to find the input value to a function which can give us the lowest possible output value.
- Bayesian has been proved to be more efficient than random, grid or manual search. Bayesian Optimization can, therefore, lead to better performance in the testing phase and reduced optimization time

➢ Evolutionary Algorithms

- Evolutionary algorithms (EA) are optimisation algorithms that work by modifying a set of candidate solutions (population) according to certain rules called Operators. One of the main advantages of the EA is their generality: Meaning EA can be used in a broad range of conditions due to their simplicity and independence from the underlying problem. Also known as Genetic Algorithm

➢ Gradient Descent

- Gradient descent is an optimization technique commonly used in training machine learning algorithms.
- It is a methodology to optimise several hyperparameters, based on the computation of the gradient of a machine learning model selection criterion with respect to the hyperparameters.
- This hyperparameter tuning methodology can be applied when some differentiability and continuity conditions of the training criterion are satisfied

Artificial Neural Networks (ANNs) Tuning

- Keras tuner is an open-source python library developed exclusively for tuning the hyperparameters of Artificial Neural Networks. it is possible to apply Grid Search and Random Search for Deep Learning models in the same way it was done when using scikit-learn Machine Learning models. To optimize some of our ANN parameters such as: how many neurons to use in each layer and which activation function and optimizer to use.

HyperOpt

- HyperOpt is an open-source Python library for Bayesian optimization developed by James Bergstra.
- It is designed for large-scale optimization for models with hundreds of parameters and allows the optimization procedure to be scaled across multiple cores and multiple machines.

HyperOpt-Sklearn

- An extension to HyperOpt was created called HyperOpt-Sklearn that allows the HyperOpt procedure to be applied to data preparation and machine learning models provided by the popular Scikit-Learn open-source machine learning library.
- HyperOpt-Sklearn wraps the HyperOpt library and allows for the automatic search of data preparation methods, machine learning algorithms, and model hyperparameters for classification and regression tasks.

BayesSearchCV:

- The BayesSearchCV class provides an interface similarto GridSearchCV or RandomizedSearchCV but it performs Bayesian optimization over hyperparameters.
- BayesSearchCV implements a “**fit**” and a “**score**” method and other common methods like predict(),predict_proba(),decision_function(), transform() and inverse_transform() if they are implemented in the estimator used.

Hyperparameters of machine learning algorithms

ML Algorithms	Hyperparameters	Definition	Defined Parameters
SVM (support vector machines)	Kernel type	the kernel function	RBF
	C	the penalty parameter	0.01–100
	σ	the bandwidth parameter	0.01–100
Cubist (regression tree)	committees	the number of model trees	1–100
	neighbors	the number of nearest neighbors	0–9
XGBoost (extreme gradient boosting)	booster	the type of model	gbtree
	max_depth	the depth of tree	3–10
	min_child_weight	the minimum sum of weights of all observations	0–5
	colsample_bytree	the number of variables supplied to a tree	0.5–1
	subsample	the number of samples supplied to a tree	0.5–1
	eta	learning rate	0.01–0.5
RF (random forest)	Mtry	the number of input variables	1–30
	Ntree	the number of trees	100–3000
ANN (artificial neural networks)	decay	learning rate	0.001–0.05
	size	the number of neurons in the hidden layer	1–10
DNN	hidden	the number of hidden layers	2–10